Database-SQL-RDBMS HOW-TO document for Linux (PostgreSQL Object Relational Database System)

Al Dev (Alavoor Vasudevan) *alavoor@yahoo.com* v45.0, 27 Gen 2001

Questo documento è una "guida pratica" volta ad una rapidissima installazione di un motore per Database SQL, e delle relative interfacce-utente, su un sistema Unix. Il documento tratta anche il linguaggio SQL (standard internazionale ANSI/ISO), e riesamina i meriti/vantaggi del motore per Database SQL, sviluppato attraverso la rete Internet, in un ambiente di "sviluppo aperto". Sarà trattata l'installazione, su un sistema Unix, dell'innovativo Database SQL Relazionale ad Oggetti "PostgreSQL", che può essere usato come Server Database per Applicazioni o come Server Database per il Web. PostgreSQL si sforza di implementare gli attuali e futuri standard SQL Internazionali ISO/ANSI. Questo documento fornisce anche informazioni sui programmi di interfaccia al database come: i Front End grafici, gli strumenti RAD (sviluppo rapido di applicazioni), i driver ODBC e JDBC, le interfacce di programmazione nei linguaggi "C", "C++", Java, Perl, e gli strumenti per l'uso di database nel Web. Le informazioni contenute in questo documento si applicano alle piattaforme Unix/Windows NT, e a tutti gli altri database SQL; queste informazioni saranno utilissime a chi non conosce i Database, il linguaggio SQL, e PostgreSQL. Questo documento contiene anche una guida a SQL, una sintassi di SQL che dovrebbe essere molto utile ai principianti. Gli esperti troveranno in questo documento un'utile guida di riferimento. Gli studenti vi troveranno informazioni che permetteranno loro di ottenere il codice sorgente del sistema di database relazionale PostgreSQL. Attraverso il sorgente si potrà apprendere come sia stato creato un motore per database SQL RDBMS. (Documentazione tradotta da Fabio Teatini <teafab@tiscalinet.it>, 7 Feb 2001 - Un ringraziamento a Michele Rossetto, che ha collaborato alla traduzione in qualità di revisore)

Indice

1 Introduzione

Con questo documento si vuole fornire un'esauriente lista di puntatori/URL, utili per installare rapidamente PostgreSQL; questo testo permetterà, inoltre, di promuovere il valore dei sistemi con Codice Open Source, come PostgreSQL e Linux.

PostgreSQL si pronuncia Post-gres-chiu-el (Postgres-QL) e **non** Postgre-es-chiu-el.

Ovunque nel mondo, i sistemi di computer hanno bisogno di un database per archiviare/recuperare le informazioni. Il computer è usato, principalmente, perché permette di archiviare, recuperare ed elaborare le informazioni, con grande rapidità ed economia di tempo. Nello stesso tempo, il sistema deve essere semplice, robusto, veloce, affidabile, economico, e molto semplice da usare. Il database diventa il principale **SISTEMA VITALE** quando archivia informazioni di importanza capitale, per le imprese nel mondo. Ogni industria nel mondo ha necessità di un sistema di database. Senza un sistema di database, le industrie di telecomunicazioni, di automobili, le banche, le compagnie aeree, ecc., non funzioneranno efficientemente. I sistemi di database più diffusi sono basati sulle specifiche SQL dell'International Standard Organisation (ISO), e sugli standard ANSI SQL (Americani). Le attuali specifiche, largamente usate nell'industria, sono le ISO/ANSI SQL 1992. È in ascesa lo standard SQL 1998/99, chiamato anche SQL-3, tuttora in fase di sviluppo. Database diffusi come i sistemi Oracle, Sybase, e Informix, sono basati su questi standard, o stanno cercando di implementarli.

Senza uno standard come l'ANSI/ISO SQL, sarebbe molto difficile sviluppare un'applicazione singola e renderla in grado di accedere a tutti i sistemi di database esistenti. L'utente finale vuole sviluppare una sola applicazione, e vuole che sia in grado di interagire (per mezzo di ISO SQL, ODBC, JDBC) con tutta la varietà di sistemi di database del mondo.

PostgreSQL, tra i database FREE che implementano parte dell'ISO SQL, ANSI SQL/98, SQL/92, e ANSI/SQL/89 RDBMS, è il più diffuso nel mondo. PostgreSQL è un innovativo database relazionale ad Oggetti, ed è volto ad essere pienamente conforme agli standard SQL come ISO/ANSI SQL. PostgreSQL è il solo RDBMS libero, nel mondo, che supporti database ad Oggetti e SQL. Questo documento vi dirà come installare il database, come avviare il database per il Web e per le applicazioni, come attivare i front-end grafici ed i programmi di interfacciamento. Se si desiderano applicazioni portabili verso molteplici database come

PostgreSQL, Oracle, Sybase, Informix, ecc., è vivamente consigliato sviluppare applicazioni per database rimanendo 100% conformi agli standard ISO/ANSI SQL, ODBC.

Con PostgreSQL si otterrà la più elevata qualità, e moltissime altre funzionalità, che provengono dal "modello di Programmazione Open Source". Il modello di Programmazione Open Source è quello che mette il codice sorgente a disposizione di tutti; lo sviluppo di questo codice avviene in internet, attraverso una vastissima rete di menti umane. Le analisi sulle tendenze del futuro mostrano che la maggior parte dello sviluppo di software avrà luogo in quella che è chiamata "Super-Autostrada dell'Informazione", che è estesa sull'intero globo. Negli anni a venire la crescita di Internet sarà esplosiva, e ciò favorirà una rapida adozione di PostgreSQL da parte dell'industria.

Applicando princìpi di statistica, matematica e scienza, alla qualità del software, si vedrà che la miglior qualità del software è ottenibile solo con "Sistemi di Programmazione Open Source", come PostgreSQL; il codice sorgente di questi sistemi è aperto ad un vastissimo numero di menti umane, interconnesse dalla super-autostrada dell'informazione. Quanto più elevato sarà il numero delle menti umane al lavoro, migliore sarà la qualità del software. Il modello di Programmazione Open Source ci preserverà anche dalla **RE-INVENZIONE DELLA RUOTA**, eliminerà la **DUPLICAZIONE DEL LAVORO**, sarà molto economico, e permetterà di abbreviare i tempi di distribuzione; il modello Open Source, inoltre, si allinea alle moderne leggi economiche, riguardo all'ottimizzazione delle risorse nazionali e globali. Un software già sviluppato da altri, **NON** dovrà essere creato nuovamente. Non dovrete sprecare il vostro tempo prezioso su qualcosa che sia già stato **BEN FATTO**. Il tempo è estremamente prezioso e deve essere utilizzato in modo efficiente, perché si hanno solo 8 ore al giorno per lavorare. Con l'approssimarsi del 21° secolo, ci sarà un mutamento nel modo in cui si potrà ottenere software per proprio uso. Tutti avranno un occhio di riguardo principalmente per i software open source, come PostgreSQL e Linux.

Acquistando programmi in forma di binari, non avrete un trattamento equo; inoltre vi sarà negata la proprietà del codice sorgente. Il codice sorgente è il bene più prezioso, mentre il codice binario non ha valore. Comprare software *può* diventare una cosa del passato. Comprate piuttosto buon hardware: è la sola cosa di cui avete bisogno; è conveniente utilizzare il proprio denaro per acquistare hardware, ed ottenere il software da internet. Il punto importante è: **il grosso del lavoro** è svolto dall'hardware del computer. L'hardware è il vero cavallo da tiro, ed il software è solo ciò che lo guida. L'hardware di un computer è così complesso che solo 6 nazioni nel mondo, finora, hanno dimostrato la capacità di progettare e produrre chip/hardware per computer. La progettazione, e la produzione di chip per computer, è una tecnologia avanzata. È un procedimento complesso, che assorbe grandi capitali, richiede grandi investimenti per impiantare fabbriche e macchinari di produzione, dove sono impiegate tecnologie a 0.18 micron (e si lavora anche su scala più piccola). Su un singolo e minuscolo chip di silicio sono miniaturizzati milioni di transistor/circuiti. Compagnie come Applied Material, AMD, Intel, Cyrix, Hitachi, IBM e altre, impiegano un significativo numero di anni-uomo per dominare le alte tecnologie come: la Progettazione di Chip, la Micro-elettronica, e la Nano-elettronica. Micro significa: un milionesimo di metro (10^-6), Nano significa: un miliardesimo di metro (10^-9). L'attuale tecnologia utilizza circuiti micro-elettronici delle dimensioni di circa 0.35 micron, impiegando l'alluminio come conduttore, mentre la tecnologia a 0.25 micron impiega il rame. Nel prossimo futuro, i chip per i computer saranno realizzati con la tecnologia a 0.10 micron, ed anche con la nano-elettronica. I conduttori di alluminio saranno sostituiti con il rame sui chip dei computer, poiché il rame è un miglior conduttore di elettroni. Tecniche di irraggiamento con elettroni, oppure con raggi X o estremo ultravioletto, tipiche dei processi fotolitografici, saranno utilizzate per eccitare i circuiti con dimensioni minori di 0.15 micron. In circa 20 anni a partire da oggi, i chip di silicio saranno sostituiti da computer molecolari e bio-chip; questi ultimi saranno miliardi di volte più veloci dei chip al silicio. Le molecole sono gruppi di atomi. E gli atomi sono piccole particelle che costituiscono tutto quello che vedete in questo mondo. I computer molecolari impiegheranno le molecole come interruttori elettronici ultra-veloci. Quando l'interruttore è su ON, indica il valore 1, e quando è su OFF, indica 0. Tutti i programmi per computer sono basati sui binari (i numeri 1 e 0). La tabella seguente mostra il progresso e l'andamento dei futuri sviluppi dei chip per computer.

```
Progressi delle future prestazioni dei chip
```

****

Tipo/Anno	1997	1999	2001	2003	2012	2020	2030
dimens.circuitali(micron)	0.25	0.18	0.15	0.13	0.05	< 0.00001	atomiche
dimensioni dei Wafer (mm)	200	300	300	300	450	Mol/Bio	Quantistiche
Tensione operativa Min.	1.8-2.5	1.5-1.8	1.2-1.5	1.2-1.5	0.5-0.6	< 0.001	infinitesima
Potenza dissipata Max	70	90	110	130	175	600	infinitesima
frequenza On-chip (MHz)	750	1,250	1,500	2,100	10,000	> 50,000	-----
capacità della DRAM	256 MB	1 GB	2 GB	4 GB	256 GB	> 1000GB	----

Come potete vedere, l'hardware è il vero protagonista dell'alta tecnologia; lavorare con il software richiede certamente un serio impegno, ma comporta una minore difficoltà tecnologica.

D'altra parte, tutti i paesi del mondo sviluppano/realizzano software. Infatti ogni persona in questo mondo, munita di un piccolo PC di basso costo, è in grado di scrivere programmi.

I database come Oracle, Informix, Sybase, IBM DB2 (Unix), sono stati scritti usando il linguaggio "C"; il codice binario, ottenuto compilando il codice sorgente, viene venduto direttamente ai clienti. I database Oracle, Sybase, Informix, sono programmi in "C" al 100% !!

Poiché negli ultimi 14 anni è stata fatta una gran mole di lavoro su PostgreSQL, non ha alcun senso ricreare da capo un altro sistema di database che soddisfi lo standard SQL ANSI/ISO. Sarà invece di grande vantaggio prendere il codice già esistente di PostgreSQL, e aggiungere ad esso le funzionalità mancanti o gli eventuali miglioramenti necessari, per usarli immediatamente.

Si prevede che la domanda di "prodotti Internet" come PostgreSQL crescerà esponenzialmente, in misura di quanto questi potranno mantenersi di alta qualità, basso costo, e diffusione estremamente vasta sia sul lato utenti, sia su quello degli sviluppatori. Quelle nazioni che non usano "prodotti Internet" perderanno clamorosamente la "Rivoluzione della rete Internet", e rimarranno indietro rispetto agli altri paesi. La ragione è che "Internet" stessa è **LA PIÚ ESTESA** industria di software del mondo, ed è una grande e potente "software house".

1.1 Computer Quantistici - La Fisica Quantistica Utile !!

Come potete vedere dalla precedente tabella, dedicata ai "Progressi delle future prestazioni dei chip", negli anni successivi al 2030 i sistemi di database come PostgreSQL gireranno su **Computer Quantistici**. I Computer Quantistici utilizzano le caratteristiche di una particella atomica, come la direzione dello spin, per creare uno stato (cioè: uno stato logico, n.d.t.). Per esempio, quando lo spin è su, una particella potrebbe essere letta come **"uno"**; quando il suo spin è giù, la particella dovrebbe essere letta come **"zero"**. Atomi e nuclei possono esistere in uno stato-sovrapposizione (di altri stati n.d.t.), in modo tale che tutti i valori da zero ad uno possono essere rappresentati. Intrappolando gli spin degli atomi, i **"qubit"** possono essere connessi gli uni agli altri; ciò conferisce loro la capacità di agire come un tutto unico, e permette di raggiungere la potenza del calcolo non lineare, che **sorpasserà di gran lunga** le possibilità dei supercomputer oggi disponibili!! A livello atomico, la Fisica Quantistica ci viene in aiuto per una migliore comprensione del comportamento delle particelle atomiche.

2 Le leggi della Fisica si applicano al Software!

In questo capitolo sarà mostrato come la scienza giochi un ruolo importante nella creazione di svariati oggetti come: il software, quest'universo, la massa, gli atomi, l'energia, ed anche voi stessi! Questo capitolo mostra anche quanto sia importante conoscere la scienza, prima di iniziare ad usarne i prodotti.

La regola d'oro è: *"NON si DEVE usare un prodotto senza comprendere come esso sia stato creato!!"* Questa regola si applica ad ogni cosa: sistemi di database, sistemi di computer, sistemi operativi, quest'universo e anche il vostro corpo umano! Ciò significa che dovreste cercare di avere tutte le informazioni, e il codice completo del sistema. È importante comprendere come funzionino il corpo umano e gli atomi al suo interno, perché gli esseri umani creano PostgreSQL, MS Windows95, ecc..

La creazione è un passo molto importante. Le persone che usano gli oggetti della scienza, devono sapere come questi vengono creati. Ciò si applica anche ai sistemi di computer e a PostgreSQL. La maggior parte della gente non ha cognizioni di scienza, e quindi non sa come siano stati creati i sistemi come MS Windows NT/95, Oracle, il corpo umano e quest'universo. Una grande maggioranza di persone non sa come sono fatti l'universo e MS Windows 95/NT, e cosa c'è dentro di essi. I sistemi complessi sono costruiti utilizzando semplicissimi mattoni elementari: ognuno dei milioni di universi che sono stati creati ha milioni di super-ammassi di galassie; ogni super-ammasso ha milioni di galassie; ogni galassia ha milioni di stelle; qualche stella ha molti pianeti; ogni pianeta è costituito da miliardi di atomi.*(Nella storia di questo mondo, **solo un universo è stato creato da un uomo** nell'antica India, eoni fa; ma nessun altro caso ci è stato riportato nella storia moderna. Esiste solo un universo creato dall'uomo).* Creare un universo richiede una tecnologia molto avanzata; molto più di quella della bomba atomica, sganciata su Hiroshima e Nagasaki, che ha causato **un'orribile distruzione**. Le armi nucleari moderne sono così minuscole e potenti che, se una di queste bombe fosse sganciata nell'Oceano Pacifico, potrebbe vaporizzare completamente il pianeta terra! L'intera varietà di armi è infinita. Esistono armi in grado, perfino, di distruggere gli universi (fornire a chiunque la tecnologia delle armi nucleari, technology to every person). Nuclear weapons and other non è una buona idea). Armi nucleari ed altre **armi divine ancora più potenti**, sono state usate sui campi di battaglia dell'antica India! Nessuno credette ad Albert Einstein (uno scienziato del 1900), quando previde che le armi nucleari sarebbero state in grado di vaporizzare grandi città.

Il software come MS Windows 95 è stato creato semplicemente con i linguaggi di programmazione "C" e assembler, utilizzando solo 1 e 0; ***gli universi come il nostro sono stati creati semplicemente dall'interazione di un'opportuna combinazione di minuscole particelle elementari, provenienti da altre dimensioni.*** *(Qualcosa di interessante è avvenuto **appena prima** l'interazione delle minuscole particelle).* Un corpo umano viene creato in seguito all'interazione di un'opportuna combinazione di piccole cellule!! *(Qualcosa di interessante avviene **appena prima** l'interazione delle piccole cellule).* Gli esseri umani hanno ereditato le proprietà di quest'universo. L'universo, in cui state vivendo attualmente, NON esisteva: tutti gli atomi contenuti nell'universo non c'erano per niente, e neanche il TEMPO esisteva!! Il neonato universo è stato generato in seguito ad un grande scoppio (il big bang), ha iniziato ad espandersi, e ha continuato a crescere. Oggi, il nostro universo non è statico, ed è ancora in espansione!! Una persona di nome **'Brahma'**, proveniente da un altro universo, ha creato l'universo in cui state vivendo attualmente. ***La Conoscenza è la MADRE di quest'universo!!*** 'Brahma' ha amato 'Madre Conoscenza' prima della nascita dell'universo in cui state vivendo!! È qualcosa di simile a quel che è successo quando voi siete nati! Senza i 'geni' di Madre Conoscenza, non è possibile creare neanche un piccolo programma in "C" !

Il nostro universo ricollasserà in qualche punto (in un 'big crunch'), e tutti gli atomi in esso contenuti svaniranno, e scompariranno completamente! Tutti gli atomi, che si vedono in questo universo, saranno dispersi!

Il numero totale di universi che sono stati creati è **INFINITO** e, analogamente, il numero totale dei sistemi operativi che possono essere creati, è anch'esso **infinito**!! È un processo ciclico infinito, che vede universi nascere e morire. Esistono milioni di universi che possono essere classificati in 3 categorie principali. Un numero infinito di universi, ed un'infinita varietà di atomi multi-dimensionali, collassano in pochi *universi-dimensionali-primari*. E gli universi-dimensionali-primari collassano in un'unica entità concentrata, chiamata *'eeshwar'* (eeshara è una parola in ***sanscrito***). Equazioni matematiche molto avanzate danno fondamento a questa teoria.

Le leggi della scienza e della statistica giocano a favore dei sistemi open-source come PostgreSQL e Linux. Con le crescenti velocità ed affidabilità di Internet, il sistema di programmazione open-source guadagnerà sempre più slancio. Quindi, se le leggi della statistica e della fisica sono corrette, con l'accrescersi della consapevolezza della scienza, e il suo apprendimento da parte della gente **IGNORANTE**, accadrà che i sistemi a codice sorgente chiuso spariranno da questo pianeta.

Sviluppare un progetto come PostgreSQL richiede risorse come energia e tempo, perciò PostgreSQL è un prodotto di energia e tempo. Poiché l'energia ed il tempo possono essere spiegati solo per mezzo della scienza, c'è una diretta correlazione tra la fisica e i progetti software come PostgreSQL e Linux. Le leggi della scienza (Fisica) si applicano a qualsiasi cosa voi facciate, anche quando state sviluppando progetti software.

La fisica è in azione anche quando state parlando (onde sonore), camminando (attrito tra la terra e i vostri piedi), quando leggete un libro, o scrivete software. Ogni scienza di questo mondo ha profonde radici nella matematica, e questo vale anche per PostgreSQL. PostgreSQL impiega l'"Algebra Moderna", che è una piccola branca della matematica. L'Algebra moderna tratta di "Teoria degli Insiemi", "Algebra relazionale", teoria dei Gruppi, Anelli, Collezioni, Insiemi, Unioni, Intersezioni, Esclusioni, Domini, Liste, ecc...

Il software come PostgreSQL esiste, oggi, a causa di energia e tempo. Inoltre massa ed energia sono UNA cosa sola, una **STESSA** entità. Il fatto che massa ed energia siano la stessa cosa, era ignoto alla gente vissuta fino a 100 anni fa! E anche **oggi** la popolazione mondiale non sa che Internet è la più estesa "azienda di sviluppo di software", e la più grande "compagnia di software" del mondo!

Le cellule nel cervello umano consumano energia mentre lavorano (creando software): convertono l'energia chimica, tratta dal cibo, in energia elettrica e calore. Anche mentre state leggendo questo paragrafo, le cellule nel vostro cervello stanno bruciando il combustibile, e stanno utilizzando piccole quantità di energia. Tutto ciò implica che il cervello umano è un motore termodinamico. Poiché il cervello umano è un motore termodinamico, ne deriva che le leggi della termodinamica si applicano al cervello, e così la termodinamica ha effetti indiretti sul software come PostgreSQL.

Può esserci un infinito numero di colori, di linguaggi di programmazione, di progetti di chip e di teorie; ma NON esiste UN SINGOLO E PERFETTO colore, linguaggio, progetto o sistema! Quel che si può avere, è solo un QUASI PERFETTO colore (lunghezza d'onda), sistema di database, o teoria! **La natura è come un *CALEIDOSCOPIO***: c'è un infinito numero di dimensioni, un'infinita varietà di particelle di altre dimensioni, ma essi si combinano in pochissime dimensioni primarie, e viceversa.

Combinando via Internet le energie di milioni di persone intorno al mondo, è possibile ottenere un sistema **QUASI PERFETTO** (ad esempio un sistema di software per database). Presa individualmente, l'energia di una singola persona è insignificante; tuttavia, interconnettendo un vasto numero di persone, l'energia totale sarà enorme, e potrà essere concentrata su un progetto per generare un sistema quasi perfetto.

L'energia è misurata in Joule, kiloJoule o kilogrammo-massa; il tempo è misurato in secondi o ore. La potenza è l'energia divisa per il tempo, ed è misurata in Watt o kiloWatt .

```
        Energia di ogni persona = y Joule
o, in termini di massa
        energia di ogni persona = y grammi
Il fattore di conversione tra massa ed energia è: E = m * c * c
dove 'c' è la velocità della luce e 'm' è la massa.
        Tempo = 8 ore (Questa è una costante perché ogni persona lavora solo 8 ore al giorno)
        Potenza = Energia / Tempo
```

```
                 = (y / (8 * 60 * 60) ) Watt
        Potenza Totale del mondo = n * (y / (8 * 60 * 60) ) Watt
dove n = numero di persone che collaborano al progetto.
```

Dall'equazione sopra riportata risulta che, incrementando la 'n', si migliorerà notevolmente la qualità del prodotto. È sorprendente come la rete globale Internet possa concentrare così tanta energia (espressa in kiloJoule) e potenza (in kiloWatt) su sistemi come Linux e PostgreSQL!

Internet può interconnettere un grande numero di persone, e così Internet possiede una grande quantità di energia e di tempo; ciò porta a produrre software della più alta qualità, in molto meno tempo rispetto alle compagnie commerciali. Anche compagnie molto grandi, come Microsoft e IBM, non possono sopraffare ed annullare le leggi della Fisica; piuttosto dovranno **ARRENDERSI** alle leggi della scienza!

La conclusione è: in virtù delle leggi della scienza, i sistemi con "codice open source", come PostgreSQL e Linux, prevarranno e saranno sempre migliori dei sistemi con "codice sorgente chiuso"; è possibile provare quest'affermazione in modo scientifico. Gli uomini non dovrebbero sprecare tempo nel creare troppi prodotti software duplicati.

3 Cos'è PostgreSQL ?

PostgreSQL è un database gratuito e libero, il cui codice sorgente completo è disponibile liberamente per tutti; è un Sistema di Database Relazionale ad Oggetti, ed ha come obiettivo la totale conformità a SQL/ISO-ANSI 1998,92; funziona su diverse piattaforme hardware e vari Sistemi operativi. Obiettivo ultimo, e scopo finale di PostgreSQL, è diventare aderente al 100% a SQL ANSI/ISO, e diventare il numero UNO del mondo tra i Database genericamente "open".

PostgreSQL si pronuncia Post-gres-chiu-el (Postgres-QL) e **non** Postgre-es-chiu-el.

Oggi PostgreSQL è **il più avanzato** sistema del mondo, ed è sorprendente che molti sistemi di database commerciali non raggiungano la qualità, le funzionalità, e le capacità di PostgreSQL !! PostgreSQL è lo sforzo congiunto di molte nazioni intorno al globo, ed è un progetto paragonabile a quello della **Stazione Spaziale Internazionale**. PostgreSQL rimarrà il **sistema di database numero uno** nei decenni a venire, perché è un sistema realizzato con il metodo open-source.

L'idea fondamentale dietro PostgreSQL è quella secondo cui, una volta che un modulo di codice sia già stato scritto, non dovreste sprecare neanche un milli-secondo del vostro tempo nel provare a reinventarlo!!

Informix Universal server (rilasciato nel 1997) è basato su una prima versione di PostgreSQL, in quanto Informix comprò Illustra Inc. e l'integrò con Informix stessa. Il database Illustra era basato su Postgres (una prima versione di PostgreSQL).

PostgreSQL è un ampliamento del sistema di gestione di database POSTGRES, è un DBMS di nuova generazione, ed è un prototipo di ricerca. PostgreSQL permette l'uso dei modelli di dati più potenti, ed una maggiore ricchezza di tipi di dati rispetto a POSTGRES; inoltre il linguaggio di interrogazione PostQuel è stato sostituito con un sottoinsieme esteso di SQL.

Lo sviluppo di PostgreSQL è stato realizzato da un gruppo di sviluppatori attivi in Internet, tutti iscritti alla mailing list di sviluppo di PostgreSQL. L'attuale coordinatore è Marc G. Fournier

- *scrappy@postgreSQL.org*

Questo gruppo è ora responsabile dello sviluppo attuale e futuro di PostgreSQL. Naturalmente gli utilizzatori del database sono gli stessi sviluppatori di PostgreSQL! Il carico di sviluppo è distribuito, in internet, su un grandissimo numero di utenti finali del database.

Gli autori di PostgreSQL 1.01 sono stati Andrew Yu e Jolly Chen. Il codice del Postgres originale, da cui PostgreSQL è derivato, è stato il risultato dello sforzo di molti laureati, studenti universitari, gruppi di programmatori, tutti al lavoro sotto la direzione del Professor Michael Stonebraker dell'Università della California, a Berkeley.

Milioni di PostgreSQL sono installati come server Database, server Database per Web, e server di Dati per Applicazioni. PostgreSQL è un sofisticatissimo sistema di database relazionale ad oggetti (ORDBMS).

PostgreSQL gira su Solaris, SunOS, HPUX, AIX, Linux, Irix, Digital Unix, BSDi, NetBSD, FreeBSD, SCO unix, NEXTSTEP, Unixware, e su tutte le versioni di Unix. Una versione per Windows NT è stata realizzata usando il pacchetto cygwin32 della Cygnus.

PostgreSQL, e gli argomenti connessi in questo documento, sono soggetti al COPYRIGHT dell'Università della California, Berkeley.

3.1 Libro Bianco

PostgreSQL in pochi dettagli:

- Titolo: PostgreSQL SQL RDBMS Database (Object Relational Database Management System)
- Versione Attuale: 7.0.1
- Età: PostgreSQL ha 15 anni. È stato sviluppato a partire dal 1985
- Autori: Sviluppato da milioni di università/compagnie su Internet, durante i trascorsi 15 ANNI

Il libro bianco su PostgreSQL si trova al sito `<http://www.greatbridge.com>`

PostgreSQL si pronuncia Post-gres-chiu-el (Postgres-QL) e **non** Postgre-es-chiu-el.

4 PostgreSQL o MySQL ? Quale dei due?

4.1 PostgreSQL ha sconfitto Oracle, IBM DB2, MS SQL server e altri!!

PostgreSQL ha battuto Oracle 8 (e 8i), IBM DB2, MS SQL server, Sybase, Interbase e MySQL, nei benchmark standard relativi a test su prestazioni, velocità, scalabilità ed affidabilità! Leggete i benchmark al sito `<http://www.aldev.8m.com>`, o presso `<http://aldev.webjump.com>`.

4.2 MySQL e altri RDBMS duplicati

MySQL è un altro server SQL open-source, ma non gestisce le transazioni. È adatto ad una base di dati molto piccola, e non supporta le funzionalità SQL avanzate. Invece PostgreSQL è un database adatto a grandi aziende, supporta le transazioni e quasi tutti i costrutti SQL. PostgreSQL è molto più avanzato dei database commerciali come Oracle, Sybase e Informix. PostgreSQL supporta meccanismi di locking molto avanzati, e molte altre funzionalità innovative che non sono disponibili sui sistemi di database commerciali!!

In un prossimo futuro lo sviluppo di MySQL cesserà, poiché MySQL, in qualità di prodotto funzionante secondo l'ANSI SQL, è un prodotto duplicato. Dovremmo scegliere il server SQL open-source più maturo ed avanzato, lasciando perdere tutti gli altri, perché non dobbiamo perdere il nostro tempo (ad occuparci di tutti i RDBMS esistenti)!! Infatti, avete tempo per occuparvi solo del server SQL più potente, che è PostgreSQL! Ne segue che tutti gli utenti di MySQL migreranno verso PostgreSQL. Inoltre MySQL è un prodotto "quasi commerciale", contrariamente a PostgreSQL, che è open-source e non richiede il pagamento di licenze. Non c'è alcuna necessità di un altro sistema di database come PostgreSQL, in questo mondo!!

I prodotti duplicati, come MySQL, confondono la base di utenti e causano divisioni di risorse. Se si vuole un sistema **"QUASI PERFETTO"**, deve esserci **un solo** sistema, e ognuno, nel mondo, deve lavorare su di esso!! I prodotti duplicati causano più danni che vantaggi, e quindi le frammentazioni di risorse devono essere fortemente scoraggiate. Questo è già successo nel caso dei sistemi di database commerciali Oracle, Sybase, Informix e MS SQL server, che hanno causato la scissione della base di utenti, e sono spesso tra loro incompatibili. ***Voglio mettere il codice sorgente del server SQL sotto il vostro controllo!!!***

Non vi serve un centinaio di sistemi di database; tutto quello che vi serve è soltanto il migliore server database che, guarda caso, è proprio 'PostgreSQL'.

ATTENZIONE: A partire da specifiche come quelle di ANSI SQL, è possibile creare un infinito numero di sistemi di database !!

Le funzionalità, assenti in MySQL e supportate da PostgreSQL, sono:

- Le transazioni
- Le stored procedure
- I trigger (per update, insert e delete)
- I database Orientati agli Oggetti
- Il sistema avanzato di locking, la gestione di multi-utenza simultanea, l'ambiente multi-transazionale
- Le Interrogazioni annidate (Sub-query)
- I cursori dal lato server
- Il caching delle query
- Il locking dei database
- Un miglior supporto delle join sulle tabelle (JOIN, UNION, MINUS, INTERSECT, join esterna)
- E molte altre funzionalità più avanzate, troppo numerose da elencare qui.

MySQL si trova al sito `<http://www.tcx.se>`

4.3 Le Limitazioni di MySQL

PostgreSQL dovrebbe essere comparato con sistemi come Oracle, perché entrambi sono pienamente conformi alle proprietà ACID, nonché robusti sistemi sviluppati da lunghissimo tempo. È del tutto fuori luogo comparare MySQL con Oracle, oppure MySQL con PostgreSQL. Per altri dettagli leggete *Why Not MySQL* `<http://openacs.org/philosophy/why-not-mysql.html>`. È chiaro, quindi, che si commetterà un grave errore nel sostituire Oracle con MySQL!! Se volete sostituire Oracle, prendete in considerazione PostgreSQL.

5 Dove ottenerlo ?

Potete comprare i CDROM di Linux Redhat, Debian o Slackware, che contengono già PostgreSQL in forma di pacchetti (sia in codice sorgente che in binari), presso:

- Linux System Labs al sito Web: `<http://www.lsl.com/>` (7 dollari U.S.A.)

- Cheap Bytes Inc al sito Web: `<http://www.cheapbytes.com/>` (7 dollari U.S.A.)
- Debian al sito web principale: `<http://www.debian.org/vendors.html>`

Anche l'organizzazione di PostgreSQL vende il "CDROM PostgreSQL", che contiene il codice sorgente completo ed i binari per molti sistemi operativi Unix, insieme a tutta la documentazione.

- CDROM PostgreSQL, presso il sito Web principale: `<http://www.postgresql.org>` (30 dollari U.S.A.)

Distribuzione dei soli binari di PostgreSQL:

- Il curatore dei pacchetti RPM di PostgreSQL è Lamar Owen, che ha l'e-mail *lamar.owen@wgcr.org*.
- pacchetti RPM di sorgenti e binari di PostgreSQL `<http://www.ramifordistat.net/postgres>`
- pacchetti RPM di sorgenti e binari di PostgreSQL `<http://www.postgresql.org>`. Fate click su "Latest News", e poi sugli RPM Redhat.
- pacchetti RPM di sorgenti e binari di PostgreSQL `<http://www.redhat.com/pub/contrib/i386/>`; il sito ftp è `<ftp://ftp.redhat.com/pub/contrib/i386/>`
- sito dei binari per Solaris, HPUX, AIX, IRIX, Linux: `<ftp://ftp.postgresql.org/pub/bindist>`

Siti Web WWW:

- Sito Web primario: `<http://www.postgresql.org/>`
- Sito Web Secondario: `<http://logical.thought.net/postgres95/>`
- `<http://www.itm.tu-clausthal.de/mirrors/postgres95/>`
- `<http://s2k-ftp.cs.berkeley.edu:8000/postgres95/>`
- `<http://xenium.pdi.net/PostgreSQL/>`
- `<http://s2k-ftp.cs.berkeley.edu:8000/postgres95/>`

I siti ftp sono elencati di seguito:

- FTP Primario: `<ftp://ftp.postgresql.org/pub>`
- FTP Secondario: `<ftp://ftp.chicks.net/pub/postgresql>`
- `<ftp://ftp.emsi.priv.at/pub/postgres/>`
- `<ftp://ftp.itm.tu-clausthal.de/pub/mirrors/postgres95>`
- `<ftp://rocker.sch.bme.hu/pub/mirrors/postgreSQL>`
- `<ftp://ftp.jaist.ac.jp/pub/dbms/postgres95>`
- `<ftp://ftp.luga.or.at/pub/postgres95>`
- `<ftp://postgres95.vnet.net:/pub/postgres95>`
- `<ftp://ftpza.co.za/mirrors/postgres>`
- `<ftp://sunsite.auc.dk/pub/databases/postgresql>`

- <ftp://ftp.task.gda.pl/pub/software/postgresql>
- <ftp://xenium.pdi.net/pub/PostgreSQL>

Il codice sorgente di PostgreSQL è ottenibile anche presso tutti i siti mirror di sunsite unc (per un totale di circa 1000 siti intorno al globo). Inoltre, il sorgente è incluso nella distribuzione Red Hat: è nel file /pub/contrib/i386/postgresql.rpm .

- Per una lista di siti mirror vai a <ftp://sunsite.unc.edu>

6 Istruzioni per un'installazione rapida di PostgreSQL

PostgreSQL si pronuncia Post-gres-chiu-el (Postgres-QL), e **non** Postgre-es-chiu-el.

Questo capitolo vi aiuterà ad installare e a lanciare il database molto rapidamente: in meno di 5 minuti.

6.1 Installazione e Collaudo

Installazione, Collaudo, Verifica, e lancio di PostgreSQL, in pochi passi. Effettuate il login come root.

```
# cd /mnt/cdrom/RedHat/RPMS
# man rpm
# ls postgre*.rpm
# rpm -qpl postgre*.rpm | less (per vedere la lista dei file)
# rpm -qpi postgre*.rpm (per vedere le informazioni dei pacchetti)
# cat /etc/passwd | grep postgres
```

Nota: Se esiste un utente "postgres", potreste dover fare backup dei suoi file, cancellare la home directory di postgres, ~postgres, e poi dovreste eliminare l'utente unix "postgres"; oppure potreste rinominarlo come "postgres2", o qualcosa di simile. L'installazione deve essere realizzata come se fosse la prima volta.

```
# rpm -i postgre*.rpm (per installare tutti i pacchetti dei client, di sviluppo, dei dati,
         e quelli principali per il funzionamento di pgaccess)
# man chkconfig
# chkconfig --add postgresql  (per avviare pg in fase di boot)
# /etc/rc.d/init.d/postgresql start  (per avviare postgres)
# man xhost
# xhost +  (affinché pgaccess abbia accesso al display)
# su - postgres
bash$ man createdb
bash$ createdb miodatabase
bash$ man psql
bash$ psql miodatabase
 ..... in psql, premete i tasti freccia su/giù per utilizzare lo storico dei comandi;
oppure usate il comando \s

bash$ export DISPLAY=<nomehost>:0.0
bash$ man pgaccess
bash$ pgaccess miodatabase
```

Adesso potete cominciare a **SPARARE** comandi SQL in psql o in pgaccess !!

```
bash$ cd /usr/doc/postgresql*
```

Qui potete leggere tutte le FAQ, i tutorial, e le guide per l'Utente, il Programmatore, l'Amministratore.

6.2 RPM di PostgreSQL

Vedere anche *"Installation Steps"* presso `<http://www.ramifordistat.net/postgres>`

Il curatore degli RPM di PostgreSQL è Lamar Owen, la cui e-mail è *lamar.owen@wgcr.org*. Altri dettagli su PostgreSQL si trovano nel sito `<http://www.postgresql.org>`

6.3 Maximum RPM

Per effettuare buone installazioni di PostgreSQL, familiarizzate con il gestore dei pacchetti RPM. Scaricate il testo 'Maximum RPM' dal sito `<http://www.RPM.org>`, e cercate il file di nome maximum-rpm.ps.gz Leggetelo in Linux usando il comando gv :

```
# gv maximum-rpm.ps.gz
```

Esiste anche rpm2deb, che converte i pacchetti RPM in pacchetti per Debian.

6.4 RPM degli Esempi

Gli esempi sono necessari per il collaudo delle varie interfacce al PostgreSQL. Installate la directory degli esempi di postgresql da :

- Linux cdrom - postgresql-*examples.rpm
- postgresql-*examples.rpm dal sito `<http://www.aldev.8m.com>` e dai siti mirror *webjump* `<http://aldev.webjump.com>`, *angelfire* `<http://www.angelfire.com/nv/aldev>`, *geocities* `<http://www.geocities.com/alavoor/index.html>`, *virtualave* `<http://aldev.virtualave.net>`, *bizland* `<http://aldev.bizland.com>`, *theglobe* `<http://members.theglobe.com/aldev/index.html>`, *spree* `<http://members.spree.com/technology/aldev>`, *infoseek* `<http://homepages.infoseek.com/~aldev1/index.html>`, *bcity* `<http://www3.bcity.com/aldev>`, *50megs* `<http://aldev.50megs.com>`
- Il codice sorgente di PostgreSQL è nei file postgresql*.src.rpm, e contiene le directory degli esempi, delle prove, e dei tutorial.

6.5 Provare PyGreSQL - interfaccia a Python

Installate il pacchetto degli esempi, come in6.4 (); poi digitate:

```
bash$ cd /usr/lib/pgsql/python
bash$ createdb thilo
bash$ psql thilo
thilo=> create table prova (aa char(30), bb char(30) );
thilo=> \q
bash$ /usr/bin/python
>>> import _pg
>>> db = _pg.connect('thilo', 'localhost')
>>> db.query("INSERT INTO prova VALUES ('ping', 'pong')")
>>> db.query("SELECT * FROM prova")
eins|zwei
----+----
ping|pong
(1 row)
>>>CTRL+D
bash$
..... Sembra funzionare - adesso installatelo opportunamente
bash$ su - root
#cp /usr/lib/pgsql/python/_pg.so /usr/lib/python1.5/lib-dynload
```

6.6 Provare Perl - interfaccia al Perl

Installate il pacchetto degli esempi, come in 6.4 (); poi digitate:

```
root# chown -R postgres.postgres /var/lib/pgsql/examples
bash$ cd /var/lib/pgsql/examples/perl5
bash$ perl ./example.pl
```

Nota: Se il comando precedente non funziona, fate quel che segue. La variabile globale @INC dovrebbe includere il modulo Pg.pm nella directory site_perl, e poi deve essere usata l'opzione -I

```
bash$ perl -I/usr/lib/perl5/site_perl/5.005/i386-linux-thread ./example.pl
```

.... Ora potete far girare programmi perl con accesso al database PostgreSQL!!

Leggete il file example.pl per usare l'interfaccia a Perl.

6.7 Provare le interfacce a libpq, libpq++

Installate il pacchetto degli esempi, come in6.4 (); poi digitate:

```
root# chown -R postgres.postgres /var/lib/pgsql/examples
bash$ cd /var/lib/pgsql/examples/libpq
bash$ gcc testlibpq.c -I/usr/include/pgsql -lpq
bash$ export PATH=$PATH:.
bash$ a.out

bash$ cd /var/lib/pgsql/examples/libpq++
bash$ g++ testlibpq0.cc -I/usr/include/pgsql -I/usr/include/pgsql/libpq++
```

```
-lpq++ -lpq -lcrypt
bash$ ./a.out  (Nota: Ignorate gli eventuali Messaggi di Errore, come segue)
> create table foo (aa int, bb char(4));
No tuples returned...
status = 1
Error returned: fe_setauthsvc: invalid name: , ignoring...
> insert into foo values ('4535', 'vasu');
No tuples returned...
status = 1
Error returned: fe_setauthsvc: invalid name: , ignoring...
> select * from foo;
aa   |bb   |
-----|-----|
4535 |vasu |
Query returned 1 row.
>
>CTRL+D
bash$
```

.... Ora potete far girare interfacce al database PostgreSQL, scritte in C/C++ !!

6.8 Provare Interfacce a Java

Installate il pacchetto degli esempi, come in6.4 (); installate anche quel che segue:

- Prelevate JDK jdk-*glibc*.rpm presso <ftp://ftp.redhat.com/pub/contrib/i386> o presso <http://www.blackdown.org>
- Prelevate postgresql-jdbc-*.rpm <ftp://ftp.redhat.com/pub/contrib/i386>

```
root# chown -R postgres.postgres /var/lib/pgsql/examples
bash$ cd /var/lib/pgsql/examples/jdbc
bash$ echo $CLASSPATH
 --> Dovrebbe visualizzare
CLASSPATH=/usr/lib/pgsql/jdbc7.0-1.2.jar:.:/home/java/jdk1.2.2/lib:/usr/lib/pgsql:/usr/lib/pgsql/classes.zip:/usr/lib/pgsql/pg.jar

con l'appropriato numero di versione di jdbc*.jar.
Le directory /usr/lib/pgsql e /usr/libjdk*/lib dovrebbero contenere i file *.jar.

bash$ export CLASSPATH=/usr/lib/pgsql/jdbc7.0-1.2.jar:.:/home/java/jdk1.2.2/lib:/usr/lib/pgsql:/usr/lib/pgsql/classes.zip:/usr/lib/pgsql/pg.jar

Aprite in edit il file psql.java, e commentate la linea del 'package'.
bash$ javac psql.java
bash$ java psql jdbc:postgresql:template1 postgres < password >[1] select * from pg_tables;
tablename       tableowner      hasindexes      hasrules
pg_type postgres        true    false   false
pg_attribute    postgres        true    false   false
[2]
CTRL+C
bash$
```

.... Ora fate girare interfacce Java al database PostgreSQL!

6.9 Provare l'interfaccia a ecpg

Installate il pacchetto degli esempi, vedere6.4 (); poi digitate:

```
root# chown -R postgres.postgres /var/lib/pgsql/examples
bash$ cd /var/lib/pgsql/examples/ecpg
bash$ ecpg test1.pgc -I/usr/include/pgsql
bash$ cc test1.c -I/usr/include/pgsql -lecpg -lpq -lcrypt
bash$ createdb mm
bash$ ./a.out
```

.... Adesso potete far girare Embedded "C"-SQL col database PostgreSQL!

6.10 Provare esempi di SQL - Tipi definiti dall'utente e funzioni

Installate il pacchetto degli esempi, vedere6.4 (); poi digitate:

```
root# chown -R postgres.postgres /var/lib/pgsql/examples
bash$ cd /var/lib/pgsql/examples/sql
Sezione in ampliamento..
```

6.11 Provare Interfacce Tcl/Tk

Un esempio di interfaccia Tcl/Tk è il programma pgaccess. Leggete il file /usr/bin/pgaccess utilizzando un editor:

```
bash$ view /usr/bin/pgaccess
bash$ export DISPLAY=<nome_della_vostra_macchina>:0.0
bash$ createdb miodb
bash$ pgaccess miodb
```

6.12 Provare interfacce ODBC

1. Prelevate il driver odbc pgsql per win32 presso `<http://www.insightdist.com/psqlodbc/>`
2. Vedere anche /usr/lib/libpsqlodbc.a

6.13 Provare l'interfaccia MPSQL Motif-worksheet

Prelevare gli RPM presso `<http://www.mutinybaysoftware.com>`

6.14 Verifica

Per verificare l'eccellente qualità di PostgreSQL, lanciate il pacchetto del test di Regressione.

Effettuate il login come root:

```
# rpm -i postgresql*test.rpm
E leggete il file README, o installate l'albero del codice sorgente con la directory regress
# rpm -i postgresql*.src.rpm
# cd /usr/src/redhat/SPECS
# more postgresql*.spec   (per vedere quali pacchetti RPM di sistema debbano
essere installati
# rpm -bp postgresql*.spec  (.. questo preparerà il pacchetto)

Al test della regressione servono i Makefile, e qualche file header come *fmgr*.h,
che possono essere creati con:
# rpm --short-circuit -bc postgresql*.spec ( .. usate l'opzione 'short circuit' come
scorciatoia!)
Quando leggete 'make -C common  SUBSYS.o', fermate la creazione del Makefile con CRTL+C.
Ora la configurazione è terminata con successo, e tutti i makefile e gli header
sono stati creati. Non dovete intervenire in altro modo.
# cd /usr/src/redhat/BUILD
# chown -R postgres postgresql*
# su - postgres
bash$ cd /usr/src/redhat/BUILD/postgresql-6.5.3/src/test/regress
bash$ more README
bash$ make clean; make all runtest
bash$ more regress.out
```

6.15 Correzione dei bug critici

Patch di emergenza, per la correzione dei bug critici, possono essere rilasciate dopo la release GA di PostgreSQL. Potete applicare queste patch opzionali a seconda delle necessità delle vostre applicazioni. Seguite i seguenti passi per applicare le patch: Spostatevi nella directory dei sorgenti di postgresql

```
# rpm -i postgresql*.src.rpm
# cd /usr/src/postgresql6.5.3
# man patch
# patch -p0 < file_di_patch
# make clean
# make
```

I file di patch sono posizionati presso

- Le patch di PostgreSQL: `<ftp://ftp.postgresql.org/pub/patches>`

7 Guida per un Avvio Rapido

Rifatevi anche al capitolo 6 (Installazione Rapida).

7.1 Creare, Eliminare, Rinominare Database

Potete usare l'interfaccia grafica 'pgaccess' per creare ed eliminare i database; oppure potete usare la utility a riga di comando 'psql'.

```
Se siete connessi come root, passate all'utente 'postgres':
# xhost +  (Per permettere a pgaccess l'accesso al display)
bash$ man createdb
bash$ createdb miodatabase
bash$ man psql
bash$ psql miodatabase
..... in psql, premete i tasti freccia su/giù per utilizzare lo storico dei comandi;
oppure usate il comando \s

bash$ export DISPLAY=<nomehost>:0.0
bash$ man pgaccess
bash$ pgaccess miodatabase
```

Adesso potete cominciare a **SPARARE** comandi SQL in psql o in pgaccess !!

Per eliminare il database, eseguite:

```
bash$ man dropdb
bash$ man destroydb   (per le versioni precedenti di pgsql)
bash$ dropdb <nomedb>
```

È anche possibile eliminare un database dall'interno di una sessione SQL, digitando:

```
> drop database <nomedb>
```

Per rinominare un database vedere 7.7 (Backup e Restore)

7.2 Creare, Eliminare gli utenti

Per creare nuovi utenti, fate login come utente unix 'postgres'. Potete usare l'interfaccia grafica 'pgaccess' per creare, eliminare utenti.

(n.d.t.: d'ora in poi ci riferiremo alla versione in lingua italiana di pgaccess, attivabile con la sequenza Database|Preferences|Preferred language italiano|Save; ma accanto, tra parentesi, indicheremo anche i comandi originali in lingua inglese)

```
bash$ man pgaccess
bash$ pgaccess <nome_database>
```

fate click sull'etichetta "Utenti" ("Users"), poi fate click su Oggetto|Nuovo (Object|New), oppure Oggetto|Cancella (Object|Delete)

Potete anche usare script da riga di comando. Usate lo script di shell chiamato 'createuser', che aziona psql

```
bash$ man createuser
bash$ createuser <nomeutente>
bash$ createuser -h host -p porta -i userid <nomeutente>
```

Per eliminare un utente postgres, usate lo script di shell 'destroyuser':

```
bash$ man dropuser
bash$ man destroyuser  (per precedenti versioni di pgsql)
bash$ destroyuser
```

7.3 Creare, Eliminare Gruppi

Non esiste, per ora, un'interfaccia grafica per impostare i gruppi di utenti. Dovete inserire/aggiornare esplicitamente i gruppi della tabella **pg_group**. Per esempio:

```
bash$ su - postgres
bash$ psql <nome_database>
..... in psql, premete i tasti freccia su/giù per utilizzare lo storico dei comandi;
oppure usate il comando \s

psql=> insert into pg_group (groname, grosysid, grolist)
psql=> values ('posthackers', '1234', '{5443, 8261}' );
INSERT 58224
psql=> grant insert on foo to group posthackers;
CHANGE
psql=>
```

I campi in **pg_group** sono:

groname - Il nome del gruppo. Questo nome dovrebbe essere puramente alfanumerico; non aggiungetegli caratteri di sottolineatura o altre punteggiature.

grosysid - L'ID del gruppo. Questo è un int4, e dovrebbe essere unico per ogni gruppo.

grolist - La lista degli ID degli utenti che appartengono al gruppo. Questo dato è un int4[].

Per eliminare il gruppo:

```
bash$ su - postgres
bash$ psql <nome_database>
..... in psql, premete i tasti freccia su/giù per utilizzare lo storico dei comandi;
oppure usate il comando \s

psql=> delete from pg_group where groname = 'posthackers';
```

7.4 Creare, Modificare, Eliminare una tabella

Potete usare l'interfaccia grafica 'pgaccess', oppure il tool da riga di comando 'psql', per creare, modificare o eliminare una tabella in un database.

```
bash$ man pgaccess
bash$ pgaccess <nome_database>
```

Fate click sui bottoni Tabelle | Nuovo | Disegna (Table|New|Design).

```
bash$ man psql
bash$ psql <nome_database>
..... in psql, premete i tasti freccia su/giù per utilizzare lo storico dei comandi;
oppure usate il comando \s
```

Al prompt di psql, digitate le istruzioni standard SQL per la gestione delle tabelle, come 'create table', 'alter table', o 'drop table'.

7.5 Creare, Modificare, Eliminare record in una tabella

Potete usare l'interfaccia grafica 'pgaccess', oppure il tool da riga di comando 'psql', per creare, modificare od eliminare record nella tabella di un database.

```
bash$ man pgaccess
bash$ pgaccess <nome_database>
```

Fate click sui bottoni Tabella | < clicca su una tabella > | Apri (Table|<clicca su una tabella>|Open).

```
bash$ man psql
bash$ psql <nome_database>
..... in psql, premete i tasti freccia su/giù per utilizzare lo storico dei comandi;
oppure usate il comando \s
```

Al prompt di psql, digitate le istruzioni standard SQL per la gestione delle tabelle, come 'insert into nome_tabella', 'update nome_tabella', o 'delete from nome_tabella'.

7.6 Cambiare il Database attivo

Potete usare l'interfaccia grafica 'pgaccess', oppure il tool da riga di comando 'psql', per cambiare il database attivo.

```
bash$ man pgaccess
bash$ pgaccess <nome_database>
```

Fate click sui bottoni Database | Apri (Database|Open) .

```
bash$ man psql
bash$ psql <nome_database>
..... in psql, premete i tasti freccia su/giù, per utilizzare lo storico dei comandi;
oppure usate il comando \s

psql=> connect <nome_database> <utente>
```

7.7 Fare Backup e Restore del database

PostgreSQL fornisce due programmi di utilità per salvare una copia di sicurezza (backup) del vostro sistema: **pg_dump** per fare backup di database individuali, e **pg_dumpall** per fare backup di tutto il database in un colpo solo.

```
bash$ su - postgres
bash$ man pd_dump
bash$ pd_dump <nome_database> > nome_database.pgdump

Per scaricare il dump di tutti i database:
bash$ man pg_dumpall
bash$ pg_dumpall -o > db_all.out
```

```
Per ricaricare (ripristinare) un database scaricato con pg_dump:
bash$ cat nome_database.pgdump | psql <nome_database>

Per ricaricare (ripristinare) tutti i database scaricati con pg_dump:
bash$ psql -e template1 < db_all.out
```

Questa tecnica può essere usata per spostare i database in nuove collocazioni, e per rinominare i database esistenti.

ATTENZIONE: Di ogni database si dovrebbe fare backup in maniera regolare. PostgreSQL gestisce autonomamente i suoi file, all'interno del file system. Per questo motivo, per fare i backup dei vostri database, non è consigliabile affidarsi ai soli sistemi di backup del file system; non c'è garanzia che i file saranno utilizzabili e in uno stato coerente, dopo il ripristino.

BACKUP DI DATABASE ESTESI: Poiché PostgreSQL ammette tabelle più grandi della dimensione massima dei file sul vostro file system, può essere problematico scaricare la tabella in un file, perché il file risultante sarà probabilmente più grande della dimensione massima ammissibile sul vostro sistema. Nel momento in cui **pg_dump** scrive su **stdout**, potete usare solo gli strumenti standard di unix per aggirare questo possibile problema: scaricate il database utilizzando la compressione:

```
bash$ pg_dump <nome_database> | gzip > nomefile.dump.gz
Ricaricate con:
bash$ createdb <nome_database>
bash$ gunzip -c nomefile.dump.gz | psql <nome_database>
Oppure
bash$ cat nomefile.dump.gz | gunzip | psql <nome_database>
Uso di split:
bash$ pg_dump <nome_database> | split -b 1m - nomefile.dump.
Nota: C'è un punto (.) dopo nomefile.dump, nel comando sopra riportato!!
Potete ricaricare con:
bash$ man createdb
bash$ createdb <nome_database>
bash$ cat nomefile.dump.* | pgsql <nome_database>
```

Naturalmente il nome del file (nomefile), ed il contenuto dell'output di **pg_dump**, non devono necessariamente coincidere con il nome del database. Inoltre il database ripristinato può avere un nuovo nome arbitrariamente scelto, e perciò questo meccanismo è anche adatto per rinominare i database.

Backup di Oggetti ESTESI: Gli oggetti estesi non vengono gestiti da pg_dump. La directory *contrib/pg_dumplo*, dell'albero dei sorgenti di Postgres, contiene un programma che può farlo.

BACKUP DEL FILESYSTEM: Potete usare gli strumenti e i comandi del Sistema Operativo Linux per effettuare il backup dell'intero database. Ma prima di usare questo metodo, per fare il backup o il ripristino, **dovete bloccare del tutto** il server database postgresql. Il backup o il ripristino del filesystem può essere **da 2 a 3 volte più rapido** dell'esecuzione del comando pg_dump, con il solo svantaggio che **si deve bloccare del tutto** il server database. È molto raccomandabile l'uso di strumenti di backup e ripristino come Arkeia, Bru. Questi strumenti vengono forniti nella sotto-intestazione "Backup and Restore Utility" della lista di analogie Mic-Lin presso `<http://aldev.8m.com>`, e i cui siti mirror sono: *webjump* `<http://aldev.webjump.com>`, *angelfire* `<http://www.angelfire.com/nv/aldev>`, *geocities* `<http://www.geocities.com/alavoor/index.html>`, *virtualnet* `<http://aldev.virtualave.net>`, *bizland* `<http://aldev.bizland.com>`, *theglobe* `<http://members.theglobe.com/aldev/index.html>`, *spree* `<http://members.spree.com/technology/aldev>`, *infoseek* `<http://homepages.infoseek.com/~aldev1/index.html>`, *bcity* <http://www3.bcity.com/aldev>, *50megs* <http://aldev.50megs.com> I comandi del SO da impiegare sono:

```
bash$ man tar
bash$ tar -cvf backup.tar /usr/local/pgsql/data
oppure, impiegando la compressione
bash$ tar -zcvf backup.tgz /usr/local/pgsql/data
```

BACKUP INCREMENTALE: Questa funzionalità è elencata nella lista dei todo, e apparirà nei futuri rilasci di PostgreSQL.

7.8 Protezione del database

Vedere il capitolo in 11 (Protezione di PostgreSQL).

7.9 Help in linea

È molto importante che impariate ad usare l'help in linea di PostgreSQL, poiché vi risparmierà molto tempo e vi consentirà un rapidissimo accesso alle informazioni.

Leggete le pagine di manuale in linea sui vari comandi come createdb, createuser, ecc..

```
bash$ man createdb
```

Leggete anche l'help in linea di psql, digitando \h al prompt di psql

```
bash$ psql miodatabase
psql> \h

Suggerimento: In psql, premete i tasti freccia su/giù per utilizzare lo storico dei comandi;
oppure usate il comando \s
```

7.10 Creare Trigger e Stored Procedure

Per creare trigger o stored procedure, si deve prima installare 'plpgsql' nel database particolare che state usando, eseguendo lo script 'createlang'. Se volete che il vostro database sia quello predefinito, installate 'plpgsql' nel 'template1', così i database creati saranno cloni del template1. Leggete la pagina web di 'createlang', e precisamente la guida dell'Utente presso /usr/doc/postgresql-7.0.2/user/index.html.

```
bash$ man createlang
bash$ createdb miodb
bash$ export PGLIB=/usr/lib/pgsql
bash$ createlang plpgsql miodb
bash$ createlang plpgsql template1
```

Leggete anche gli esempi di trigger e di stored procedure che trovate in 6.4 (). Ecco un frammento di codice tratto dall'RPM degli esempi:

```
create function tg_pfield_au() returns opaque as '
begin
    if new.name != old.name then
        update PSlot set pfname = new.name where pfname = old.name;
    end if;
    return new;
end;
' language 'plpgsql';

create trigger tg_pfield_au after update
    on PField for each row execute procedure tg_pfield_au();
```

Un altro esempio di trigger:

```
create trigger check_fkeys_pkey_exist
        before insert or update on fkeys
        for each row
        execute procedure
        check_primary_key ('fkey1', 'fkey2', 'pkeys', 'pkey1', 'pkey2');
```

Potete installare anche il pacchetto di TEST: postgresql-test-7.0.2-2.rpm, e potete leggere gli esempi di script sql posti in /usr/lib/pgsql/test/regress/sql

Per vedere la lista di trigger nel database, digitate:

```
bash$ psql miodb
psql=> \?
psql=> \dS
psql=> \d pg_trigger
psql=> select tgname from pg_trigger order by tgname;
```

Per leggere la lista di funzioni e stored procedure nel database, digitate:

```
bash$ psql miodb
psql=> \?
psql=> \dS
psql=> \d pg_proc
psql=> select proname, prosrc from pg_proc order by proname;
psql=> \df
```

7.11 Documentazione di PostgreSQL

Per altre domande, leggete gli eccellenti manuali di PostgreSQL, molto approfonditi. La documentazione di PostgreSQL è distribuita insieme al pacchetto. Leggete le guide 'User's Guide', 'Programmer's Guide', 'Administrator's Guide', e gli altri manuali. I documenti dei rilasci sono reperibili presso <http://www.postgresql.org/users-lounge/docs>.

8 Tuning - Come migliorare le prestazioni del server PostgreSQL

Generalmente il server database è un box autonomo connesso alla rete. Poiché il server database è il solo processo unix funzionante sulla CPU, potete fare svariate ottimizzazioni per aumentare la produttività del server.

8.1 Messa a punto del SO, per il server Database

Per ottenere più potenza elaborativa, da una data CPU, fate quanto segue:-

- Ricompilate il kernel di linux per renderlo più piccolo e snello. Eliminate le voci che non vengono utilizzate. Leggete il kernel howto situato presso `<http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html>`
- Chiudete i processi unix non necessari: eseguite chkconfig su sistemi linux/unix

```
bash$ su - root
bash# man chkconfig
bash# chkconfig --help
bash# chkconfig --list | grep on | less
Ora, dalla lista così ottenuta, mettete off i processi che volete lasciar disattivati al boot:
bash# chkconfig --level 0123456 <nome servizio> off
La prossima volta, al boot della macchina, questi servizi non verranno inizializzati.
Ora chiudete manualmente i servizi che avete disattivato.
bash# cd /etc/rc.d/init.d
bash# ./<nome servizio> stop
```

- Non fate girare altri processi applicativi che non siano necessari.
- Non lasciate girare X-Window a vuoto, inutilizzato. Infatti i processi di X-window consumano memoria, appesantiscono la CPU, e possono essere una seria falla nella sicurezza, favorendo attacchi esterni. I window manager, che X-window generalmente usa, sono: KDE, GNOME, CDE, XDM, e altri. Dovete uscire da X-window immediatamente dopo averlo utilizzato e, per la maggior parte del tempo, la macchina del server database dovrebbe presentare a video solo la console, con la linea di comando fissa sul prompt di login.

8.2 Come ottimizzare il processo del server Database

Suggerimenti di carattere generale per l'ottimizzazione:

- Gli indici possono velocizzare le interrogazioni. Il comando **explain** vi permette di verificare il modo in cui PostgreSQL sta interpretando la vostra interrogazione, e quali indici stia usando.
- Usate il comando **cluster** per raggruppare i dati nelle tabelle di base secondo la corrispondenza con un indice. Leggete la pagina di manuale **man cluster(1)** per maggiori dettagli.
- Se state effettuando un gran numero di inserimenti, considerate la possibilità di farli mediante un solo grande processo batch, usando il comando **copy**. È molto più rapido che effettuare inserimenti individuali.

- Le istruzioni non comprese in un blocco-transazione **begin work/commit**, vengono considerate come fossero in una propria transazione. Considerate la possibilità di raccogliere più istruzioni in un unico blocco-transazione. Ciò riduce l'overhead della transazione. Verificate anche la possibilità di ricreare gli indici, quando vengono effettuate vaste variazioni dei dati.
- È consigliato l'acquisto del "Performance Tuning guide", e il "tuning support" (l'assistenza all'ottimizzazione), dalla *PostgreSQL Corp.* `<http://www.postgresql.org>`.

Suggerimenti di carattere specialistico per l'ottimizzazione:

- L'ottimizzazione interna di PostgreSQL è un argomento complesso; per affrontarla serve una conoscenza profonda del codice sorgente e del funzionamento interno di postgresql. È molto raccomandabile che i seguenti suggerimenti specialistici siano provati soltanto da professionisti:
- Potete disabilitare **fsync()**, inizializzando il postmaster con l'opzione **-o -F**. Ciò impedirà ad **fsync()** di effettuare il flushing dopo ogni transazione. Tuttavia esistono rischi di perdita dei dati, dovuti a tensione elettrica mancante, o ad errori di lettura del mezzo di memorizzazione. Potete ridurre il rischio di perdita dei dati dovuti a mancanza di tensione, usando un *APC UPS* `<http://apc.com/products/ups.cfm>` (Uninterrupted Power Supply=fornitore di potenza senza interruzioni); e potete usare i sistemi di dischi RAID (*Antares-Sparc-Raid* `<http://www.linuxdoc.org/HOWTO/Antares-RAID-sparcLinux-HOWTO/index.html>` *Software-Raid* `<http://www.linuxdoc.org/HOWTO/Software-RAID-HOWTO.html>` *Old-Software-Raid* `<http://www.linuxdoc.org/HOWTO/Software-RAID-0.4x-HOWTO.html>` *Root-Raid* `<http://www.linuxdoc.org/HOWTO/Root-RAID-HOWTO.html>` *Boot-Root-Raid* `<http://www.linuxdoc.org/HOWTO/Boot+Root+Raid+LILO.html>`) come protezione dai guasti dovuti al mezzo di memorizzazione.
- Usate l'opzione **-B** di postmaster per aumentare il numero di buffer di memoria condivisa, usata dai processi del back-end. Se rendete troppo grande questo parametro, postmaster potrebbe non partire perché potreste aver superato i vostri limiti, sullo spazio di memoria condivisa, imposti dal kernel. Ogni buffer è di 8K, ed il valore predefinito è di 64 buffer.
- Usate l'opzione **-S** del back-end per incrementare l'ammontare massimo di memoria, usata da ogni processo di backend, per gli ordinamenti temporanei. Il valore **-S** è misurato in kilobyte, ed è predefinito a 512 (cioè 512K). Non è saggio rendere troppo grande questo valore, perché potreste terminare la memoria quando l'interrogazione richiede diversi ordinamenti concorrenti.

9 PostgreSQL gestisce Database Estremamente Estesi, più grandi di 200 GB

PostgreSQL è già utilizzato da molte aziende che si appoggiano a grandi database. Si suggeriscono le seguenti tecniche:

9.1 Tipi di CPU: a 32-bit o 64-bit

Quando la dimensione del database supera i 5 GigaByte, le prestazioni delle macchine con cpu a 32 bit decadranno rapidamente. Potete far girare database di 30 GB su cpu a 32 bit, ma le prestazioni saranno scadenti. Le macchine con cpu a 32 bit impongono una limitazione di 2 GB alla RAM, di 2 GB alla dimensione del file system, e altre limitazioni al sistema operativo. Usate il file system speciale per Linux, realizzato da SGI, IBM o HP oppure ext3-fs, per supportare file di dimensioni maggiori di 2 GB su macchine a 32 bit con Linux.

Per database estremamente estesi, è fortemente consigliato l'uso di macchine a 64 bit come le Digital Alpha, Sun Ultra-sparc con cpu a 64 bit, Silicon graphics con cpu a 64-bit, Intel Merced con cpu IA-64, macchine HPUX a 64bit o macchine IBM a 64-bit. Compilate PostgreSQL per cpu a 64-bit, e potrà gestire database enormi, e grandi query. Inoltre le prestazioni di PostgreSQL, per query su grandi tabelle e grandi database, saranno molto più rapide che su macchine con cpu a 32 bit. Il vantaggio di una macchina a 64 bit, è che potete avere un grandissimo spazio di indirizzamento di memoria, e il sistema operativo può supportare file system molto estesi, può fornire migliori prestazioni con grandi database, può supportare memoria (RAM) molto maggiore, ha maggiori funzionalità, ecc..

9.2 CPU Multiple

Per database estesi è consigliabile utilizzare box SMP che abbiano 4, 16 o 32 CPU. Alternativamente potete usare box con 4 o 5 CPU singole; in questo caso si deve ripartire il database in 4 o 5 database separati, ed ogni database girerà su un box separato. Ogni CPU sarà connessa con una scheda fast ethernet NIC (100MBit). Per esempio: se avete 200 tabelle in un database, potete distribuire 200 tabelle in 4 database da 50 tabelle l'uno. In tal modo, potete distribuire il carico di lavoro equamente tra 4 macchine diverse. Questa è un'alternativa economica a un box di CPU a 4 vie. Per realizzare ciò, dovreste effettuare **'Interrogazioni su database multipli'**, il montaggio di NFS nella LAN, e 'CREATE VIEW' per tabelle in sola lettura. Così ogni CPU "può vedere" tutti i database, cioè tutte le 200 tabelle. In futuro PostgreSQL potrà fornire supporto per **'Query su database multipli'**; questa funzionalità (già presente nella lista TODO), potrà apparire nell'imminente versione 7.1. Ad esempio, le query tra database multipli, che usano gli alias a e b per i nomi delle tabelle, possono apparire come segue:

```
select a.col1, a.col2, b.col4, b.col7
from
        database1.mia_tabellaa a, database2.mia_tabellab b
where
        a.col1 = b.col3 and
        a.col4 = b.col9;

update mia_tabellaa
set
        col1 =  b.col2
from
        database1.mia_tabellaa a, database2.mia_tabellab b
where
        a.col4 = b.col9;
```

9.3 Server di duplicazione

Un server di duplicazione per grandi aziende è disponibile presso <http://www.erserver.com>, e da <http://www.pgsql.com>. L'assistenza è venduta ($$$$) commercialmente da PostgreSQL Inc. Utilizzate un server di duplicazione per fornire ridondanza ed elevata disponibilità. Un server di duplicazione è un prodotto complesso e sofisticato.

10 Quanto posso aver fiducia in PostgreSQL ? Il Pacchetto del Test di Regressione crea la fiducia del cliente

Grazie alle *"Leggi della Fisica"*, è possibile verificare **SCIENTIFICAMENTE** se PostgreSQL lavora aderente alle specifiche ISO/ANSI SQL. Per confermare il valore di PostgreSQL, il pacchetto (in src/test/regress) del test della regressione è stato incluso nella distribuzione di Postgres. Il test della regressione verificherà sia le operazioni SQL standard, sia l'estendibilità delle capacità di PostgreSQL. Il pacchetto del test contiene già centinaia di programmi di test per SQL.

Per verificare la validità di PostgreSQL, dovreste usare la potenza dei computer alla loro massima velocità, piuttosto che usare la potenza del cervello umano. I computer possono eseguire i test della regressione milioni o anche miliardi di volte più celermente di quanto possano fare gli esseri umani. I computer moderni possono eseguire miliardi di test SQL in tempi brevissimi. Nel prossimo futuro la velocità dei computer sarà di diversi zilioni di volte superiore quella del cervello umano! Perciò, per confermare la validità del software, è più indicato utilizzare la potenza dei computer.

Potete realizzare altri test in caso di necessità e, se pensate che potranno essere utili ad altri in Internet, potete trasferirli sul sito web primario di PostgreSQL. Il test della regressione aiuta a consolidare la fiducia dei clienti, e facilita un rapido dispiegamento di PostgreSQL su sistemi di produzione.

Il test della regressione può essere considerato come un documento tecnico **"MOLTO SOLIDO"**, sul cui valore sono d'accordo sia sviluppatori che utenti finali. Gli sviluppatori di PostgreSQL usano in modo estensivo il test della regressione nella fase di sviluppo, ed anche prima di rilasciare il software al pubblico, allo scopo di assicurarne la buona qualità.

Il test della regressione permette di valutare accuratamente le capacità di PostgreSQL. Se una funzionalità, o una sintassi, o una caratteristica, è presente nel pacchetto del test della regressione, allora è supportata; tutte le altre, che NON sono listate nel pacchetto, potrebbero NON essere supportate da PostgreSQL!! Potreste dover fare delle verifiche in merito, e aggiungerle al pacchetto del test della regressione.

11 Protezione del Database

La protezione del Database è distribuita su diversi livelli:

- Protezione dei file del Database. Tutti i file, archiviati internamente al database, sono protetti dalla lettura di qualsiasi altro account che non sia l'account del superuser *postgres*
- Come impostazione predefinita, le connessioni da un client al server database sono ammesse solo attraverso socket UNIX locali, e non attraverso socket TCP/IT. Il back-end deve essere stato lanciato con l'opzione -i, per permettere ai client non locali di connettersi.
- Le connessioni dei client possono essere ristrette secondo l'indirizzo IP e/o lo username, mediante il file **pg_hba.conf** in **$PG_DATA**.
- Le connessioni dei client possono essere autenticate mediante uso di altri pacchetti esterni.
- Ad ogni utente di PostgreSQL è assegnato uno username e (facoltativamente) una password. Come impostazione predefinita, gli utenti non hanno accesso in scrittura su database non creati da loro.
- Gli utenti possono essere assegnati a dei gruppi, e l'accesso alle tabelle può essere ristretto in base ai privilegi del gruppo.

11.1 Autenticazione degli Utenti

L'Autenticazione è il processo con cui il server backend ed il postmaster si assicurano che il richiedente l'accesso ai dati corrisponda effettivamente all'utente di cui afferma l'identità. Tutti gli utenti che richiedono i servizi di Postgres sono messi a confronto con il contenuto della classe **pg_user**, per accertare che siano autorizzati a farlo. Tuttavia, la verifica dell'effettiva identità dell'utente è effettuata in diversi modi:

- **Dalla shell dell'utente:** Un server backend, lanciato dalla shell di un utente, conserva lo user-id dell'utente (reale) prima di effettuare un **setuid** allo user-id dell'utente **postgres**. Lo user-id reale è usato come dato fondamentale per le verifiche relative al controllo d'accesso. Nessun'altra autenticazione viene effettuata.
- **Dalla rete:** Se il sistema Postgres è stato installato direttamente dai suoi pacchetti di distribuzione, l'accesso alla porta TCP Internet, del processo postmaster, è utilizzabile da chiunque. Il DBA (Data Base Administrator = Amministratore del database) configura il file **pg_hba.conf**, situato nella directory **$PGDATA**, per specificare quale sistema di autenticazione verrà usato concordemente con l'host che effettua la connessione, e a quale database verrà connesso. Leggete **pg_hba.conf(5)** (man 5 pg_hba.conf) per consultare la descrizione dei sistemi di autenticazione disponibili. Naturalmente, l'autenticazione basata su host non è infallibile, neanche in Unix. Per ostacolare i malintenzionati risoluti è possibile anche mascherare l'host di origine. Tali questioni sulla sicurezza vanno oltre la competenza di Postgres.

11.2 Controllo d'Accesso basato su host (host-based)

Il controllo d'accesso host-based, è il nome per i controlli elementari che PostgreSQL effettua su quale parte del database sia concessa ai client, e su come gli utenti di questi client devono accreditarsi. Ogni sistema di database contiene un file chiamato **pg_hba.conf**, nella sua directory **$PGDATA**, che controlla chi può connettersi ad ogni database. Ogni client che accede ad un database, deve essere citato nel gruppo degli iscritti in **pg_hba.conf**. Altrimenti tutti i tentativi di connessione da quel client saranno respinti, e daranno il messaggio d'errore **"User authentication failed"**.

Leggete le pagine di manuale in linea di **pg_hba.conf(5)** (man 5 pg_hba.conf).

Il formato generale del file **pg_hba.conf**, è costituito da un insieme di record, uno per ogni riga. Le righe vuote, e quelle che iniziano con il carattere "#", sono ignorate. Un record è composto da un gruppo di campi separati da spazi e/o tabulazioni.

I client possono effettuare le connessioni usando socket di dominio Unix, o socket di dominio Internet (ad esempio TCP/IP). Le connessioni, effettuate mediante socket di dominio Unix, sono controllate usando record del seguente formato:

```
database_locale metodo_di_autenticazione
```

dove

database_locale specifica il database cui è riferito il record. Il valore **all** specifica che quanto segue verrà applicato a tutti i database.

metodo_di_autenticazione specifica il metodo che un utente deve usare per autenticare sé stesso, quando si connette a quel database usando socket di dominio Unix. I diversi metodi sono descritti più avanti.

Le connessioni, effettuate usando i socket di dominio Internet, sono controllate usando record con il seguente formato.

```
host database indirizzo-TCP/IP maschera-TCP/IP metodo_di_autenticazione
```

L'indirizzo **TCP/IP** viene *moltiplicato logicamente* (mediante l'operatore binario AND) sia alla maschera TCP/IP specificata, sia all'indirizzo TCP/IP del client connesso. Se i due valori risultanti sono uguali, il record viene usato per la connessione. Se ad una connessione corrispondessero più record validi, viene usato il primo che appare nel file. Sia l'indirizzo TCP/IP, sia la maschera TCP/IP, sono espressi nella notazione decimale puntata. Se per una connessione non viene trovato un record corrispondente, tra i metodi di autenticazione viene applicato il 'reject' (vedere 11.3 (Metodi di Autenticazione)).

11.3 Metodi di Autenticazione

I seguenti metodi di autenticazione sono supportati sia da Unix che dai socket di dominio TCP/IP:

- **trust** La connessione è permessa incondizionatamente.
- **reject** La connessione è rifiutata incondizionatamente.
- **crypt** Al client viene chiesta una password per l'utente. Questa viene inviata cifrata (usando crypt(3)), e confrontata con la password contenuta nella tabella pg_shadow. Se le password sono coincidenti, la connessione viene permessa.
- **password** Al client viene chiesta una password per l'utente. Questa viene inviata in chiaro e confrontata con la password contenuta nella tabella **pg_shadow**. Se le password sono coincidenti, la connessione viene permessa. Dopo la parola chiave 'password' può essere specificato un file di password facoltativo, da usarsi in alternativa alla tabella **pg_shadow** per il confronto delle password fornite. Vedere **pg_passwd**.

I seguenti metodi di autenticazione sono supportati soltanto per i socket di dominio TCP/IP:

- **krb4** Kerberos V4 è usato per autenticare l'utente.
- **krb5** Kerberos V5 è usato per autenticare l'utente.
- **ident** Il server ident è usato sul client per autenticare l'utente (RFC 1413). Un nome di mappa opzionale può essere specificato dopo la parola chiave **ident**; questo nome permette ai nomi utente ident di essere mappati tra i nomi degli utenti di Postgres. Le mappe sono conservate nel file **$PGDATA/pg_ident.conf**.

Ecco alcuni esempi:

```
# Ci fidiamo di qualsiasi connessione via socket di dominio Unix.
local   trust
# Ci fidiamo di qualsiasi connessione via TCP/IP da questa macchina.
host    all 127.0.0.1       255.255.255.255     trust
# Questa macchina non è desiderata.
host    all 192.168.0.10    255.255.255.0       reject
# Questa macchina non può cifrare, quindi le chiediamo le password in chiaro.
host    all 192.168.0.3     255.255.255.0       password
# Il resto di questo gruppo di macchine dovrebbe fornire password cifrate.
host    all 192.168.0.0     255.255.255.0       crypt
```

11.4 Controllo dell'Accesso

Postgres fornisce meccanismi per consentire agli utenti, che mettono i loro dati a disposizione di altri utenti, di limitarne l'accesso.

- **Superutenti del Database** I super-utenti del Database (cioè gli utenti che hanno impostato **pg_user.usesuper**) saltano tranquillamente tutti i controlli d'accesso descritti più avanti, con due eccezioni: gli aggiornamenti manuali del catalogo di sistema non sono permessi, se l'utente non ha impostato **pg_user.usecatupd**; inoltre la cancellazione dei cataloghi dei sistemi (o le modifiche dei loro schemi) non è mai concessa.
- **Privilegio d'Accesso** L'uso del privilegio d'accesso per limitare lettura, scrittura e impostazione di regole sulle classi, è previsto nei comandi SQL **grant/revoke(l)**.
- **Soppressione di classe e modifiche di schema** I comandi per cancellare o modificare le strutture di una classe esistente, come alter, drop table, e drop index, hanno effetto solo per il proprietario della classe. Come è già stato precedentemente segnalato, queste operazioni non sono mai consentite sui cataloghi di sistema.

11.5 Connessioni TCP/IP Protette mediante SSH

Potete usare **ssh** per cifrare la connessione di rete tra i client e il server Postgres. Se fatto correttamente, ciò dovrebbe portare ad una connessione di rete adeguatamente sicura.

La documentazione per **ssh** fornisce la maggior parte delle informazioni per cominciare ad operare. Riferitevi all'url `<http://www.heimhardt.de/htdocs/ssh.html>` per approfondire. La realizzazione di una connessione con SSL può essere fatta in sole due fasi.

Lanciare un tunnel sicuro mediante ssh: La realizzazione di una connessione con SSL può essere fatta in sole due fasi.

- Collegatevi con un tunnel, alla macchina back-end, come segue:

```
ssh -L 3333:wit.mcs.anl.gov:5432 postgres@wit.mcs.anl.gov
```

- Il numero **3333**, posto subito dopo l'argomento **-L**, è il numero della porta della vostra estremità del tunnel. Il secondo numero, **5432**, è l'altra estremità remota del tunnel (il numero di porta che sta usando il vostro backend). Il nome o l'indirizzo, posti tra i numeri delle porte, appartengono al server della macchina; questo vale anche per l'ultimo argomento di **ssh**, che comprende anche il nome dell'utente opzionale. Senza il nome dell'utente, **ssh** proverà con il nome col quale siete al momento connessi sulla macchina client. Potete usare qualsiasi nome-utente accettabile per la macchina server, e questo nome-utente non è necessariamente correlato a postgres.
- Con una sessione **ssh** attiva potete connettervi, con un client postgres, al vostro host locale, sul numero di porta da voi specificato nella fase precedente. Se il client è **psql**, avrete bisogno di un'altra shell; infatti la sessione di shell da voi usata, nella fase 1, è ora occupata da **ssh**.

```
psql -h localhost -p 3333 -d mpw
```

- Notate che avete specificato l'argomento **-h** per forzare il client ad usare il socket TCP/IP, invece del socket Unix. Potete omettere l'argomento relativo alla porta, se scegliete **5432** come quella relativa alla vostra estremità del tunnel.

11.6 Autenticazione Kerberos

Kerberos è un sistema standard di autenticazione sicura; è diffuso tra le aziende, ed è adatto a reti pubbliche per il calcolo distribuito.

Disponibilità: Il sistema di autenticazione Kerberos non viene distribuito insieme a Postgres. Le versioni di Kerberos sono solitamente rese disponibili, come software opzionale, dal rivenditore del sistema operativo. In aggiunta, una distribuzione del codice sorgente può essere ottenuta direttamente dal Progetto Athena del MIT.

```
Nota: Potreste voler ottenere la versione del MIT anche se il vostro rivenditore ne fornisce
una sua versione; infatti le versioni di qualche rivenditore possono essere state storpiate o rese
non-interoperanti con la versione del MIT.
```

Le richieste di informazioni, riguardanti il 'vostro' Kerberos, dovrebbero essere rivolte al vostro rivenditore, o al Progetto Athena del MIT. Notate che le FAQL (Frequently-Asked Questions Lists) sono periodicamente pubblicate sulla mailing list di Kerberos (mandate una mail per sottoscrivervi), e sui news group USENET.

Installazione: L'installazione di Kerberos è descritta in dettaglio nelle 'Kerberos Installation Notes'. Assicuratevi che il file chiave del server (**srvtab** o **keytab**) sia leggibile, in un modo o nell'altro, dall'account di Postgres. Postgres, ed i suoi client, possono essere compilati per usare o la Version 4, o la Version 5 dei protocolli Kerberos MIT; per farlo, si imposta la variabile KRBVERS, nel file **src/Makefile.global**, al valore corretto. Potete anche cambiare la collocazione in cui Postgres si aspetta di trovare le librerie associate, i file header e il file chiave del server. A compilazione completata, Postgres deve essere registrato come un servizio di Kerberos. Leggete le 'Kerberos Operations Notes', e le relative pagine di manuale, per avere maggiori dettagli sulla registrazione dei servizi.

Funzionamento: Dopo le operazioni di installazione, Postgres dovrebbe operare in tutti i modi consueti per un normale servizio Kerberos. Per i dettagli sull'uso dell'autenticazione, leggete la *PostgreSQL User's Guide*, e precisamente le sezioni relative a **postmaster** e **psql**. Nella modalità di connessione della Version 5 di Kerberos, sono state adottate le seguenti convenzioni sulla nomenclatura degli utenti e dei servizi (vedere anche la tabella più sotto):

- Si assume che, tra i nomi principali dell'utente (i cosiddetti 'aname'), sia compreso l'effettivo nome utente di Unix/Postgres come primo componente.
- Si assume che il servizio Postgres abbia due componenti: il nome del servizio ed un nome di host, rispettando un uso consolidato dalla Version 4 (ciò significa che tutti i suffissi di dominio sono soppressi).

```
               Tabella: Esempi di Parametri di Kerberos
 -----------------------------------------------------
 Parametro     Esempio
 -----------------------------------------------------
 user          frew@S2K.ORG
 user          aoki/HOST=miyu.S2K.Berkeley.EDU@S2K.ORG
 host          postgres_dbms/ucbvax@S2K.ORG
 -----------------------------------------------------
```

---

12 Strumenti GUI per PostgreSQL (Interfacce-utente grafiche)

Il browser Web sarà l'interfaccia grafica più utilizzata del futuro. È opportuno che convertiate tutte le vostre applicazioni, già funzionanti in Windows 95/NT, in applicazioni basate su Web.

Dovreste usare Server Web-Application come 18.3 () (basato su Java) o 18.4 () (basato su Python) o 18.5 ().

Il miglior linguaggio di web-scripting (anche compilabile) è il 22 (PHP+Compilatore Zend). **PHP** è estremamente potente, perché combina la potenza di Perl, Java, C++, e Javascript, in un unico linguaggio che gira su tutti i Sistemi Operativi: Unix-like e Windows NT/95.

I migliori strumenti, in ordine di preferenza, sono:

- 18.3 (), più il Java di Borland, JBuilder per Linux `<http://www.inprise.com>`
- 18.4 ()
- 18.5 ()
- PHP script e compilatore Zend presso 22 (PHP+Compilatore Zend)
- X-Designer supporta C++, Java e MFC `<http://www.ist.co.uk/xd>`
- Qt per Windows95 e Unix presso `<http://www.troll.no>` e `<ftp://ftp.troll.no>`
- Code Crusader è sul cdrom di Linux; è freeware, ed basato sul Code Warrior della MetroWorks `<http://www.kaze.stetson.edu/cdevel/code_crusader/about.html>`
- Code Warrior della MetroWorks `<http://www.metrowerks.com>`
- GNU Prof C++ IDE da *(Redhat)* `<http://www.redhat.com>` Cygnus `<http://www.cygnus.com>`
- Borland C++ Builder per Linux `<http://www.inprise.com>`
- Borland Java JBuilder per Linux `<http://www.inprise.com>`

Linguaggi scelti in ordine di preferenza sono:

1. Java, ma i suoi programmi sono molto lenti, e si paga la licenza. C++ è **5 volte più veloce** di Java!!
2. Python (Potente liguaggio di scripting orientato agli oggetti).
3. Scripting PHP per server Web, HTML, DHTML, con scripting Javascript per client, e le Applet-Java.
4. Linguaggio di scripting Perl, utilizzando Perl-Qt o Perl-Tk 14 ()
5. L'Onnipresente ed Onnipotente linguaggio C++ (GNU g++):
    - Fast CGI(scritto in GNU C++) con Javascript/Java-Applet come frontend-GUI-Web.
    - GNU C++ e QtEZ o QT
    - GNU C++ con Lesstiff o Motif.

Ci sono altri strumenti a disposizione: Nella distribuzione chiamata 'pgTcl', c'è una libreria Tcl/Tk di interfaccia a PostgreSQL. Esiste un IDE (integrated development environment) per Tcl/Tk, chiamato SpecTcl.

- Lesstiff Motif tool `<ftp://ftp.redhat.com/pub/contrib/i386/lesstiff*.rpm>`
- Vibe Java/C++, presso il sito `<http://www.LinuxMall.com/products/00487.html>`
- JccWarrior `<ftp://ftp.redhat.com/pub/contrib/i386/jcc*.rpm>`
- Tcl/Tk `<http://www.scriptics.com>`
- Estensione di Tcl orientata agli oggetti, chiamata INCR, al sito `<http://www.tcltk.com>`

- sito di Visual TCL <http://www.neuron.com>
- rpm RedHat di Visual TCL, presso <ftp://ftp.redhat.com/pub/contrib/i386/visualtcl*.rpm>
- <http://sunscript.sun.com/>
- <http://sunscript.sun.com/TclTkCore/>
- <ftp://ftp.sunlabs.com/pub/tcl/tcl8.0a2.tar.Z>
- Java FreeBuilder <ftp://ftp.redhat.com/pub/contrib/i386/free*.rpm>
- SpecTCL <ftp://ftp.redhat.com/pub/contrib/i386/spec*.rpm>
- Java RAD Tool per PostgreSQL Kanchenjunga <http://www.man.ac.uk/~whaley/kj/kanch.html>
- Applixware Tool <http://www.redhat.com>
- XWPE X Windows Programming Environment <http://www.identicalsoftware.com/xwpe/> o presso <http://www.rpi.edu/~payned/xwpe/> <ftp://ftp.redhat.com/pub/contrib/i386/xwpe*.rpm>
- XWB X Windows Work Bench <ftp://ftp.redhat.com/pub/contrib/i386/xwb*.rpm>
- NEdit <ftp://ftp.redhat.com/pub/contrib/i386/nedit*.rpm>

Potete anche utilizzare Borland C++ Builder, Delphi, Borland JBuilder, PowerBuilder per Windows95, connettendovi ad uno unix box con PostgreSQL, mediante i driver ODBC/JDBC.

13 Driver di interfaccia per PostgreSQL

13.1 Driver ODBC per PostgreSQL

ODBC sta per 'Open DataBase Connectivity'; è uno standard diffuso, istituito da Microsoft, utilizzato per accedere alle informazioni contenute in vari database di diversi produttori. Le applicazioni scritte usando i driver ODBC, funzionano con vari database come PostgreSQL, Oracle, Sybase, Informix ecc..

- *PostODBC* <http://www.insightdist.com/psqlodbc> è già incluso nella distribuzione. Visitate il sito web principale <http://www.postgresql.org>. È incluso nel CDROM di PostgreSQL.
- Il codice del progetto Open source ODBC, è al sito <http://www.iodbc.org>
- <http://www.openlinksw.com> Open Link Software Corporation vende ODBC per PostgreSQL ed altri database. Open Link, inoltre, mette a disposizione ODBC gratuiti (in numero limitato); verificatelo.
- Insight ODBC per PostgreSQL <http://www.insightdist.com/psqlodbc>. Questo è il sito ufficiale di PostODBC.
- Pacchetto FreeODBC <http://www.ids.net/~bjepson/freeODBC/>. Questa è una versione gratuita di ODBC.
- ODBC 32 Explorer per PostgreSQL <http://members.nbci.com/anhr>.

13.2 Driver UDBC per PostgreSQL

UDBC è una versione statica di ODBC, indipendente dai gestori dei driver e dal supporto DLL; è usata per incorporare il supporto della connessione ai database direttamente nelle applicazioni.

- <http://www.openlinksw.com> Open Link Software Corporation vende UDBC per PostgreSQL ed altri database. Open Link, inoltre, mette a disposizione UDBC gratuiti (in numero limitato); verificatelo.

13.3 Driver JDBC per PostgreSQL

JDBC sta per 'Java DataBase Connectivity'. Java è un linguaggio indipendente dalla piattaforma, sviluppato da Sun Microsystems. I programmatori Java sono incoraggiati a scrivere applicazioni per database usando il JDBC, allo scopo di agevolare la portabilità tra i database come PostgreSQL, Oracle, informix, ecc. Se scrivete applicazioni Java, potete procurarvi i driver per PostgreSQL presso i seguenti siti: (il driver per JDBC è già incluso nella distribuzione di PostgreSQL, è in postgresql-jdbc*.rpm)

- <http://www.demon.co.uk/finder/postgres/index.html> connettività di PostgreSQL al Java di Sun
- <ftp://ftp.ai.mit.edu/people/rst/rst-jdbc.tar.gz>
- <http://www.openlinksw.com> Open Link Software Corporation vende JDBC per PostgreSQL ed altri database. Open Link, inoltre, mette a disposizione JDBC gratuiti (in numero limitato); verificatelo.
- sito UK sul JDBC <http://www.retep.org.uk/postgres>
- sito con FAQ sul JDBC <http://eagle.eku.edu/tools/jdbc/faq.html>

l'home page, la guida, e le FAQ di JDBC, si trovano presso:

- JDBC HOME <http://splash.javasoft.com/jdbc>
- JDBC guide <http://www.javasoft.com/products/jdk/1.1/docs/guide/jdbc>
- JDBC FAQ <http://javanese.yoyoweb.com/JDBC/FAQ.txt>

Vedere la sezione: 6.8 (Provare l'interfaccia di Java a PostgreSQL)

13.4 Java per PostgreSQL

I programmatori Java troveranno molto utili queste risorse per PostgreSQL.

- <ftp://ftp.redhat.com/pub/contrib/i386>, e cercate postgresql-jdbc-*.rpm
- <http://www.blackdown.org>

Vedere la sezione: 6.8 (Provare l'interfaccia di Java a PostgreSQL)

14 Driver d'interfaccia ai database di Perl, per PostgreSQL (DBI)

14.1 Interfaccia di Perl a PostgreSQL

PERL è l'acronimo di 'Practical Extraction and Report Language'. Perl è disponibile su tutti i sistemi operativi, e per tutte le piattaforme hardware del mondo. Potete usare Perl su: Windows95/NT, Apple Macintosh iMac, tutte le versioni di Unix (Solaris, HPUX, AIX, Linux, Irix, SCO ecc..), mainframe MVS, desktop OS/2, OS/400, Amdahl UTS, e molti altri. Perl gira **ANCHE** su molti sistemi operativi e hardware poco diffusi, o generalmente sconosciuti!! Così non sorprendetevi se vedete perl girare su sistemi operativi usati molto raramente. Potete immaginare quanto estesa sia la base di utenti e di sviluppatori del Perl. Il linguaggio Perl ha, come il linguaggio "C", una vita molto lunga, e sarà usato per **migliaia di anni** nel futuro! Perl gira **10 volte** più rapidamente di Java, e qualche volta è più rapido anche del "C". Java è un sistema molto complesso; la sua macchina virtuale, e l'interprete, lo rendono estremamente lento, instabile ed inaffidabile. Perl è molto semplice, rapido, ed è orientato agli oggetti.

L'interfaccia di Perl a PostgreSQL è inclusa nella distribuzione di PostgreSQL. Cercate nella directory src/pgsql_perl5.

- Per Pgsql_perl5, contattate l'Email: *E.Mergl@bawue.de*
- Home page di Perl: `<http://www.perl.com/perl/index.html>`
- Tutorial su Perl, cercate il link del Tutorial presso: `<http://reference.perl.com/>`
- FAQ su Perl sono al sito: `<http://www.yahoo.com/Computers_and_Internet/Programming_Languages/Perl/>`
- Anzitutto procuratevi la Madre di tutti i Moduli Perl, presso il sito `<http://www.perl.com/CPAN/modules/by-module/CPAN>` ; digitate '/usr/bin/cpan', 'man CPAN', e troverete migliaia di moduli `<http://www.perl.com/CPAN-local/modules/by-module>`.
- Rpm della GUI (Interfaccia Grafica) Perl-Qt: `<ftp://ftp.redhat.com/pub/contrib/i386>` , e cercate PerlQt-1.06-1.i386.rpm
- Perl-Qt, GUI (Interfaccia grafica) Perl: `<http://www.accessone.com/~jql/perlqt.html>`
- Perl-XForms, GUI (Interfaccia grafica) Perl: `<ftp://ftp.redhat.com/pub/contrib/i386>`, e cercate Xforms4Perl-0.8.4-1.i386.rpm
- Perl-Tk GUI (Interfaccia grafica) Perl: `<ftp://ftp.redhat.com/pub/contrib/i386>`
- Kit per GUI di Perl: `<http://reference.perl.com/query.cgi?ui>`
- Interfacce di Perl ai Database: `<http://reference.perl.com/query.cgi?database>`
- Traduttore da Perl a "C": `<http://www.perl.com/CPAN-local/modules/by-module/B/>` , e cercate Compiler-a3.tar.gz
- Compilazione di Perl in un eseguibile. Perl2Exe è una utility a riga di comando per convertire script perl in file eseguibili `<http://www.indigostar.com/perl2exe.htm>`
- Traduttore da Bourne shell a Perl: `<http://www.perl.com/CPAN/authors/id/MERLYN/sh2perl-0.02.tar.gz>`
- I traduttori da awk a Perl a2p, e da sed a Perl s2p, sono inclusi nella distribuzione di PERL.
- Cercate anche nei newsgroup comp.lang.perl.*, specifici per PERL.

14.2 Interfaccia DBI di Perl ai Database

14.2.1 COS'È DBI ?

DBI, l'Interfaccia di Perl ai Database, è una API (Application Programming Interface) d'accesso ai database per il linguaggio Perl. Le specifiche delle API DBI di Perl, definiscono un insieme coerente di funzioni, variabili e convenzioni, che forniscono un'interfaccia ai database indipendente dal database utilizzato. L'iniziativa dei Driver Database (Perl DBI) ha standardizzato l'interfaccia di numerosi motori database commerciali; cosicché potete, ad esempio, passare da Oracle a PostgreSQL con poca fatica.

14.2.2 Driver DBI per PostgreSQL

Prima di installare DBD PostgreSQL (Driver), dovete installare DBI. Prelevate il driver DBI:

- Anzitutto procuratevi la Madre di tutti i Moduli Perl presso il sito: `<http://www.perl.com/CPAN/modules/by-module/CPAN>` , e digitate '/usr/bin/cpan', 'man CPAN'; qui troverete migliaia di moduli `<http://www.perl.com/CPAN-local/modules/by-module>`.
- Moduli DBI `<http://www.perl.com/CPAN-local/modules/by-module/DBI>`.
- Moduli DBI `<http://www.symbolstone.org/technology/perl/DBI>`
- FAQ su DBI `<http://www.symbolstone.org/technology/perl/DBI/doc/faq.html>`
- Documentazione di riferimento per Perl DBI `<http://www.symbolstone.org/technology/perl/DBI>`
- Mailing List su DBI `<http://www.fugue.com/dbi>`
- Documentazione di riferimento per uso dei Database in Perl `<http://www.perl.com/reference/query.cgi?section=database>`
- Prelevamento di DBI in formato rpm (Attenzione: può essere una vecchia versione) `<http://rpmfind.net/linux/rpm2html/search.php?query=DBI>`

Prelevate DBD-Pg come segue

- Anzitutto procuratevi la Madre di tutti i Moduli Perl, presso il sito `<http://www.perl.com/CPAN/modules/by-module/CPAN>` , e digitate '/usr/bin/cpan', 'man CPAN'; qui troverete migliaia di moduli `<http://www.perl.com/CPAN-local/modules/by-module>`.
- Moduli DBD `<http://www.perl.com/CPAN-local/modules/by-module/DBD>`. e cercate i file DBD-pg, oppure *DBD* `<http://www.perl.com/CPAN/modules/by-module/DBD>`.
- CPAN, l'esauriente Archivio di rete dedicato a Perl `<http://www.perl.com/CPAN>`. Giunti su questo sito, selezionate 'Database' (sulla casella di ricerca superiore), e fate click sul tasto 'Go'.
- Il pacchetto pre-compilato per Windows NT/2000, è disponibile in `<http://www.edmund-mergl.de/export/DBD-Pg.zip>`.
- Prelevamento di DBD in formato rpm (Attenzione: può essere una vecchia versione) `<http://rpmfind.net/linux/rpm2html/search.php?query=DBD>`
- Moduli Perl (ce ne sono migliaia) `<http://www.perl.com/CPAN-local/modules/by-module>`.

14.2.3 Supporto tecnico per DBI

- Inviate commenti e segnalazioni di bug, includendo l’output di perl -v e perl -V, la versione di PostgreSQL, la versione di DBD-Pg, e la versione di DBI nelle vostre segnalazioni. *E.Mergl@bawue.de*

14.2.4 Documenti sulla DBI

Ci sono alcune fonti di informazioni sulla DBI.

Documentazione POD: I POD sono grandi raccolte di documentazione, solitamente incluse nei programmi perl; esse documentano il codice “sul posto”, fornendo un’utile risorsa per i programmatori e gli utenti dei moduli. I POD, per DBI e driver, stanno diventando sempre più una consuetudine, e la documentazione per questi moduli può essere letta con i seguenti comandi:

```
I POD per le specifiche DBI possono essere letti con il comando
        $ perldoc DBI

Gli utenti dello strato di emulazione Oraper, incluso nel DBD::Oracle, possono
leggere come programmare con l’interfaccia Oraperl, digitando:
        $ perldoc Oraperl

Gli utenti del modulo DBD::mSQL possono leggere le descrizione di alcune delle funzioni
riservate, e delle particolarità del driver, digitando:
        $ perldoc DBD::mSQL

Le FAQ (Frequently Asked Questions) sono anche disponibili come
documentazione POD. Leggetele digitando:
        $ perldoc DBI::FAQ

POD in generale: Le informazioni su come scrivere POD, e sulla filosofia dei POD in
generale, possono essere lette digitando:
        $ perldoc perlpod
```

Gli utenti, con il modulo Tk installato, possono essere interessati a sapere che è disponibile un lettore di POD, chiamato tkpod, basato su Tk; esso formatta i documenti POD in un modo da renderli facilmente leggibili.

Vedere anche:

- Informazioni dalla mailing list sulla DBI `<http://www.symbolstone.org/technology/perl/DBI/tidbits>`
- Sito web del DBI Perl Journal `<http://www.tpj.com>`
- “DBperl”, articolo pubblicato nella edizione di Novembre 1996 del “Dr. Dobbs Journal”.
- “The Perl5 Database Interface”, un libro scritto da Alligator Descartes e pubblicato da O’Reilly and Associates.

Le mailing list, cui gli utenti possono partecipare, sono:

- Mailing list `<http://www.fugue.com/dbi>`
- **dbi-announce** Email: *dbi-announce-request@fugue.com* , con ’subscribe’ nel corpo del messaggio.

- **dbi-dev** Per sviluppatori: Email: *dbi-dev-request@fugue.com* , con 'subscribe' nel corpo del messaggio
- **dbi-users** discussioni generali Email: *dbi-users-request@fugue.com* , con 'subscribe' nel corpo del messaggio
- Archivi della Mailing List USA `<http://outside.organic.com/mail-archives/dbi-users/>`
- Archivi della Mailing List Europea `<http://www.rosat.mpe-garching.mpg.de/mailing-lists/PerlDB-Interest>`

14.2.5 DBI è supportata sotto le piattaforme Windows 95 / NT ?

Ora, le versioni per Win32 della DBI, e di DBD::Oracle, sono una parte standard della DBI e quindi, scaricando una versione della DBI maggiore della 0.81, dovrebbero funzionare bene. Dalla DBI, attraverso ODBC, potete accedere ai database Microsoft Access e SQL-Server. Esiste uno 'strato di emulazione' per il modulo Win32::ODBC. È chiamato DBI::W32ODBC, ed è Fornito con DBI-0.79 (o più recente), Vi servirà il modulo Win32::ODBC.

- Win32 DBI `<http://www.symbolstone.org/technology/perl/DBI>`
- Win32 ODBC `<http://www.roth.net>`
- Interfaccia Perl a Microsoft SQL server `<http://www.algonet.se/~sommar/mssql>`

14.2.6 Supporto Commerciale e Addestramento

PERL CLINIC: Perl Clinic può stringere accordi di supporto commerciale per Perl, DBI, DBD::Oracle e Oraperl. Il supporto è fornito dalla compagnia in cui lavora Tim Bunce, autore della DBI. Per maggiori informazioni sui loro servizi, vedere:

- Assistenza `<http://www.perlclinic.com>`
- Assistenza `<http://www.perldirect.com>`
- Addestramento `<http://www.westlake.com/training>`

14.3 Provare l'interfaccia al Perl

Vedere la sezione: 6.6 (Provare l'interfaccia di Perl a PostgreSQL)

15 Strumenti di Gestione di PostgreSQL

15.1 PGACCESS - Uno strumento Grafico (GUI) per la gestione di PostgreSQL

PgAccess è un'interfaccia in Tcl/Tk a PostgreSQL. È già inclusa nella distribuzione di PostgreSQL. Potreste voler verificare la presenza di una nuova versione sul seguente sito:

- `<http://www.flex.ro/pgaccess>`
- Se volete fare dei commenti, o suggerire dei miglioramenti, mandate un'e-mail a: *teo@flex.ro*

Uso di pgaccess:

```
# man xhost
# xhost +
# su - postgres
bash$ man pgaccess
bash$ export DISPLAY=<nomehost>:0.0
bash$ pgaccess miodatabase
```

Caratteristiche di PgAccess

Finestre di PgAccess: Finestra Principale, Crea nuova Tabella (Table Builder), Visualizzazione della Tabella (e della query), Costruttore visuale della query (Visual query designer).

Tabelle

- apertura tabelle in visualizzazione, per un massimo di 200 record (limite modificabile tramite menu Preferenze)
- ridimensionamento delle colonne, mediante il trascinamento della riga verticale della griglia (più facile da effettuare in spazio tabella, piuttosto che nella sua intestazione)
- A-capo automatico nelle celle; memorizzazione del layout per ogni tabella
- importazione/esportazione da e verso file esterni (SDF,CSV)
- uso di condizioni di filtro (inserite filtri come (prezzo>3.14))
- possibilità di ordinamenti (inserite manualmente il/i campo/i per l'ordinamento)
- possibilità di inserimento/modifica direttamente nelle celle
- l'assistente generatore di tabella è stato perfezionato
- l'editing del campo è stato migliorato

Interrogazioni

- definizione, modifica, e archiviazione di "interrogazioni definite dall'utente"
- archiviazione di interrogazioni come viste
- esecuzione di interrogazioni
- visualizzazione del risultato di interrogazioni di tipo select
- cancellazione e rinomina di query
- Funzionalità di drag & drop di Visual query builder. Per chiunque di voi abbia installato il plugin Tcl/Tk per Netscape Navigator, potete vederlo al lavoro facendo click qui `<file:/usr/doc/postgresql-tcl-6.5.3/qbtclet.html>`

Sequenze

- definizione, cancellazione e verifica di sequenze

Funzioni

- definizione, verifica e cancellazione di funzioni in linguaggio SQL

Le implementazioni future avranno

- progettazione di tabelle (aggiunta di nuovi campi, rinomina, ecc.)
- definizione di funzione
- generatore di report
- un elementare linguaggio di scripting

INFORMAZIONI SU LIBGTCL

Sarà necessaria la libreria libgtcl, che funge da interfaccia del Tcl al PostgreSQL; questa libreria è uno dei moduli caricabili del Tcl/Tk. La libreria e il sorgente sono situati nella directory /src/interfaces/libpgtcl di PostgreSQL. In particolare, vi servirà una libreria libpgtcl che sia caricabile da Tcl/Tk. Ciò è tecnicamente diverso dal tipico file oggetto caricabile di PostgreSQL, perché libpgtcl è una collezione di file oggetto. Sotto Linux questa libreria è denominata libpgtcl.so. Potete prelevare, dal sito sopra indicato, una versione già compilata per sistemi Linux su i386. È sufficiente copiare libpgtcl.so nella vostra directory delle librerie di sistema (/usr/lib). Una delle soluzioni possibili è quella di rimuovere, dal codice sorgente, la linea contenente "load libpgtcl.so", e caricare pgaccess.tcl non con wish, ma con pgwish (o wishpg), cioè quel wish che è stato linkato con la libreria libpgtcl.

Se, durante la compilazione dei sorgenti di pgaccess, crypt non dovesse funzionare, usate l'opzione -lcrypt.

15.2 GtkSQL, Strumento Grafico di Query per PostgreSQL

GtkSQL è uno strumento grafico per effettuare query (come il psql di PostgreSQL). È stato rilasciato sotto la licenza GNU GPL, ed è stato sviluppato utilizzando Gtk+ 1.2.3 e PostgreSQL 6.3. Il sito principale di GtkSQL è `<http://gtksql.sourceforge.net>`

Le sue principali funzionalità sono:

1. buffer SQL multipli
2. parole chiave SQL, completamento automatico dei nomi di tabelle e di campi
3. visualizzazione confortevole della definizione delle tabelle
4. supporto per PostgreSQL e MySQL (e si può aggiungere facilmente anche il supporto per altri database)

La versione attuale è GtkSQL v. 0.3. Potete reperire il sorgente presso `<https://sourceforge.net/project/?form_grp=533>`

15.3 Strumento interattivo di Query in Windows per PostgreSQL (WISQL o MPSQL)

MPSQL fornisce gli utenti di un interfaccia SQL grafica a PostgreSQL. MPSQL è simile a SQL Worksheet di Oracle, e allo strumento di interrogazione WISQL di Microsoft SQL Server. Ha una gradevole GUI, e lo storico dei comandi. È permesso anche il "copia e incolla", ed ha altre pregevoli funzionalità per migliorare la vostra produttività.

- `<http://www.troubador.com/~keidav/index.html>`
- Email: *keidav@whidbey.com*

- <http://www.ucolick.org/~de/> nel file tcl_syb/wisql.html
- <http://www.troubador.com/~keidav/index.html>
- Email: *de@ucolick.org*

15.4 Strumento Interattivo di Query (ISQL) per PostgreSQL, denominato PSQL

ISQL funziona su terminali a riga di comando. È incluso nella distribuzione, ed è chiamato "psql". È molto simile a Sybase ISQL, Oracle SQLplus. Dando il comando 'psql', dal prompt di unix passerete al prompt psql> .

```
bash# su - postgres
bash$ man psql
bash$ psql miodatabase
Digitate \h per leggere l'help dei comandi.
```

È molto facile da usare, ed è alla portata di ogni utente. Si può accedere a psql anche da script di shell.

15.5 MPMGR - Uno strumento di gestione di database per PostgreSQL

MPMGR fornirà un'interfaccia grafica per la gestione di database per PostgreSQL. Potete trovarlo presso

- <http://www.mutinybaysoftware.com/>
- Email: *keidav@mutinybaysoftware.com*
- <http://www.troubador.com/~keidav/index.html>
- Email: *keidav@whidbey.com*
- <http://www.ucolick.org/~de> nel file tcl_syb/wisql.html
- WISQL per PostgreSQL <http://www.ucolick.org/~de/Tcl/pictures>
- Email: *de@ucolick.org*

15.6 Gli strumenti PgAdmin, PhpPgAdmin

- Strumento PgAdmin per Windows 95/NT. Strumento di progettazione di database PostgreSQL per Windows 95/NT <http://www.pgadmin.freeserve.co.uk>
- Strumento di amministrazione basato su Web: PhpPgAdmin per Postgresql, si trova presso <http://www.phpwizard.net/projects/phpPgAdmin>

15.7 PgBash - strumento SQL per la shell

PgBash ha funzionalità simili a psql. Inoltre PgBash fornisce le utili funzionalità di bash, l'ambiente operativo interattivo e flessibile, come gli alias, le funzioni, l'elaborazione dello storico di riga di comando, ecc.

Il sito principale di PgBash è presso <http://www.psn.co.jp/PostgreSQL/pgbash/index-e.html>

PgBash è una shell che ha incluso un "SQL diretto" o, in altri termini, ha l'interfaccia per PostgreSQL con "SQL incorporato"; è stata ottenuta mediante perfezionamento della shell bash (versione 2.03). PgBash può essere usata come shell di log-in, come sub-shell (la shell generata all'interno di un'altra shell), e come shell per esecuzione di programmi.

Le funzionalità di un "SQL Diretto" permettono di inviare immediatamente allo "standard output" i risultati delle query. Le funzionalità di "SQL Incorporato" permettono di memorizzare i valori, restituiti dalle query, in variabili di shell che possono essere gestite con il linguaggio di script della shell.

Un'istruzione SQL (il cui carattere finale è un punto e virgola), viene gestita come un comando della shell; è possibile mandare in esecuzione le istruzioni SQL con filtri (pipeline), ridirezione e opzioni per l'esecuzione sullo sfondo dei job. Inoltre, usando il comando exec_sql, è possibile far eseguire istruzioni SQL con svariate opzioni. L'esempio pratico di SQL è mostrato di seguito.

```
prompt> /usr/local/bin/pgbash                .....Avvio di pgbash
pgbash> connect to db2@xxx.com user sakaida; .....connessione al database
pgbash> select * from test limit 100; | more .....con uso di filtro (pipeline)
pgbash> select * from test; > /tmp/sel.dat & .....con redirezione e job sullo sfondo
pgbash> addr='Osaka'
pgbash> insert into test values(
>  111,'name',                               .....potete scrivere su una nuova riga
> '$addr'                                    .....uso di variabile di shell
> );
pgbash> connect to db3@yyy.com user postgres;
pgbash> set connection db2;                  .....impostate l'attuale D/B a db2
pgbash> select * from test;                  .....select di test del db2
pgbash> exec_sql -d db3 "select * from test3".....cambiate la connessione a db3
pgbash> ls
pgbash> begin;
pgbash> declare cur cursor for select * from test;
pgbash> fetch in cur into :AA,:BB;           .....impostate i valori delle variabili di shell
pgbash> echo "AA=$AA, BB=$BB"
pgbash> end;
pgbash> fc fetch                             .....utilizzate lo storico (history) dei comandi
pgbash> !echo                                .....rieseguite echo
pgbash> disconnect all                       .....interrompete tutte le connessioni
pgbash> exit                                 .....Uscita da pgbash
```

15.8 Lo strumento Webmin per PostgreSQL

Lo strumento webmin permette di amministrare una macchina Unix attraverso una pagina web, opzionalmente resa sicura. Nella sua ultima release (vers. 0.82), webmin ha un modulo PostgreSQL. Con questo modulo potete aggiungere utenti, gruppi, database, tabelle, viste di tabelle.

Potete trovare webmin presso <http://www.webmin.com/webmin>

16 CPU per PostgreSQL

Per avere la lista delle CPU utilizzabili con PostgreSQL, vedere il documento <http://metalab.unc.edu/LDP/HOWTO/CPU-Design-HOWTO.html>; inoltre questo documento fornisce dettagli sulla progettazione di CPU.

Le seguenti CPU (a 64-bit e 32-bit) sono utilizzabili con PostgreSQL. Linux gira sotto tutte queste CPU.

- Il principale sito sulle CPU è: Google Search engine CPU "Computers>Hardware>Components>Microprocessors" <http://directory.google.com/Top/Computers/Hardware/Components/Microprocessors>

Quella che segue è una lista di CPU open-source GNU/GPL:

- Sito delle CPU Open-source: Google Search "Computers>Hardware>Open Source" <http://directory.google.com/Top/Computers/Hardware/Open_Source>
- OpenRISC 1000, è un processore free a 32-bit con core IP, ed è in competizione con i processori proprietari ARM e MIPS; si trova presso <http://www.opencores.org/cores/or1k>
- OpenRISC 2000, si trova presso <http://www.opencores.org>
- Le CPU "LEON" Sparc ESA-32bit ed ESA-64bit, della European Space Agency <http://www.estec.esa.nl/wsmwww/leon>
- GNU/GPL Freedom 64-bit F-CPU <http://www.f-cpu.org>, o presso <http://f-cpu.tux.org> sito mirror <http://www.f-cpu.de>.
- STM 32-bit, CPU RISC a 2-vie superscalare <http://www.asahi-net.or.jp/~uf8e-itu>
- Microprocessore free e core DSP IP, scritti in Verilog o VHDL <http://www.cmosexod.com>
- Core hardware free per velocizzare lo sviluppo <http://www.scrap.de/html/opencore.htm>
- Opencores org: core IP liberi/gratuiti, open source <http://www.opencores.org>
- Linux open hardware, e sistemi EDA liberi <http://opencollector.org>
- ARM CPU <http://www.arm.com/Documentation>
- Le CPU Cogent <http://www.cogcomp.com>

Segue una lista di CPU commerciali:

- CPU russa E2k a 64-bit (Una CPU molto veloce !!!) sito web: <http://www.elbrus.ru/roadmap/e2k.html>. Ora Elbrus è partner (alleato) della Sun Microsystems (USA)
- CPU Coreana a 64-bit della Samsung, originaria del DEC Alpha <http://www.samsungsemi.com> Il sito della CPU Alpha a 64bit è <http://www.alpha-processor.com> Attualmente esiste una collaborazione tra Samsumg e Compaq (USA) per la CPU Alpha
- CPU crusoe Transmeta e, nel prossimo futuro, CPU Transmeta a 64-bit <http://www.transmeta.com>
- CPU Ultra-sparc a 64-bit della Sun <http://www.sun.com> , oppure <http://www.sunmicrosystems.com>
- Le CPU MIPS RISC CPU <http://www.mips.com>
- CPU con Architettura MIPS della Silicon Graphics <http://www.sgi.com/processors>
- Le CPU di Architettura IDT MIPS <http://www.idt.com>
- IBM Power PC (motorola) <http://www.motorola.com/SPS/PowerPC/index.html>
- Processore RISC SuperH a 64-bit SH7750Hitachi <http://www.hitachi.com>, in vendita a 40 $ per cpu in quantità di 10'000 unità.

- Processore Fujitsu a 64-bit <http://www.fujitsu.com>
- Processore HAL-Fujitsu (California) Super-Sparc a 64-bit <http://www.hal.com>, compatibile anche con l'architettura sparc di Sun
- CPU Seimens Pyramid CPU della Pyramid Technologies
- CPU Intel serie X86 a 32-bit come Pentium, Celeron ecc..
- CPU AMD serie X86 a 32-bit come K-6, Athlon ecc..
- CPU Cyrix serie X86 a 32-bit della National come Cyrix ecc..
- Altre CPU prodotte in altri paesi (Taiwan, Corea, Giappone) ?? Fatemi sapere...

Altri siti importanti dedicati alle CPU, si trovano presso:

- Notizie sulle CPU per 24-ore al giorno e da tutto il mondo <http://www.newsnow.co.uk/cgi/NewsNow/NewsLink.htm?Theme=Processors>
- Sito sull'architettura dei computer <http://www.cs.wisc.edu/~arch/www>
- CPU ARM <http://www.arm.com/Documentation>
- Great CPUs <http://www.cs.uregina.ca/~bayko/cpu.html>
- Risorse sul Microdesign <http://www.mdronline.com>

17 Predisporre multi-box di PostgreSQL con un solo monitor

Se volete spendere tutti i quattrini che potete in commutatori (noti anche come "switch hardware", n.d.t.), potete **usare la VNC (Virtual Network Computing) Technology**, offerta dal gigante delle telecomunicazioni AT&T. VNC è sotto licenza GPL, ed è software libero. Usando VNC potete far girare i programmi PostgreSQL su computer senza monitor, ed inviarne la visualizzazione a box remoti muniti di monitor!! I box devono, però, essere connessi in rete tramite schede ethernet. VNC è situato presso <http://www.uk.research.att.com/vnc>

Potete anche impilare una serie di box con CPU, connetterli ad un solo monitor, e usare lo switch box KVM (Keyboard, Video, Monitor) per selezionare l'host. Ciò permetterà di risparmiare spazio e di evitare molto disordine, e permetterà anche di ridurre il numero di monitor, tastiera e mouse (consentendo un risparmio valutabile da 100 a 500 dollari USA per unità, ovunque voi siate).

Usando questo switch box, potete impilare molti server PostgreSQL (per gli ambienti di sviluppo, collaudo, produzione), server Web, server ftp, server intranet, server Mail, server News; potrete mettere questa torre su uno scaffale. Gli switch box possono essere usati per controllare altrettanto bene box con Windows 95/NT o OS/2.

Visitate questi siti:

- DataComm Warehouse Inc al tel. 1-800-328-2261. Fornisce un'ampia gamma di hardware per computer <http://www.warehouse.com>. 4-port Manual KVM è uno switch manuale a 4 porte (PS/2), e costa circa 89.99 $ Part No. DDS1354
- Network Technologies Inc <http://www.networktechinc.com/servswt.html> (120 dollari/PC 8 porte), dove troverete una lista di Switch-Server e Switch-video.
- Scene Double Inc, Inghilterra <http://www.scene.demon.co.uk/qswitch.htm>

- Cybex corporation <http://www.cybex.com>
- Raritan Inc <http://www.raritan.com>
- RealStar Solutions Inc <http://www.real-star.com/kvm.htm>
- Belkin Inc <http://www.belkin.com>
- Better Box Communications Ltd. <http://www.betterbox.com/info.html>
- Recatevi al negozio di hardware più vicino, e chiedete "Server Switch", noti anche come "KVM Auto Switches".

Usate il motore di ricerca yahoo per trovare altre compagnie con "Server Switches" o "KVM Switches".

È fortemente consigliato riservare uno unix-box per ognuno dei data-server PostgreSQL, al fine di ottenere le migliori prestazioni. Nessun altro programma/processo/applicazione dovrebbe girare su questo box. Consultate la sezione Affari del vostro giornale locale, per trovare un fornitore locale che venda soltanto box intel, e monitor monocromatici da 13" (monitor dal bassissimo costo). i Rivenditori locali vendono soltanto hardware **senza** Microsoft Windows/DOS. Non avete bisogno di monitor a colori per il server database, e comunque potete amministrare in remoto con il monitor a colori di una workstation PC.

Presso i negozi online, potete comprare hardware di computer puro e semplice. Potete ottenere buoni prezzi in "Aste Online"

- Negozio online e sala d'asta <http://www.egghead.com>
- Negozio online <http://www.buy.com>
- Bidding store, negozio ad offerte <http://www.ubid.com>

Prelevate i cdrom di RedHat (o qualche altra distribuzione di) Linux dai siti sottoelencati:

- Sito web di Linux System Labs: <http://www.lsl.com/> (7 dollari USA)
- Sito web di Cheap Bytes Inc: <http://www.cheapbytes.com/> (7 dollari USA)

Assicuratevi che l'hardware da voi acquistato sia supportato da RedHat Linux. Consultate il sito ftp di RedHat per conoscere l'hardware raccomandato (schede SCSI, schede video..), prima dell'acquisto. Con appena 600$ otterrete un potente box intel con RedHat Linux, su cui far girare PostgreSQL. Usate odbc/jdbc/perl/tcl per connettervi a PostgreSQL da Windows95, OS/2, Unix Motif, o attraverso browser web (ad es. Redbaron, Opera, Netscape, e altri). (I browser Web stanno diventando rapidamente i client GUI standard).

Usando gli switch KVM potete controllare molte cpu box, con solo un monitor ed una tastiera!

18 Web-Application-Server per PostgreSQL

Svariati Web-Application-Server funzionano con PostgreSQL, sia nelle versioni open-source che in quelle commerciali. I Web-Application-Server open-source diffusi, sono gli Application Server basati su Perl come Mason, WIRM, Velocigen, Enhydra(Java) e Zope(Python); gli Web-Application-Server commerciali sono IBM Websphere, BEA Weblogic.

È consigliato l'uso di un web server sicuro come Apache + mod_ssl + OpenSSL. Vedere il server sicuro Redhat StrongHold, presso <http://www.c2.net/products/sh3>.

I Web Application Server possono essere classificati in base al linguaggio di programmazione supportato. Scegliete un Web Application server in base al linguaggio di programmazione che **vi piace di più**.

Una classificazione dei Web Application server è:

- Basati sul linguaggio **PERL**
- Basati sul linguaggio **PHP** (che è simile al PERL, e un po' somigliante a Java)
- Basati sul linguaggio **Python** (linguaggio di scripting orientato agli Oggetti)
- Basati sul linguaggio **Java** (Java della Sun Microsystems)
- Basati sul linguaggio **Tcl** (linguaggio di scripting Tcl/Tk, chiamato "Tickle")
- Basati sul linguaggio **C++** (C++ e CORBA)
- Basati sul linguaggio **Pike** (linguaggio di scripting simile al C++)

18.1 Web Application Server in PERL

Il linguaggio Perl, come il linguaggio "C", ha una storia molto lunga, e sarà usato per **migliaia di anni** nel futuro! Perl gira **10 volte** più rapidamente di Java, e qualche volta è più rapido anche del "C". Java è un sistema molto complesso; la sua macchina virtuale e l'interprete lo rendono estremamente lento, instabile ed inaffidabile. Perl è molto semplice, rapido, ed è orientato agli oggetti.

Inoltre, i programmi in Perl possono essere anche facilmente compilati per migliorarne le prestazioni. Usate Perl2Exe, una utility a riga di comando per convertire script perl in file eseguibili `<http://www.indigostar.com/perl2exe.htm>`

Per PERL sono disponibili i seguenti Web Application server:

- Mason `<http://www.masonhq.com>` è un potente motore per sviluppo e distribuzione di siti web, basato su Perl. Con Mason potete incorporare il codice Perl nel vostro HTML, e potete costruire pagine a partire da componenti riusabili e condivisibili.
- BingoX `<http://opensource.cnation.com/projects/BingoX>` è un Web Application Framework open source, orientato agli oggetti; è stato scritto in mod_perl, il che riduce drammaticamente il tempo richiesto per costruire siti web e applicazioni dinamici, funzionanti con database estesi.
- SmartWorker è una collezione di classi Perl, che permette di costruire applicazioni web come se fossero vere applicazioni, e non solo modelli in HTML con codice incorporato in modo casuale. SmartWorker `<http://www.smartworker.org>`
- Progetti di integrazione Apache-Perl. Con mod_perl è possibile scrivere moduli Apache interamente in Perl. In aggiunta a ciò, l'interprete, persistente ed incorporato nel server, si sottrae all'overhead dovuto all'avvio di un interprete esterno, e allo svantaggio dei tempi di avvio di Perl. Fate visita a `<http://perl.apache.org>`, e date uno sguardo anche al progetto **mod_perl_garden**, presso `<http://modperl.sourcegarden.org>`
- Apache::ASP `<http://www.apache-asp.org>` , fornisce una versione delle Active Server Page per il Server Web Apache, utilizzando Perl come linguaggio di scripting ospitato. Apache::ASP consente ad uno sviluppatore di creare applicazioni web dinamiche, con gestione delle sessioni e l'incorporazione del codice perl. Esistono anche potentissime estensioni che comprendono librerie di tag XML, XSLT rendering, e nuovi eventi non facenti parte delle API ASP originarie.
- WIRM (Web Interface Repository Manager) è un application server, basato su Perl, che fornisce un ambiente di programmazione d'alto livello per sviluppare sistemi informativi via web. WIRM consiste di un database relazionale ad oggetti, e di una suite di interfacce in Perl per la visualizzazione, l'integrazione e l'analisi di dati multimediali eterogenei. WIRM fornisce gli strumenti per creare punti di vista sensibili al contesto, a partire da un database multimediale; gli sviluppatori possono, quindi,

costruire rapidamente siti web dinamici, che adattino il loro contenuto, e la presentazione, a classi multiple di utenti finali. Visitate <http://www.wirm.org>

- EmbPerl <http://perl.apache.org/embperl>. Embperl vi dà la capacità di incorporare codice Perl nei vostri documenti HTML. Usare Perl significa essere capaci di usare un linguaggio di programmazione molto elaborato, largamente impiegato nell'ambito del WWW. Potete anche utilizzare le migliaia di moduli Perl che sono già state scritte - compreso DBI - per l'accesso a database di un numero sempre maggiore di sistemi di database.
- ePerl <http://www.engelschall.com/sw/eperl> , interpreta un file ASCII, riempito di istruzioni in Perl 5, passando al vaglio il codice Perl 5 durante la scansione dei dati in ASCII puro. Può operare in varie maniere: come filtro Unix a sé stante, come modulo Perl 5 integrato (per compiti di generazione di file di tutti i tipi), e come potente linguaggio di scripting da Webserver per programmazione di pagine HTML dinamiche.
- XPP <http://opensource.cnation.com/projects/XPP> sta per "XPP Parses Perl" (XPP analizzatore sintattico di Perl), o anche "XPML Page Parser" (analizzatore di Pagina XPML), ed è un parser di HTML rapido/efficiente; XPP processa il codice perl incorporato, ma anche tag come quelli di HTML, a partire da pagine html dinamiche chiamate "pagine XPML".
- Gamla - un RAD e application server basato su perl. Il progetto gamla aspira alla creazione di uno strumento per sviluppo rapido di applicazioni (RAD), e di un web application server basati su Perl. Tutto il codice sorgente, prodotto in seno al progetto Gamla, sarà di pubblico dominio.

  Gamla si trova presso <http://gamla.iglu.org.il>
- AxKit <http://www.axkit.org> è un Application Server XML per Apache (e mod_perl). Esso permette di convertire al volo XML in qualsiasi altro formato, come HTML, WAP o testo puro; per far questo, si impiegano tecniche standard W3C, oppure si usa codice flessibile realizzato allo scopo. AxKit utilizza anche un interprete Perl incorporato per fornire alcune tecniche, sorprendentemente potenti, per trasformazioni di XML.

  AxKit mette l'accento sulla separazione tra il contenuto e la sua presentazione. La tecnica di filtraggio, impiegata da AxKit, permette la conversione del contenuto in un formato presentabile a stadi; in questo modo si permette, a certe piattaforme, di vedere i dati differentemente da come vengono visti da altre. AxKit permette ai web designer di concentrarsi sulla progettazione del sito web, per lavorare sul solo contenuto di base; i webmaster, invece, potranno concentrarsi sulle loro competenze di fondo.

Web Application Server Commerciali per Perl:

- Zelerate AllCommerce <http://www.zelerate.org/html/eng/home.shtml> è un sistema di gestione delle relazioni commerciali, dei volumi, dei clienti. Quest'applicazione per Internet, scalabile e dalle elevate prestazioni, è scritta in Perl, ed impiega un database di backend.
- VelociGen serve contenuto dinamico archiviato in XML, o su database, o a partire da dati liberi, e lo fa alla stessa velocità con cui vengono servite pagine HTML statiche: fino a 60 volte più velocemente della CGI, e senza dover modificare le vostre applicazioni pre-esistenti. VelociGen, inoltre, rende più facile sviluppare nuovi programmi con l'uso di tag XML dal lato server, fornisce protezione dai crash, ed equilibra il carico su macchine multiple.

  VelociGen si inserisce in maniera identica all'interno di qualsiasi Web server di qualsiasi piattaforma; con esso vengono incrementate le prestazioni del server, e velocizzati i tempi di risposta del sito web, funzionante con contenuto dinamico. VelociGen può elaborare grandi volumi di richieste simultanee, fino a 10 volte più rapidamente dei Servlet Java, e 4 volte più rapidamente di Cold Fusion.

  Velocigen <http://www.binevolve.com/velocigen>

18.2 Web Application Server per PHP

Per PHP sono disponibili i seguenti Web Application server

- Midgard PHP Web Application server è basato sul linguaggio di scripting PHP, e PHP gira estremamente rapido: più rapidamente di Java. Il sito principale di Midgard si trova presso <http://www.midgard-project.org>. Gli script PHP possono essere compilati con il compilatore ed ottimizzatore Zend <http://www.zend.com>. PHP gira molto velocemente, ed è da 5 a 10 volte più veloce di Java.

  Vedere D (Installazione di Midgard), e anche il PHP HOWTO presso <http://www.linuxdoc.org/HOWTO/PHP-HOWTO.html>

- Ariadne <http://www.muze.nl/software/ariadne> è un sistema per applicazioni web. Esso consiste di una infrastruttura completa, volta a facilitare sviluppo e gestione di applicazioni web, impiegando il PHP. Il sistema usa un approccio modulare, utilizzando allo scopo interfacce astratte per tutte le transazioni. Ne deriva la massima libertà di cambiare alcune parti dei sistemi, modificandole o aggiungendo nuove funzionalità, senza dover riprogrammare le altre parti.

18.3 Lutris Corp "Enhydra Enterprise" (Java)

Enhydra supporta il database PostgreSQL. Enhydra è un Web-Application-Server per Java/XML enormemente diffuso, creato dalla 'Lutris Corporation'. È il miglior server per Applicazioni Web in Java/XML. Supporta EJB, Servlets, JSP, JNDI, JDBC, JTA, CORBA, XMLC/Rocks, DODS e l'internazionalizzazione. È usato da molte delle 500 maggiori compagnie a grande capitalizzazione di USA ed Europa; queste compagnie (come "France Telecom") stanno finanziando direttamente Enhydra. È stato scritto in Java puro al 100%, ed è prelevabile presso <http://www.enhydra.org>. Enhydra è un progetto realizzato con codice open source, ma è venduto e supportato commercialmente da Lutris Corp. Visitate il sito <http://www.lutris.com>

Leggete il tutorial sulla configurazione di *PostgreSQL con Enhydra* <http://www.enhydra.org/software/documentation/enhydra/NewApp-DODS-Tutorial-PGSQL.html> , e leggete anche come effettuare la *Configurazione del database con Enhydra* <http://www.enhydra.org/software/documentation/enhydra/Enhydra-NewApp-DODS-Tutorial.htm>.

Con Enhydra dovreste usare JBuilder della Borland Corp.. Jbuilder si trova presso <http://www.inprise.com>

Leggete anche l'Enterprise-Java-HOWTO presso <http://www.linuxdoc.org/HOWTO/Enterprise-Java-for-Linux-HOWTO.html>

18.4 Zope (in linguaggio Python)

Tra i linguaggi di scripting "puramente" orientati agli oggetti, Python sta diventando molto diffuso. Zope è un server Web-Application, e fornisce un'interfaccia a PostgreSQL. Zope è disponibile presso <http://www.zope.org> Python si trova presso <http://www.python.org>

18.5 OpenACS (in linguaggio Tcl)

OpenACS (Open ArsDigita Community System) <http://openacs.org> è un toolkit avanzato per costruire applicazioni web scalabili, orientate alle community. Funziona su AOLserver, un server per applicazioni web, e PostgreSQL, un RDBMS che rispetta le proprietà ACID. Questi due prodotti sono di alta qualità, e sono disponibili gratuitamente sotto licenze open-source.

ACS è stato creato da ArsDigita, <http://www.arsdigita.com>, e questo loro ACS (ArsDigita Community System) mira ad essere indipendente dal database per quanto possibile, benché sia basato su Oracle (ed è quindi chiaro che Open ACS deve prendersi tutto il tempo necessario per effettuare il porting).

Vedere anche <http://www.appserver-zone.com>

18.6 Web Application Server CORBA in C++

- il Web Application Server PortalSphere è realizzato in C++ , e gira sotto Unix (e Linux) per l'estrema velocità e stabilità. Aderendo strettamente allo standard CORBA, PortalSphere supporta sia HTTP (il protocollo di comunicazione standard in Internet) che il protocollo punto-a-punto IIOP, adatto a comunicazioni client-server a velocità ultra rapida. Queste caratteristiche di PortalSphere, associate ad un accesso diretto (nativo) a tutti i database diffusi, gli conferiscono prestazioni lampo, e l'eccezionale capacità intrinseca di gestire eventi in tempo reale su internet. PortalSphere è oltre 100 volte più veloce di HTTP/CGI, Munito di accesso diretto (nativo) ai database più diffusi, Scalabile fino a più di 10'000 sessioni-utente concorrenti.

  Visitate PortalSphere presso il sito <http://www.portalsphere.com/overview.html>.

- FlashPoint è un progetto di Web Application Server in C++,C,PERL; è in grado di supportare servizi con applicazioni web ad alta velocità in un ambiente multi-thread, può supportare svariati linguaggi di sviluppo inclusi C & C++; inoltre supporta buone procedure di ingegneria del software, che in altri ambienti sarebbero terribilmente ardue da implementare. Flashpoint può essere usato a fianco di Apache, e in qualche caso può sostituirlo, a seconda delle vostre necessità. Visitate <http://www.bouldersoftware.com/products/flashpoint>, e scaricate il pacchetto da *FlashPoint Redhat RPM* <http://www.bouldersoftware.com/products/flashpoint/download.html>.

- "C Server Pages" <http://cserverpages.20m.com> è un Application Server, scritto in C/C++, efficiente e scalabile; permette di potenziare le pagine web server, ed i Template scritti in C++, con Elementi Dinamici incorporati. Potete usare l'approccio che preferite. Potete costruirvi i vostri oggetti di lavoro usando il C++. Le vostre pagine possono essere i client CORBA per qualsiasi ORB sulla piazza. È presente connettività a tutti i database SQL.

18.7 Pike, Roxen Web Application Server

Pike è un linguaggio di programmazione dinamico, con una sintassi simile al quella del C++. È semplice da apprendere, non richiede lunghi passi di compilazione, e ha potenti tipi di dati incorporati che ne permettono una gestione semplice e rapida. Pike è stato rilasciato sotto la licenza GNU/GPL, o general public license.

Pike è un potentissimo linguaggio di scripting orientato agli oggetti; poiché ha una sintassi identica a quella del C++, se ne prevede una diffusione **esploderà**, nei prossimi anni.

Pike si trova presso il sito <http://pike.roxen.com>, e il web server Roxen è presso <http://www.roxen.com>.

Roxen è un web server modulare con un'interfaccia DB completa, ed include supporto a Postgres. Ha pieno supporto per SSL, ed è stato rilasciato sotto la GPL. Roxen è stato scritto utilizzando il linguaggio di scripting Pike.

18.8 Web Application Servers Directory

Visitate *Web Application Servers* <http://198.85.71.76/html.html>. Recatevi nella directory con "Yellow Pages".

19 Applicazioni e Strumenti per PostgreSQL

19.1 4GL per applicazioni web database con PostgreSQL - AppGEN Development System

AppGEN può essere scaricato da

- <http://www.man.ac.uk/~whaley/ag/appgen.html>
- <ftp://ftp.mcc.ac.uk/pub/linux/ALPHA/AppGEN>.

AppGEN è un linguaggio ad alto livello di quarta generazione, ed un generatore di applicazioni per produrre applicazioni basate sul World Wide Web (WWW). Queste applicazioni sono tipicamente usate in internet o nelle intranet aziendali. Le applicazioni AppGEN sono implementate come script C conformi al Common Gateway Interface (CGI), che è uno standard supportato dalla maggior parte dei Web Server.

Per usare AppGEN vi servirà quanto segue:

PostgreSQL, sistema di gestione di database relazionali (RDMBS).

Un Web server compatibile con CGI, come l'HTTPD di NCSA.

Un compilatore ansi C come GCC.

AppGEN consiste dei seguenti eseguibili Unix (Linux):

- defgen, che produce un modello di applicazione elementare a partire da una struttura di dati logica. Le applicazioni sono in grado di inserire, aggiornare, cancellare e ricercare su record del database, e allo stesso tempo di mantenere automaticamente l'integrità referenziale.
- appgen, il compilatore di AppGEN che compila il codice sorgente appgen in linguaggio C ed eseguibile in CGI, e documenti formattati HTML, pronti ad essere resi disponibili su un server web.
- dbf2sql, una utility per convertire i file .dbf (dBase III compatibili) in script eseguibili SQL. Questo permette ai dati, archiviati nella maggior parte dei pacchetti di gestori di database per DOS/Windows, di essere trasferiti ad un server SQL come PostgreSQL.
- In aggiunta, AppGEN include una collezione di documenti HTML, di file GIF e di applet Java, che vengono usati nella fase di esecuzione dal sistema. E, naturalmente, come tutto il buon software, viene incluso anche l'intero codice sorgente.

L'autore, Andrew Whaley, può essere contattato all'indirizzo

- *andrew@arthur.smuht.nwest.nhs.uk*

19.2 Interfaccia Web WWW per PostgreSQL - DBENGINE

Dbengine, un'interfaccia Web plug 'n play per PostgreSQL, è stata creata da Ingo Ciechowski. La trovate al sito

- <http://www.cis-computer.com/software/dbengine>

Qualche informazione su DBENGINE: dbengine è un'interfaccia tra il Web e Postgres95, che fornisce un semplice accesso a qualsiasi database esistente in pochissimi minuti.

PHP offre un linguaggio simile al Perl per i vostri documenti, ma non è un vero Perl; per contro AppGen e wdb-p95 richiedono la creazione di un qualche file di configurazione per ognuno dei vostri database; sembra quindi che, prima di poter iniziare a usare questi strumenti, si debba imparare una qualche specie di nuovo meta linguaggio.

Diversamente da altri strumenti, per iniziare ad usare dbengine non dovete imparare qualche speciale linguaggio di programmazione o di scripting. Inoltre non si deve realizzare un file di configurazione per ogni database, così non dovete familiarizzare con questa nuova struttura. Comunque, nel caso vogliate accedere a tutte le funzionalità di dbengine, dovreste conoscere il linguaggio Perl.

L'intero sistema può essere configurato mediante la semplice manipolazione di un database aggiuntivo, che contiene informazioni riservate riguardanti il vostro accesso al database. Potete anche specificare Campi virtuali che vengono calcolati al volo poco prima di essere visualizzati sullo schermo.

19.3 Modulo del Webserver Apache per PostgreSQL - NeoWebScript di NeoSoft

Apache è un notissimo Server Web. Un modulo per interfacciare PostgreSQL ad Apache Webserver, è reperibile presso:

- `<http://www.neosoft.com/neowebscript/>`

NeoWebScript è un linguaggio di programmazione che permette a programmi semplici o complessi di essere incorporati in file HTML.

Quando una pagina HTML con codice NeoWebScript incorporato viene richiesta, il server Web, abilitato all'esecuzione dei NeoWebScript, esegue il codice script incorporato, producendo una pagina web dal contenuto appositamente creato dal programma.

NeoWebScript è un modo rapido, sicuro, facile da apprendere, per realizzare programmi interattivi nel codice HTML delle pagine web, ed ospitati su server. Con NeoWebScript i contatori, i form per email, i graffiti wall, i guest book ed il tracciamento dei visitatori, sono facilissimi, anche per un programmatore alle prime armi. Potrete verificare che NeoWebScript non teme rivali, come PERL e JavaScript.

Se volete installare NeoWebScript sul vostro server web, il vostro Webmaster dovrebbe leggere, per cominciare, la nostra Sysop FAQ. In 'Theory of Operations' si renderà chiaro il funzionamento di NeoWebScript, mentre l'installazione procederà passo dopo passo. 'Management' tratta di questioni riguardanti la configurazione ed il funzionamento del server; i test vi permetteranno di verificare la correttezza delle operazioni di NeoWebScript, e 'troubleshooting' tratta dei problemi del server.

Usare NeoWebScript-2.2 non comporta alcuna spesa per voi, per il vostro ISP, la vostra intranet o extranet. Potrete leggere la licenza durante la registrazione per il download; notate che esiste un costo di 99 $ se volete incorporarlo nei vostri prodotti, od usarli in un server commerciale (come SSL)

NeoWebScript è un modulo per il server Web Apache che vi permette di includere il linguaggio di programmazione Tcl/Tk, nelle vostre pagine web, come tool di scripting. È stato inventato da Karl Lehenbauer, Chief Technical Officer della NeoSoft, ed è documentato, migliorato ed esteso dai programmatori della NeoSoft e da redattori di documenti tecnici.

Il server web Apache è il più diffuso server web del mondo, ed il suo uso è stimato al 68 % dei siti sondati.

Tcl/Tk è il potente e gratuito linguaggio di scripting multipiattaforma sviluppato dal Dr. John Ousterhout. Utilizzando le sue stesse parole

"Tcl/Tk permette agli sviluppatori di software di far girare i programmi dieci volte più rapidamente di quanto possano fare i toolkit basati su C o C++. Esso è anche un pregevole linguaggio di interconnessione, che permette di far funzionare insieme le applicazioni già esistenti, e di dotarle di migliore grafica e di interconnessione ad Internet".

Karl Lehenbauer, Fondatore e Chief Technical Officer di NeoSoft, è stato parte del team di sviluppo delle primissime versioni di Tcl/Tk. Insieme con Mark Diehkans, è l'autore di Extended Tcl, noto anche come TclX o NeoSoft Tcl, che è un potente insieme di estensioni al linguaggio. Molti degli attuali comandi Tcl sono stati creati in Extended Tcl, e sono stati inclusi del nucleo del linguaggio dal Dr. Ousterhout.

NeoSoft Inc., 1770 St. James Place, Suite 500, Houston, TX 77056 USA

19.4 HEITML: estensione lato server dell'HTML, e linguaggio 4GL per PostgreSQL

Lo strumento heitml è un altro modo di interfacciare postgres con il world wide web. Per maggiori dettagli contattate

```
Helmut Emmelmann H.E.I. Informationssyteme GmbH
Wimpfenerstrasse 23 Tel. 49-621-795141
68259 Mannheim Germany Fax. 49-621-795161
```

- E-mail di Mr.Helmut Emmelmann *emmel@h-e-i.de*
- Sito web principale di Heitml `<http://www.heitml.com>`
- Sito web secondario di Heitml `<http://www.h-e-i.deom>`

heitml è un'estensione dal lato server dell'HTML e, nello stesso tempo, è un linguaggio 4GL. Ci si può scrivere applicazioni web utilizzando lo stile HTML, usando nuovi tag simili a quelli dell'HTML.

heitml (pronunciato h-ai-TML) è un'estensione dell'HTML, ed è un linguaggio di quarta generazione ricco di funzionalità; esso permette alle applicazioni, basate su Web, di interagire con i dati archiviati nei database SQL, evitando di far ricorso a complessi script CGI.

heitml estende HTML nel lato server, convertendo dinamicamente i file ".hei" nel formato HTML, e rimanendo così compatibile con qualsiasi browser web. Heitml accetta la comoda sintassi HTML e fornisce un vasto assortimento di Tag pre-sviluppati e Librerie, per gestire i compiti che in precedenza richiedevano il CGI. Come XML, heitml fornisce tag definiti dall'utente. Con heitml, i markup definiti dall'utente possono essere tradotti in HTML ed inviati ad un browser.

heitml può interessare sia i designer HTML che i programmatori professionali. I designer HTML possono usare i Tag heitml per costruire pagine web dinamiche, per accedere a database SQL, o per creare applicazioni web complete. Contatori, database di registrazione, form di ricerca, form di e-mail, o menu gerarchici, possono tutti essere creati semplicemente usando Tag pre-sviluppati, simili ai Tag HTML, reperibili nelle moltissime Librerie di Componenti.

I programmatori troveranno in heitml un linguaggio completo di quarta generazione incorporabile nel codice HTML

```
(ad es. con i Tag <if>, <while>, e <let>),
```

inoltre heitml ammette una potente valutazione delle espressioni con i tipi di dati interi, reali, booleani, stringhe, e n-uple. Le n-uple hanno riferimenti semantici come nei moderni linguaggi orientati agli oggetti e vengono memorizzate in una heap. Le variabili heitml, comprese tutte le strutture di dati complessi memorizzate sulla heap, conservano i loro valori tra le pagine utilizzando la Modalità di Sessione (Session Mode). È possibile definire tag personalizzati o tag di ambiente, e persino ri-definire i tag HTML.

heitml rende possibile

- - - sviluppare Siti Web in modo strutturato e modulare, riducendo drasticamente il superlavoro di manutenzione.

- - - sviluppare Siti Web intelligenti ed interattivi, con contenuto che si adatta dinamicamente alle necessità dell'utente.

- - - mostrare il contenuto dei database SQL con il solo utilizzo della nostra libreria di Tag "dba" predefiniti.

- - - sviluppare database complessi ed applicazioni per Cataloghi di Vendita utilizzando Variabili di Sessione (Session Variables)

heitml gira sotto Linux con qualsiasi Server Web che utilizzi l'interfaccia CGI, ed è particolarmente veloce (evitando gli overhead del CGI) all'interno del Server Web APACHE, utilizzando le API apache. Attualmente MSQL (Versione 1 e 2), PostgreSQL (Versione 6), mysql, e gli altri database d'uso comune sono supportati). heitml, inoltre gira su Linux, BSDi, Solaris e SunOS, come anche su Windows NT con CGI, ISAPI, ODBC, e Windows 95.

heitml (su linux) è gratuito per studio, per uso privato e non-commerciale. I Siti Web Commerciali devono pagare la licenza. La versione pienamente operativa di heitml, utilizzabile liberamente per un periodo di prova, è disponibile per il download. (Notate, comunque, che ogni Pagina Web ".hei" da voi sviluppata, visualizzerà un messaggio che identifica la vostra versione di heitml come quella per un uso non-commerciale. Dopo la registrazione, riceverete una chiave per disattivare il messaggio senza dover re-installare il programma.)

heitml (pronunciato "h-ai"-TML) estende e migliora significativamente la funzionalità del linguaggio HTML con la definibilità dei tag e con le sue reali qualità di linguaggio di programmazione. Ciò rende possibile realizzare applicazioni con contenuto dinamico ed applicazioni con accesso ai database, semplicemente all'interno del mondo HTML, senza dover ricorrere al CGI e a linguaggi di programmazione o di scripting esterni. Ciò significa che voi, in qualità di autori web, potete aggiungere applicazioni nelle vostre pagine, usando semplicemente alcuni tag nuovi senza ricorrere al CGI e senza programmare. Invece, in qualità di utente esperto o di programmatore, potete creare e programmare potenti librerie di tag. Questo approccio rende heitml adeguato sia ai principianti dell'HTML sia ai programmatori professionisti. heitml gira sul server web e genera codice HTML dinamicamente, così heitml è compatibile con gli standard di internet e con qualsiasi browser web. Inoltre heitml permette un accesso completo ai database e mette al riparo l'utente dalla complessità non necessaria del CGI. heitml è stato sviluppato secondo le più recenti ricerche riguardo la realizzazione di compilatori e dei sistemi di transazione.

Le pagine heitml vengono sviluppate proprio come le pagine HTML, con un editor di testi o con un editor HTML, e vengono posizionate su un server web come si fa solitamente. Tuttavia ora le pagine possono contenere tag heitml dinamici, e possono accedere a librerie di tag. Potete usare questi tag per accedere al database, per creare contenuto dinamico, per inviare email, ed anche per creare applicazioni potenti come database di registrazione e sistemi di commercio elettronico.

I principianti dell'HTML ed i programmatori professionisti, saranno stupiti di come sia rapido e facile progettare applicazioni eccitanti, come un Guestbook Interattivo, senza dover ricorrere all'apprendimento di complessi e difficili script CGI; sarà sufficiente usare gli strumenti forniti dalla vostra Libreria dba.

heitml è rifornito di un'ampia gamma di librerie di tag per creare guestbook, applicazioni per la manutenzione dei database, moduli per interrogazioni estendibili, potenti moduli per email, oppure per strutturare il vostro sito web utilizzando un menu gerarchico. Questi strumenti sono pronti all'uso, ed è sufficiente aggiungere i tag opportuni al vostro sito.

Se siete esperti programmatori potete fare pieno uso dell'architettura basata sulle n-uple dinamiche e persistenti: heitml non è soltanto un linguaggio di scripting con gestione dinamica dei tipi di dati, con una potente valutazione di espressioni, con procedure ricorsive, e con la funzionalità del passaggio dei parametri; esso fornisce anche la persistenza dinamica delle n-uple per mantenere automaticamente i dati della sessione, di qualsiasi dimensione essi siano.

19.5 Web Server AOL di America On-line per PostgreSQL

AOLserver, il webserver commerciale gratuito, supporta connessioni ai database di PostgreSQL. Per maggiori informazioni vedere

- home page di AOL Web Server `<http://www.aolserver.com>`
- Introduzione a AOLserver di Philip Greenspun `<http://photo.net/wtr/aolserver/introduction-1.html>`

AOLserver è un webserver veloce, dotato di multithread, ed in grado di usare il Tcl. Non solo: esso è una piattaforma completa per sviluppo di back-end per database. Con AOLserver potete gestire connessioni multiple a PostgreSQL (e ad altri RBDMS) che possono essere divise tra diversi thread. AOLserver ha una API Tcl e C che vi permette di sviluppare potenti siti web dinamici. Tutto ciò avviene fin dal 1995. AOLserver è stato rilasciato sotto la licenza APL (AOLserver Public License) oppure sotto licenza GPL, e perciò è software del tutto libero. L'API Tcl è quella maggiormente utile per i siti web. AOLserver ha un set di potenti chiamate Tcl, come ns_sendmail (per inviare e-mail), ns_httpget (per ottenere un URL), ns_schedule (una funzionalità simile a cron per pianificare procedure da far girare in momenti specifici), ecc. Potete anche estendere molto facilmente le capacità di AOLserver con le API Tcl. Ogni server virtuale AOLserver può avere la sua propria "library" privata di script Tcl, i quali vengono sottoposti ad analisi (parsing) da AOLserver, e divengono accessibili da qualsiasi pagina interna a quel server virtuale. Potete sviluppare pagine per AOLserver in tre modi: - HTML semplice - pagine .tcl – si tratta di programmi tcl che possono restituire codice HTML per mezzo della chiamata ns_write. - pagine .adp – AOL Dynamic Pages. Potete sviluppare le vostre pagine in semplice HTML ma potete mandare in esecuzione codice Tcl usando <% %> o <%= %> alla maniera di PHP o di ASP. Se è già un grande server web con una superba architettura, AOLserver è addirittura splendido quando è usato per connettività a database. AOLserver ha il suo proprio strato (layer) di astrazione del database, che vi mette in grado di connettervi a diversi RDBMS senza dover modificare affatto il vostro codice. Le connessioni al RDBMS sono associate tra loro, sono persistenti, e sono divise tra diversi thread. Ciò permette connessioni molto rapide ed un efficiente uso delle risorse. AOLserver è munito di driver per tutti i più importanti RDBMS: PostgreSQL, Oracle, Sybase, Informix, Illustra, Solid, Interbase, MySQL.

19.6 Problem/Project Tracking System Application Tool per PostgreSQL

Si trova presso

- `<http://www.homeport.org/~shevett/pts/>`

19.7 Conversione di file dbase dbf a PostgreSQL

Il programma dbf2msql funziona bene con mSQL e PostgreSQL. Potete trovarlo presso

- `<ftp://ftp.nerosworld.com/pub/SQL/dbf2sql/>`
- `<ftp://ftp.postgresql.org/pub/contrib/dbf2pg-3.0.tar.gz>`
- Pg2Xbase is a better package then dbf2pg `<http://w3.man.torun.pl/~makler/prog/pg2xbase>`

Questo programma è stato scritto da Maarten Boekhold, della Facoltà di Ingegneria Elettronica, sezione di Architettura dei Computer e Tecniche Digitali a Delft, in Olanda.

- *M.Boekhold@et.tudelft.nl*

Potete anche usare un metodo di python per leggere i file dbf e caricarli (farne la load) in un database postgres.

- Vedere `<http://www.python.org>`

19.8 Conversione di file database MDB di Microsoft Access in PostgreSQL

MDB Tools è un insieme di librerie e di programmi di servizio, progettati per facilitare l'esportazione di dati dai database MS Access (file mdb) nei database multiutente come Oracle, Sybase, DB2, Informix, MySQL, Postgresql, o simili.

- Prelevate il tool MDB dal sito `<http://mdbtools.sourceforge.net>`
- Mailing list `<http://lists.sourceforge.net/mailman/listinfo/mdbtools-dev>`

19.9 Client Zeos

"Zeos" è un programma prodotto per lo sviluppo e l'amministrazione delle applicazioni database, con uso di server SQL OpenSource: MySQL, PostgreSQL e InterBase `<http://www.zeos.dn.ua/eng/index.html>`

19.10 Report Writer in Java

Generic Report Writer è un software, guidato da menu, per effettuare report. Non è un'interfaccia di tipo drag-and-drop. Funziona con PostgreSQL, MySQL, e Access. Probabilmente funzionerà con qualsiasi altro database a patto che si abbia la versione 1 del driver JDBC Tipo 4. Si trova presso `<http://www.geocities.com/SiliconValley/Ridge/4280/GenericReportWriter/grwhome.html>`

20 Strumento di Progettazione di Database - Strumento per rappresentare le Relazioni tra Entità, in forma di Diagramma

"DeZign for databases" (`<http://www.heraut.demon.nl/dezign/index.html>`) è uno strumento per lo sviluppo di database, che impiega un diagramma entità-relazioni. Esso aiuta, con metodi visuali, la schematizzazione di entità e relazioni, e genera automaticamente schemi SQL per i DBMS maggiormente usati.

"DeZign for databases" fornisce i livelli-dati logico/fisico a partire da singole specifiche, realizzate mediante migrazione della chiave esterna automatica al momento della progettazione. Le opzioni di visualizzazione multipla comprendono: entità / chiave_primaria / attributi_delle_chiavi_esterne_inclusive / attributi_delle_chiavi_esterne_esclusive. "DeZign for databases" supporta anche i domini (tipi di dati definiti dall'utente).

I report generati da DeZign possono essere usati per rendere più semplice la gestione di progetti complessi, in tutti i livelli di gestione. Potete generare report, dizionari di dati e database, con un semplice clic. Sono supportati i seguenti database: Oracle, Interbase, IBM DB2, Sybase, MS Access (95/97/2000), MS SQL Server, Paradox, dBase, Informix, SQL-Anywhere, MySQL e PostgreSQL.

"DeZign for databases", di Heraut, è collocato presso (`<http://www.heraut.demon.nl>`)

21 EARP - strumento di Progettazione/Implementazione di Database Web

- <http://www.oswego.edu/Earp>
- <ftp://ftp.oswego.edu> nella directory 'pub/unix/earp'.

21.1 Cos'è EARP ?

Il programma "Easily Adjustable Response Program (EARP) è stato creato da David Dougherty. EARP è uno strumento di Progettazione/Implementazione di Database Web, costruito sopra un sistema di database PostgreSQL. Le sue funzionalità includono:

- Un Sistema di Progettazione Visuale.
- Un'interfaccia a sendmail. (può gestire posta elettronica in arrivo ed in partenza)
- Un Perfezionato Meccanismo di Protezione.
- Un driver cgi.

21.2 Implementazione

La principale implementazione di EARP è un binario CGI che gira, sotto il demone http, per fornire accesso al server database. Tutti gli strumenti per la progettazione sono stati incorporati all'interno del driver, ed i progetti sono destinati esclusivamente al web. Gli strumenti stessi richiedono un browser grafico; la compatibilità degli oggetti, progettati con gli strumenti, è indipendente dall'implementazione, basata sulle preferenze individuali di progettazione.

21.3 Come funziona?

Una delle principali funzionalità di EARP consiste nella produzione di pagine html, con interfaccia al database, seguendo l'approccio Orientato agli Oggetti. La maggior parte delle pagine consisteranno in diversi oggetti. Ogni oggetto è prodotto da qualche tipo di strumento e, per un dato un nome, gli oggetti sono tra loro collegati in una sequenza, richiamabile dalla pagina degli strumenti. Gli oggetti sono anche riutilizzabili attraverso pagine multiple. Esistono strumenti elementari per HTML, Query, Cattura di input dai form, Formattazione Estendibile degli oggetti Query e Input, e Collegamento di oggetti all'interno di altri oggetti. Strumenti ancora più avanzati includono lo strumento di gestione della posta elettronica e quello di Interrogazione in regime di multithread.

Un'altra funzionalità di EARP è la gestione avanzata della sicurezza. L'accesso alle diverse aree del sistema EARP può essere limitato in vari modi. Per realizzare la sua avanzata gestione della sicurezza, EARP effettua controlli su ogni connessione al sistema, determinando a quali id e gruppi appartiene l'agente connesso. L'accesso alle aree è definito a parte, e la combinazione (tra connessione ed area permessa, n.d.t.) decide se l'accesso ad un'area specifica di Earp sia permessa o no. Tutto quello che è richiesto, per implementare le funzionalità della sicurezza, è un server http che supporti l'autenticazione dell'utente, quantomeno in modo basilare.

21.4 Dove ottenere EARP ?

EARP è disponibile tramite ftp anonimo da

- <ftp://ftp.oswego.edu> nella directory 'pub/unix/earp'.

22 PHP Hypertext Preprocessor - Linguaggio di scripting dal lato server, incorporabile nel codice html, per PostgreSQL

Strumenti di Interfaccia al WWW sono reperibili presso:

- <http://www.php.net>
- <http://www.vex.net/php>

Per PHP esiste anche un compilatore chiamato Zend, che ne migliora immensamente le prestazioni. Prima di tutto scriverete la vostra applicazione nel linguaggio di scripting PHP, passando attraverso le fasi di sviluppo, collaudo e correzione degli errori. Quando il progetto sarà pronto per l'utilizzo, userete il compilatore Zend per compilare il codice PHP, così da creare eseguibili che gireranno molto rapidamente.

Il vecchio nome di PHP è Professional Home Pages, mentre il nuovo nome è PHP Hypertext Pre-Processor

- Siti Mirror si trovano in molti paesi come www.CODICEPAESE.php.net
- <http://www.fe.de.php.net>
- <http://www.sk.php.net>
- <http://php.iquest.net/>
- L'e-mail per eventuali domande è: *rasmus@lerdorf.on.ca*

PHP è un linguaggio di scripting dal lato server, incorporato nel codice html. PHP permette, come Javascript, di scrivere semplici script nei file .HTML ma, al contrario di JavaScript, PHP non è dipendente dal browser. JavaScript è un linguaggio lato client, incorporabile nel codice html, mentre PHP è un linguaggio dal lato server. PHP è concettualmente simile al prodotto LiveWire Pro di Netscape. Se vi piace il software gratuito in rapida evoluzione e fornito del codice sorgente, probabilmente PHP vi piacerà.

- Il codice del supporto di PostgreSQL è stato scritto da Adam Sussman *asussman@vidya.com*

22.1 Le funzionalità più rilevanti

- Supporto per Standard CGI, FastCGI e modulo di Apache - Come programma CGI standard, PHP può essere installato su qualsiasi macchina Unix su cui giri qualsiasi web server per Unix. Con il supporto per il nuovo standard FastCGI, PHP può trarre vantaggio dall'incremento di velocità ottenuto attraverso questo meccanismo. Come modulo di Apache, PHP diventa un'alternativa alla programmazione CGI estremamente potente, e dotata di **velocità fulminante**.
- Registrazione degli Accessi - Con la proprietà della registrazione degli accessi di PHP, gli utenti possono conservare i propri conteggi degli accessi e delle registrazioni. Per questo risultato non sono utilizzati in alcun modo i file di registrazione d'accesso del sistema centrale, e viene fornito un controllo degli accessi in tempo reale. Lo script di visualizzazione del Log fornisce un veloce riassunto degli accessi ad un insieme di pagine, appartenenti ad un singolo utente. In aggiunta a ciò, il pacchetto può essere configurato per generare un titolo di coda per ogni pagina che mostri informazioni sugli accessi. (...)

- Controllo degli Accessi: Un meccanismo di filtraggio, basato sul web, gestisce la configurazione del controllo d'accesso. È possibile creare regole per tutte le pagine web che appartengano ad una certa persona, la quale voglia porre diverse restrizioni su chi possa vedere queste pagine, e su come esse saranno viste. Le pagine possono essere o protette da password, o completamente inaccessibili, oppure con registrazione disabilitata, o potranno avere altre forme di controllo ancora; dipende dal dominio del client, dal browser, dall'indirizzo e-mail, o anche dal documento di provenienza.
- Supporto a PostgreSQL - Postgres è un evoluto RDBMS, libero e gratuito. PHP supporta le "query SQL" di PostgreSQL, incorporandole direttamente nei file .html .
- RFC-1867 Supporto per il File Upload - Il File Upload è una nuova funzionalità di Netscape 2.0. Essa permette agli utenti di effettuare il trasferimento di file verso un server web. PHP fornisce la decodifica Mime per compiere quest'operazione, e fornisce anche una procedura di base per realizzare qualcosa di utile con i file, una volta che siano stati ricevuti.
- Controllo dell'autenticazione basata su HTTP - PHP può essere usato per creare meccanismi di autenticazione personalizzati, basati su HTTP, per il server web Apache.
- Variabili, Array, Array Associativi - PHP supporta diversi tipi di variabili, gli array, ed anche gli array associativi, simili a quelli del Perl. I valori assunti da tutti questi tipi di dati possono essere passati da una pagina web ad un'altra, usando form con i metodi GET o POST.
- Condizionali, Cicli con While - PHP supporta un linguaggio di scripting simile al C, e pieno di funzionalità. Ciò che della pagina web dovrebbe essere visualizzato, viene deciso con l'uso delle strutture condizionali if/then/elseif/else/endif, dei cicli con while, e dell'istruzione switch/case.
- Espressioni Regolari Estese - Le espressioni regolari vengono intensamente utilizzate per pattern matching (comparazione con modelli di stringa, n.d.t.), per sostituzione di modelli (di stringa, n.d.t.), e per manipolazione di stringhe in generale. PHP supporta tutte le comuni operazioni realizzate con le espressioni regolari.
- Controllo degli Header Raw HTTP - La capacità delle pagine web di poter inviare header raw HTTP, in base a qualche condizione, è essenziale per la progettazione ad alto livello dei siti web. È di uso frequente l'invio di una Location: l'header URL redirige il client chiamante verso qualche altro URL. Può essere anche impiegato per disabilitare il caching, oppure per manipolare l'ultimo aggiornamento dell'intestazione delle pagine.
- Creazione di immagini GIF in tempo reale - PHP supporta la libreria di immagini GD di Thomas Boutell, che rende possibile generare "al volo" immagini GIF.
- Supporto al "Safe Mode" ISP - PHP supporta una singolare modalità di protezione, "Safe Mode", che rende sicura l'esecuzione degli script PHP da parte di molti utenti presenti sullo stesso server.
- Molte altre funzionalità vengono aggiunte in ogni nuova versione di PHP. Visitate il sito web principale presso `<http://www.php.net>`
- È libero e gratuito! - Questa è una qualità fondamentale e decisiva. Il pacchetto è completamente libero. È posto sotto la licenza GNU/GPL, che consente di usare il software per qualsiasi proposito, che sia commerciale o no.

22.2 PHP - Breve Storia

Il PHP ha preso vita come un piccolo, semplice wrapper (incapsulatore) cgi scritto in Perl. Il nome di questo primo pacchetto era Personal Home Page Tools, che più tardi si trasformò in Personal Home Page Construction Kit.

Per includere facilmente le query SQL nelle pagine web, fu scritto un tool apposito. Quest'ultimo era, fondamentalmente, un altro wrapper CGI che analizzava sintatticamente le query SQL, e che ha reso facile la creazione di form e tabelle basate su queste query. Questo strumento è stato denominato FI (Form Interpreter).

La versione 2.0 di PHP/FI è una riscrittura completa di questi due pacchetti combinati in un singolo programma. PHP è evoluto verso un semplice linguaggio di programmazione, inglobato dei file HTML. PHP elimina la necessità dei tanti piccoli programmi cgi Perl, permettendovi di posizionare semplici script direttamente nei vostri file HTML. Ciò accelererà le prestazioni complessive delle vostre pagine web, poiché viene eliminato l'overhead dovuto alle numerose biforcazioni (forking) di Perl. PHP rende anche più facile gestire siti web estesi, posizionando tutti i componenti di una pagina web in un singolo file html. Includendo il supporto per vari database, viene anche reso banale lo sviluppo di pagine web con accesso ai database. Molte persone trovano che l'incorporazione di codice sia molto più facile da gestire, rispetto alla creazione di file HTML e CGI separati.

Oggi PHP/FI è stato rinominato PHP.

22.3 Ebbene, cosa posso fare con PHP ?

La prima cosa su cui porre attenzione, se volete eseguire una pagina con PHP, è che viene aggiunto un titolo, a piede di pagina, con informazioni riguardanti il numero di volte in cui la vostra pagina è stata visitata (purché nel binario sia stata compilata la gestione della registrazione degli accessi). Questa è solo una piccolissima parte di ciò che PHP può fare per voi. Infatti gioca un altro importantissimo ruolo come interprete di form cgi, che rende conto della parte FI del vecchio nome. Per esempio, se create un form su una delle vostre pagine web, avrete bisogno di qualcosa per elaborare l'informazione raccolta da quel form. Anche se volete soltanto passare l'informazione ad un'altra pagina web, dovrete avere un programma cgi che lo faccia per voi. PHP rende estremamente facile ricevere i dati del form, e lavorare con essi.

22.4 Un semplice esempio

Supponete di avere un form:

```
<FORM ACTION="/cgi-bin/php.cgi/~userid/display.html" METHOD=POST>
<INPUT TYPE="text" name="name">
<INPUT TYPE="text" name="age">
<INPUT TYPE="submit">
<FORM>
```

Il vostro file display.html potrebbe allora contenere qualcosa come:

```
<?echo "Ciao $name, hai $age anni!<p>" >
```

È questa la semplicità! PHP crea automaticamente una variabile per ogni campo di input del form. Potete così usare queste variabili nel file dell'URL, che compare come argomento di ACTION.

Il passo successivo, una volta che avete compreso come usare le variabili, è iniziare a inserire qualche tag per la gestione del flusso logico, nelle vostre pagine. Per esempio, se volete visualizzare diversi messaggi al variare dell'input dell'utente, dovreste usare le istruzioni logiche if/else. Sulla scorta dell'esempio sopra riportato, possiamo visualizzare diverse cose, a seconda dell'età che l'utente inserisce. Cambiamo il nostro display.html come segue:

```
<?
    if($age>50);
```

```
            echo "Ciao $name, sei anziano!<p>";
        elseif($age>30);
            echo "Ciao $name, sei molto vecchio!<p>";
        else;
            echo "Ciao $name.";
        endif;
    >
```

PHP fornisce un linguaggio di scripting molto potente, che realizzerà molto più di quanto il precedente esempio dimostri. Vedere la sezione relativa al Linguaggio Script PHP per altre informazioni.

Potete usare PHP anche per configurare i permessi d'accesso alle vostre pagine. Ciò può essere fatto usando un meccanismo incorporato di filtraggio (nella struttura del PHP n.d.t). Con esso potreste, per esempio, specificare che solo persone di certi domini abbiano accesso alla visualizzazione delle vostre pagine; oppure potreste creare una regola che protegga con password certe pagine. Vedere la sezione relativa al Controllo d'Accesso per altri dettagli.

PHP è anche in grado di ricevere file da qualsiasi browser web, aderente alle specifiche RFC-1867. Questa caratteristica permette che si possa fare il caricamento sia di testi, sia di binari. Col controllo d'accesso del PHP e le sue funzioni logiche, avete pieno controllo su chi abbia il permesso di caricare un file su server, e su cosa possa essere fatto con il file una volta che sia stato ricevuto. Leggere la sezione File Upload per altri dettagli.

PHP ha il supporto per il pacchetto del database PostgreSQL. PHP supporta query SQL inserite nei vostri file .HTML .

PHP fornisce anche supporto per il pacchetto di database mysql. PHP supporta le query SQL inserite nei vostri file .HTML .

22.5 Ridirezione CGI

22.5.1 Note su Apache 1.0.x

Impiegare un modulo di ridirezione cgi, con il server Apache, è un buon modo di eseguire PHP. Notate che non dovete preoccuparvi dei moduli di ridirezione, se state usando PHP nella forma di modulo di Apache. Sono disponibili due di questi moduli di ridirezione. Uno di questi è stato sviluppato da Dave Andersen

- *angio@aros.net*

ed è reperibile presso

- `<ftp://ftp.aros.net/pub/util/apache/mod_cgi_redirect.c>`

L'altro è distribuito con Apache, ed è chiamato mod_actions.c. I due moduli sono molto simili. Differiscono leggermente nel loro uso. Sono stati testati entrambi, ed entrambi funzionano con PHP.

Cercate, nella documentazione di Apache, le informazioni su come aggiungere un modulo. Generalmente si aggiunge il nome del modulo in un file chiamato Configuration. Se volete usare il modulo mod_actions, la riga da aggiungere è:

Module action_module mod_actions.o

Se state usando il modulo mod_cgi_redirect.c , aggiungete questa riga:

Module cgi_redirect_module mod_cgi_redirect.o

Poi compilate il vostro httpd, ed installatelo. Per configurare la ridirezione cgi dovrete o creare un nuovo tipo mime nel vostro file mime.types, oppure potete usare il comando AddType nel vostro file srm.conf, per aggiungere il tipo mime. Il tipo mime da aggiungere dovrebbe essere qualcosa come questo:

```
application/x-httpd-php phtml
```

Se state usando il modulo mod_actions.c, dovrete aggiungere la seguente riga al vostro file srm.conf :

```
Action application/x-httpd-php /cgi-bin/php.cgi
```

Se state usando mod_cgi_redirect.c, dovreste aggiungere questa riga a srm.conf:

```
CgiRedirect application/x-httpd-php /cgi-bin/php.cgi
```

Non provate ad usare, nello stesso tempo, mod_actions.c e mod_cgi_redirect.c.

Una volta che uno di questi moduli di ridirezione cgi siano stati installati e configurati correttamente, sarete in grado di far eseguire l'analisi sintattica del PHP sui file di estensione .phtml. Per di più, se aggiungete index.phtml alla riga di configurazione chiamata DirectoryIndex del vostro file srm.conf, e se il vostro file index si chiama index.phtml, allora la pagina di massimo livello in una directory sarà automaticamente processata da php.

22.5.2 HTTPD di Netscape

Potete automaticamente ridirigere le richieste di file con una data estensione, e farle gestire da PHP, usando il modulo Redirection CGI del Server Netscape. Questo modulo può essere trovato negli Archivi di File dell'Home Page di PHP. Il file README nel pacchetto, spiega esplicitamente come configurarlo per usarlo con PHP.

22.5.3 HTTPD di NCSA

Attualmente NCSA non supporta i moduli; così, volendo effettuare ridirezioni cgi con questo server, dovrete modificare il codice sorgente del server. Negli archivi di file di PHP esiste una patch, adatta allo scopo, per NCSA 1.5

22.6 Eseguire PHP dalla riga di comando

Se avete compilato la versione CGI di PHP, potete usarla dalla riga di comando semplicemente digitando: php.cgi nome_file, dove nome_file è il file che volete processare. Potete anche creare script PHP autonomi inserendo, sulla prima riga del vostro script, quanto segue:

```
#!/usr/local/bin/php.cgi -q
```

La "-q" sopprime la stampa degli header HTTP. Potete anche non mettere quest'opzione, se volete.

22.7 Pacchetto PHPGem

PHPGem è uno script PHP che accelera la creazione di script PHP specifici per lavorare con le tabelle. Funziona con diversi server SQL come PostgreSQL, MySQL, mSQL, ODBC, e Adabas. Voi inserite una descrizione e i parametri per i campi delle vostre tabelle (nome del campo, possibilità o no di ricerca nel campo, ecc.), e PHPGem fornisce in output un altro script PHP, che lavorerà con le tabelle (per visualizzazione / inserimento / modifica / cancellazione / duplicazione e per ricerche). PHPGem funziona con tabelle annidate in livelli multipli. Inoltre è consentito indicare un livello d'accesso per ogni tabella, per ogni campo, e per ogni utente. PHPGem supporta anche le immagini.

PHPGem si trova presso `<http://sptl.org/phpgem>`

23 Interfaccia di Python a PostgreSQL

Python è un linguaggio di scripting interpretato, orientato agli oggetti. È di semplice uso (la sintassi è leggera, semplice, e le istruzioni sono semplici e lineari), ed ha molte estensioni per costruire GUI, interfacce al WWW, ecc. Un browser web intelligente (simile a HotJava) è attualmente in sviluppo (da novembre 1995), e ciò dovrebbe aprire molte porte ai programmatori. Python è sotto copyright dello Stichting S Mathematisch Centrum di Amsterdam, nei Paesi Bassi, ed è liberamente distribuibile. Esso contiene il supporto per il caricamento dinamico di oggetti, classi, moduli, ed eccezioni. Aggiungere interfacce alle nuove librerie di sistema, per mezzo del codice C, è semplice ed immediato, e rende Python facile da usare per realizzazioni su misura. Python è un linguaggio di scripting ad altissimo livello con interfaccia a X. Il pacchetto di Python è distribuito con i cdrom di Linux; nel pacchetto è compresa la maggior parte dei moduli standard di Python, tra cui i moduli di interfaccia a Tix, un insieme di widget per TK.

PyGreSQL è un modulo python che si interfaccia ad un database PostgreSQL. In esso è acclusa la libreria di query per PostgreSQL; questa permette un facile utilizzo delle potenti caratteristiche di PostgreSQL, dall'interno di uno script Python. PyGreSQL è stato scritto da D'Arcy J.M. Cain e Pascal Andre.

- Nuovo sito di PyGreSQL `<http://www.druid.net/pygresql/>`
- Curato da D'Arcy presso `<http://www.druid.net/~darcy/>`
- Il sito non aggiornato è presso `<ftp://ftp.via.ecp.fr/pub/python/contrib/Database/PyGres95.README >`
- D'Arcy J.M. Cain *darcy@druid.net*
- Pascal Andre *andre@chimay.via.ecp.fr*
- Pascal Andre *andre@via.ecp.fr*

23.1 Dove ottenere PyGres ?

I siti base dei vari pacchetti sono:

- Python `<ftp://ftp.python.org:/pub/www.python.org/1.5/python1.5b2.tar.gz>`
- PyGreSQL `<ftp://ftp.druid.net/pub/distrib/PyGreSQL-2.1.tgz>`
- Sito obsoleto `<ftp://ftp.via.ecp.fr/pub/python/contrib/Database/PyGres95-1.0b.tar.gz >`

Dovreste comunque ricercare qualche altro sito mirror più vicino a voi. Riferitevi alle vostre fonti di informazione, per trovare questi siti. PyGreSQL dovrebbe risiedere nelle directory contrib dei siti di Python e PostgreSQL.

23.2 Informazioni e supporto

Se vi servono informazioni riguardanti questi pacchetti, visitate i loro siti web:

- Python : `<http://www.python.org/>`
- PostgreSQL : `<http://epoch.cs.berkeley.edu:8000/postgres95/index.html>`
- PyGreSQL `<ftp://ftp.druid.net/pub/distrib/PyGreSQL-2.1.tgz>`
- Sito di PyGreSQL non aggiornato : `<http://www.via.ecp.fr/via/products/pygres.html>`

Per il supporto:

- Mailing list per PyGreSQL. Potete iscrivervi inviando un'email a *majordomo@vex.net* con la riga "subscribe pygresql nome@dominio" nel corpo, mettendo il vostro indirizzo di posta elettronica al posto di "nome@dominio".
- Newsgroup per Python : newsgroup comp.lang.python
- PyGreSQL : prendete contatto con Andre, all'indirizzo *andre@via.ecp.fr* , per segnalazioni di bug, idee, annotazioni

23.3 Provare l'interfaccia di Python

Vedere la sezione: 6.5 (Provare l'interfaccia di Python a PostgreSQL)

24 WDB-P95 - Interfaccia di collegamento tra PostgreSQL e il WWW

24.1 Informazioni su wdb-p95

WDB-P95 - Un'interfaccia Web ai Database PostgreSQL, è stata creata da J. Douglas Dunlop. Si trova presso

- Il nuovo WDB di J Rowe è in `<http://www.lava.net/beowulf/programming/wdb>`
- Le nuove versioni di WWW-WDB sono presso `<http://www.eol.ists.ca/~dunlop/wdb-p95/>`
- Per domande, o per iscriversi alle Mailing list, contattate *dunlop@eol.ists.ca*

Questa è una versione modificata di wdb-1.3a2, che fornisce a PostgreSQL un'interfaccia al WWW. Questa versione richiede anche un Browser in grado di gestire le tabelle HTML, per ottenere un output tabellare. Questo requisito non vale per il wdb originale, ed è abbastanza facile ripristinare le impostazioni originali.

Potete provare CASI Tape e Image Query. Date una sbirciata al File di Definizione dei Form (Form Definition File, FDF), usato per creare il CASI Tape ed anche Image Query, che include una JOIN di 2 tabelle.

Questa versione contiene tutti i file necessari per installare ed eseguire WDB-P95, come interfaccia ai vostri database PostgreSQL. Creare una versione di questo sistema su un altro database, dovrebbe essere relativamente facile - A patto che supporti lo standard SQL e abbia un'interfaccia Perl.

24.2 Il server PostgreSQL, pgperl, e httpd, devono essere sullo stesso host?

No, il server PostgreSQL non deve essere sullo stesso host: se WDB-P95 è richiamato dal demone http, allora devono trovarsi sullo stesso host - E se WDB-P95 è stato scritto per usare Pg.pm, anche pgperl deve essere sullo stesso host. Pgperl è stato scritto usando la libreria libpq, e così sarà in grado di accedere a qualsiasi server PostgreSQL posto ovunque nella rete, proprio come qualsiasi altro client di PostgreSQL. Quanto segue illustrerà il concetto:

(WWW Client (Netscape)) => (HTTP Server (NCSA's http) + WDB-P95 + pgperl + libpq)=> (PostgreSQL server)

Le parentesi tonde () rappresentano le macchine.

Ogni macchina può essere di diverso tipo: NT, SUN, HP, ... ma vi occorre la libreria di interfaccia libpq per il tipo di macchina che ospiterà WDB-P95, quindi dovrete compilare pgperl. (Il sistema è stato progettato per usare le tabelle HTML, perciò si dovrà utilizzare un client WWW recente)

25 Interfaccia dei linguaggi "C", "C++", ESQL/C, a PostgreSQL, ed operatori Orientati ai bit

25.1 Interfaccia "C"

È inclusa nella distribuzione, ed è chiamata 'libpq'. È simile alle librerie Oracle OCI, Sybase DB-lib, Informix CLI.

25.2 Interfaccia "C++"

È inclusa nella distribuzione, ed è chiamata 'libpq++'. Vedere la sezione: 6.7 (Provare l'interfaccia di C e C++ a PostgreSQL)

25.3 ESQL/C

ESQL/C significa 'Embedded C Pre-compiler' (Pre-compilatore C incorporato). ESQL/C, è l'analogo di Oracle Pro*C, Informix ESQL/C. PostgreSQL ESQL/C è un'application-programming interface (API) SQL, che consente al programmatore C di creare applicazioni specifiche con gestione dei database. PostgreSQL ESQL/C vi permette di usare un linguaggio di terza generazione a voi familiare, e di continuare a trarre vantaggio dallo Structured Query Language (SQL).

ESQL/C consiste dei seguenti software:

- Le ESQL/C sono librerie di funzioni C, e forniscono accesso al server database.
- I file header di ESQL/C forniscono le definizioni per le strutture dei dati, per le constanti, e le macro utili al programma ESQL/C.
- Il preprocessore ESQL/C è un preprocessore di codice sorgente, che converte un file C con istruzioni SQL, in un file eseguibile.

Dove trovarlo:

- ESQL/C per PostgreSQL è già incluso nella distribuzione.

- Sito principale <ftp://ftp.lysator.liu.se/pub/linus>
- Email : *linus@epact.se*

Vedere la sezione:6.9 (Provare l'interfaccia di Embedded SQL/C a PostgreSQL)

Per elaborare i file 'ecpg' (*.pgc), utilizzando la colorizzazione dell'editor Vim, dovete fare quanto segue:-

```
bash$ su - postgres
bash$ mkdir $HOME/vim
Ora create un file '$HOME/vim/myfilestypes.vim' con le seguenti righe

    " myfiletypefile
        au! BufRead,BufNewFile *.pgc    set filetype=esqlc
```

Dovreste avere il vostro file $HOME/.gvimrc . Altrimenti createne uno; riferitevi al documento "Vim-howto", reperibile presso <http://metalab.unc.edu/LDP/HOWTO/Vim-HOWTO.html> . Inserite le seguenti righe nel file $HOME/.gvimrc

```
let myfiletypefile = "~/vim/myfiletypes.vim"
```

Ora aprite una sessione di vim con il comando

```
bash$ gvim sample.pgc
```

e otterrete la sintassi evidenziata mediante colorizzazione.

25.4 Operatori Orientati ai bit per PostgreSQL

Gli Operatori Orientati ai bit sono stati scritti da Nicolas Moldavsky

- *nico@overnet.com.ar*

Sono funzioni "C", che implementano gli operatori orientati ai bit (AND, OR, XOR, operatore complemento) su pgsql. Prelevatele con un FTP anonimo dal sito

- <ftp://ftp.overnet.com.ar/pub/utils/linux/bitpgsql.tgz>

È incluso un Makefile per Linux.

26 Codice Giapponese Kanji per PostgreSQL

Si trova presso il seguente sito

- <ftp://ftp.sra.co.jp/pub/cmd/postgres/>

27 Versione di PostgreSQL per Windows 95/Windows NT

I binari di PostgreSQL per Windows NT, sono disponibili presso:

- Binari di PostgreSQL per Windows NT `<http://www.askesis.nl>`
- `<http://www.postgresql.org>`

Scaricate i binari, scompattateli, e seguite le istruzioni descritte in 27.6 (Installazione di PostgreSQL) partendo dal passo 13.

Se volete ri-compilare l'albero dei sorgenti, seguite le istruzioni più sotto. La versione per NT è stata realizzata con il pacchetto cygwin32 di Cygnus, che contiene gcc, gmake, per Win NT/95.

- Il pacchetto Cygwin 32 si trova presso `<http://www.cygnus.com/misc/gnu-win32>`

Presso questo sito potete anche ottenere il file cdk.exe (il file auto estraente di gnu-win32)

27.1 Autori della versione per NT

Gli autori della versione di PostgreSQL per Windows NT, sono:

- Daniel Horak *horak@mmp.plzen-city.cz*
- Joost Kraaijeveld *JKraaijeveld@askesis.nl*
- Kevin Lo *kevlo@FreeBSD.org*
- Home page della versione per NT `<http://www.freebsd.org/~kevlo/postgres/portNT.html>`

27.2 Installazione del pacchetto Cygwin

1. Scaricate `<ftp://go.cygnus.com/pub/sourceware.cygnus.com/cygwin/latest/full.exe>`
2. Eseguite full.exe ed installate nella directory c:\Unix\Root .
3. Eseguite Cygwin. Digitate 'mount –help' per leggere la documentazione. Potete usare l'opzione -f per forzare il montaggio. Poi eseguite "umount / " e "mount c:\Unix\Root /"

27.3 Messa a punto della finestra di Bash

Dopo l'installazione del pacchetto Cygwin, fate quanto segue per impostare l'ambiente di lavoro:

1. Installate l'editor 'Vim', evoluzione di Vi. Vedere `<http://metalab.unc.edu/LDP/HOWTO/Vim-HOWTO.html>`

2. La finestra predefinita della bash di cygwin, è una finestra di 24 righe con sfondo nero. Per impostare il colore di sfondo e le dimensioni della finestra di bash, fate click su NT-Start->Control-panel->MS DOS; cambiate il colore di sfondo nel colore grigio, e portate la lunghezza della finestra a 70 righe.

(OPPURE) fate click col tasto destro sulla barra del titolo della finestra, e cambiate le proprietà.

3. Aprite in edit cygnus.bat , situato nella directory c:\cygnus\cywinb20 , ed impostate quanto segue:

```
set HOME=c:\cygnus\cygwinb20
bash --login
```

Elaborate anche /.bash_profile , e mettete queste righe

```
set -o vi
PATH=$PATH:/usr/local/bin:/usr/bin
export PATH
```

4. Per abilitare l'elaborazione dello storico della riga di comando, impartite il comando:

bash$ set -o vi

Usando i tasti l,k,j,h potete usare i comandi di vi per elaborare lo storico della riga di comando. Potete così ripetere o modificare i comandi precedentemente digitati, risparmiando tempo nella digitazione.

5. Potete effettuare il montaggio di dispositivi/directory usando questo comando:

```
bash$ umount /
bash$ mount "c:\cygnus"  /
bash$ mount "c:\cygnus\cygwin-b20\postgres" /usr/local/pgsql
```

6. Leggete l'help in linea con:

```
bash$ mount --help
bash$ ls --help
```

27.4 Installazione dei tool di Andy Piper

1. Recatevi presso `<ftp://ftp.xemacs.org/pub/xemacs/aux/>` e scaricate cygwin-b20-local.tar.bz2 nella directory c:/Unix/Root .
2. cd c:/Unix/Root; bunzip2 cygwin-b20-local.tar.bz2
3. tar -xvf cygwin-b20-local.tar
4. cd /local/bin; sh check_cygwin_setup.sh
5. Dopo aver superato il passo 4, vedrete il seguente messaggio:

```
You don’t have /bin would you like to mount cygwin as /bin?"
[ y/n ]
Selezionate ’n’, e alle altre opzioni va selezionato ’y’
```

6. mount c:/Unix/Root/cygwin-b20/H-i586-cygwin32/i586-cygwin32/bin /bin
7. cd c:/Unix/Root/cygwin-b20/H-i586-cygwin32/i586-cygwin32; mkdir libexec share man etc sbin info
8. cp -R /local/{ bin,libexec,share,man,etc,sbin,info,include }

27.5 Installazione del pacchetto Cygwin32 IPC di Ludovic Lange

1. Recatevi presso <http://www.multione.capgemini.fr/tools/pack_ipc> e scaricate cygwin32_ipc-1.03.tgz nella directory c:/Unix/Root .
2. tar -zxvf cygwin32_ipc-1.03.tgz
3. cd cygwin32_ipc-1.03/src ed eseguite 'make'
4. mkdir -p c:/usr/local/{bin,include,lib,include/sys}

```
cp /cygwin32_ipc-1.03/bin/* c:/usr/local/bin
cp /cygwin32_ipc-1.03/include/sys/* c:/usr/local/include/sys
cp /cygwin32_ipc-1.03/lib/* c:/usr/local/lib
cp c:/usr/local/bin/* /bin
cp c:/Unix/Root/cygwin-b20/H-i586-cygwin32/bin/* /bin
```

5. mount c:/usr/local/bin /usr/local/bin

```
mount c:/usr/local/include /usr/local/include
mount c:/usr/local/lib /usr/local/lib
cp /local/lib/* /usr/local/lib
```

27.6 Installazione di PostgreSQL

1. Scaricate l'ultima versione del codice sorgente di PostgreSQL
2. Postgres tratta tutti i file come binari, ed appaiono quindi i caratteri lf/cr, perciò eseguiamo le azioni 2, 3, 4, e 5:

```
mkdir -p c:/Postgres/{Source,Binary}
mkdir c:/Postgres/Binary/pgsql
mkdir -p /usr/src/pgsql
mkdir -p /usr/local/pgsql
```

3. Copiate il codice sorgente di Postgres nella directory c:/Postgres/Source , poi

```
tar -zxvf postgresql-6.5.3.tar.gz
```

4. mv postgresql-6.5.3 pgsql
5. Ora montate le directory:

```
mount c:/Postgres/Source/pgsql /usr/src/pgsql
mount c:/Unix/Root/cygwin-b20/share /sw/cygwin-b20/share
```

6. mkdir -p /usr/local/pgsql/{bin,include,lib,data}
7. cd /usr/src/pgsql/src/win32
8. Copiate i file header:

```
cp un.h c:/Unix/Root/cygwin-b20/H-i586-cygwin32/i586-cygwin32/include/sys
cp endian.h c:/Unix/Root/cygwin-b20/H-i586-cygwin32/i586-cygwin32/include
cp tcp.h c:/Unix/Root/cygwin-b20/H-i586-cygwin32/i586-cygwin32/include/netinet
```

9. ln -s /usr/local/lib /usr/src/pgsql/src/backend/libpostgres.a
10. cd /usr/src/pgsql/src, poi eseguite './configure'
11. make > make.txt 2>&1
12. make install > make.install.txt 2>&1
13. cp /usr/local/pgsql/lib/pq.dll /usr/local/pgsql/bin
14. Dopo il make install dovrete cambiare tutti i file di testo nella directory bin e lib, in modo che essi non contengano cose come cr/lf ed eof.
15. Utilizzare un qualsiasi editor per creare .bashrc nella directory /, come segue:

```
PATH=$PATH:/usr/local/pgsql/bin:/usr/local/bin
PGDATA=/usr/local/pgsql/data
PGLIB=/usr/local/pgsql/lib
LD_LIBRARY_PATH=/usr/local/pgsql/lib:/usr/local/lib
export LD_LIBRARY_PATH PATH PGDATA PGLIB
```

16. source /.bashrc, poi eseguite 'initdb –username=xxxx' Notate che il proprietario del sistema DB deve essere diverso dal root/amministratore
17. Modificate il file /usr/local/pgsql/data/pg_hba.conf, come segue:

```
host        all     163.17.11.109   255.255.255.0   trust
```

18. ipc-daemon.exe&
19. postmaster -i&
20. Eseguite ' psql -h nome_host template1'

28 Mailing List

28.1 Account di E-mail per PostgreSQL

Ottenete un account gratuito di e-mail da

- Yahoo `<http://www.yahoo.com>` fate click su e-mail
- Lycos `<http://www.lycos.com>` fate click su new e-mail accounts
- hotmail `<http://www.hotmail.com>` fate click su new e-mail accounts

Sottoscrivendo (subscribe) la mailing list di PostgreSQL con Yahoo, noterete che Yahoo ha funzionalità aggiuntive, come la creazione di cartelle separate da riservare alle e-mail su PostgreSQL, per non mescolarle con le vostre e-mail personali. Selezionate il menu Email- > Options- > Filters e scegliete "seperate folder" per le email. Con questo account di e-mail, potete accedere alla posta (elettronica, ndt) da qualsiasi luogo del mondo, purché abbiate accesso ad una pagina web.

Se avete una qualsiasi altra e-mail, potete usare "Mail Filters" per ricevere automaticamente le mail di PostgreSQL in una cartella separata. Ciò vi permetterà di gestire la posta in maniera ordinata.

28.2 Mailing List in lingua Inglese

Vedere l'argomento Mailing Lists sulla pagina web principale situata presso:

- `<http://www.postgresql.org/>`
- Inviate domande a: *pgsql-questions@postgresql.org*
- Per sviluppatori: *pgsql-hackers@postgresql.org*
- Domande su versioni specifiche per le altre piattaforme: *pgsql-ports@postgresql.org*
- Domande relative alla documentazione: *pgsql-docs@postgresql.org*

Otterrete le risposte/repliche via e-mail in meno di un giorno.

Potete anche sottoscrivervi alle mailing list. Per sottoscrivervi o sganciarvi dalla lista, inviate una mail a:

- *pgsql-questions-request@postgresql.org*
- *pgsql-hackers-request@postgresql.org*
- *pgsql-ports-request@postgresql.org*
- *pgsql-docs-request@postgresql.org*

Il corpo del messaggio deve contenere la singola riga

subscribe

(oppure)

unsubscribe

28.3 Archivio di Mailing List

Le mailing list sono anche archiviate, in formato html, nella seguente locazione:

- Date-wise listing, disponibile per mezzo di MHonarc, via WWW, presso `<http://www.postgresql.org/mhonarc/pgsql-questions>`
- `<ftp://ftp.postgresql.org>` e la directory è /pub/majordomo

Esiste anche un motore di ricerca specifico per domande su pgsql, ed è a disposizione sul sito web principale di PostgreSQL.

28.4 Mailing List Spagnola

Ora esiste una mailing list "non ufficiale" su postgreSQL in lingua spagnola. Per sottoscriversi, l'utente deve inviare un messaggio a:

- *majordomo@tlali.iztacala.unam.mx*

Il corpo del messaggio deve contenere la riga seguente: inscripcion pgsql-ayuda

29 Documentazione e Libri di Riferimento

29.1 Guide per l'Utente e Manuali

La documentazione che segue è inclusa nella distribuzione di PostgreSQL, nei formati postscript, HTML, e "pagine man" di unix. È collocata nella directory /usr/doc/postgresql* . Se avete accesso ad internet, potete trovare questi documenti presso `<http://www.postgresql.org/docs>` e presso `<http://www.postgresql.org/users-lounge/docs>`.

- "Installation Guide"
- "User Guide" per PostgreSQL
- "Implementation Guide", che illustra PostgreSQL dettagliatamente ed in profondità.
- Manuali In Linea.
- Manuali In Linea (formato HTML).
- Esistono anche manuali in formato Postscript per ottenere copie stampate.

29.2 Documentazione disponibile in linea

- Elenco e descrizione dei tipi di dati ed operatori predefiniti

```
È parte del comando PSQL
```

- Elenco delle parole chiave SQL supportate

```
C'è uno script nella directory /tools dell'albero del codice sorgente.
```

- Elenchi delle istruzioni supportate:

```
Usate il comando psql \h
```

- Concetti elementari sui database relazionali sotto PostgreSQL (implementazione), e svariati esempi (di interrogazioni) disponibili in linea (potete vedere anche 6.4 ()):

```
Date un'occhiata ai test di regressione in src/test. Vi trovate le directory
regress/sql e suite/*.sql
```

- Tutorial su PostgreSQL.

```
script tutorial su SQL si trovano nella directory src/tutorial
```

Vedere anche "Tutorial SQL per principianti", nell'B () di questo documento.

29.3 Libri di Riferimento utili

- "Understanding the New SQL: A Complete Guide" - di Jim Melton e Alan R.Simon

```
Edito da Morgan Kaufman, è uno dei migliori libri su SQL. Tratta di SQL92.
```

- "A Guide to THE SQL STANDARD" - di C.J.Date

```
della Addison-Wesley Publishing, è un altro buon libro.  È un testo su SQL
molto diffuso.
```

- SQL - The Standard Handbook, Novembre 1992

```
di Stephen Cannan e Gerard Otten
McGraw-Hill Book Company Europe , Berkshire, SL6 2QL, England
```

- SQL Instant Reference, 1993

```
di Martin Gruber, Redattore Tecnico: Joe Celko
SYBEX Inc.  2021 Challenger Drive Alameda, CA 94501
```

- C.J.Date, "An introduction to Database Systems" (sesta edizione), Addison-Wesley, 1995, ISBN 0-201-82458-2

```
Questo libro è la Bibbia dei Sistemi di Gestione dei Database.
Il libro illustra dettagliatamente normalizzazione, SQL,  ripristino, con-
correnza, sicurezza, integrità,  ed estensioni al modello relazionale ori-
ginale; inoltre tratta le attuali questioni riguardanti i sistemi client /
server, ed  i/l modelli/o  Orientati/o  agli Oggetti.  Per approfondire la
lettura, sono stati inclusi molti riferimenti.  Lo raccomando alla maggior
parte degli utenti.
```

- Stefan Stanczyk, "Theory and Practice of Relational Databases", UCL Press Ltd, 1990, ISBN 1-857-28232-9

```
Il libro tratta nei dettagli la teoria dei database relazionali, l'algebra
relazionale, il calcolo relazionale,  e la  normalizzazione. Ma non tratta
di casi  del mondo reale,  e gli esempi non vanno oltre quelli semplici. È
consigliato per la maggior parte degli utenti.
```

- "The Practical SQL Handbook" Terza Edizione, Addison Wesley Developers Press ISBN 0-201-44787-8

```
Lo consiglio alla maggior parte degli utenti.
```

- Michael Stonebraker, "Readings in Database Systems", Morgan Kaufmann, 1988, ISBN 0-934613-65-6

```
Questo libro  è una raccolta di articoli,  pubblicati negli  anni passati,
dedicati ai database. Non è un libro per lettori occasionali: si tratta di
di un testo di consultazione per studenti avanzati (laureati),  o per svi-
luppatori di sistemi di database.
```

- C.J.Date, "Relational Database - Selected Readings", Addison-Wesley, 1986, ISBN 0-201-14196-5

```
Il libro è una raccolta di articoli,  pubblicati negli anni passati, dedi-
cati ai database. Non è materia per semplici utenti: si tratta di un testo
di consultazione  per studenti avanzati (laureati),  o per sviluppatori di
sistemi di database.
```

- Nick Ryan e Dan Smith, "Database Systems Engineering", International Thomson Computer Press, 1995, ISBN 1-85032-115-9

```
Questo libro entra nei dettagli dei metodi d'accesso,  e nelle tecniche di
archiviazione.
```

- Bipin C. Desai, "An introduction to Database Systems", West Publishing Co., 1990, ISBN 0-314-66771-7

```
Non è materia per semplici utenti, ma è per studenti avanzati (laureati) o
per sviluppatori di sistemi di database.
```

- Joe Celko "INSTANT SQL Programming"

```
Wrox Press Ltd.
Unit 16, 20 James Road, Tyseley
Birmingham, B11 2BA, England
1995
```

- Michael Gorman "Database Management Systems: Understanding and Applying Database"

```
Technology
QED and John Wiley
1991
```

- Michael Gorman "Enterprise Database for a Client/Server Environment" QED and John Wiley

```
Presenta i requisiti per  la creazione  di applicazioni database  client /
server per mezzo  di metamodelli da repository,  e l'uso di ANSI SQL stan-
dard 1993
```

Sono disponibili centinaia di altri titoli su SQL! Cercateli in una libreria.

29.4 Documenti delle specifiche SQL ANSI/ISO - SQL 1992, SQL 1998

I documenti delle specifiche ISO ANSI/ISO possono essere trovati presso i seguenti siti:

- `<http://www.naiua.org/std-orgs.html>`
- `<http://www.ansi.org/docs>` fate click sul file cat_c.html e cercate "Database SQL"
- SQL92 standard `<http://www.jcc.com>` e fate click sul file sql_stnd.html
- Specifiche ANSI/ISO SQL `<http://www.contrib.andrew.cmu.edu/~shadow/sql.html>`. Qui troverete il Reference di SQL.

29.5 Sintassi di 1992 ANSI/ISO SQL

Vedere l'A () di questo documento.

29.6 Sintassi di 1998 ANSI/ISO SQL

Le specifiche SQL 1998 (SQL 3) sono ancora in sviluppo. Vedere "Accesso Elettronico alla bozza di lavoro di SQL3" in questo documento: 29.9 ()

29.7 Tutorial su SQL per principianti

Vedere l'B () di questo documento.

29.8 Estensioni Temporali a SQL92

- Documento sulle Estensioni Temporali a SQL-92 `<ftp://FTP.cs.arizona.edu/tsql/tsql2/>`
- Specifiche SQL-3 Temporali `<ftp://FTP.cs.arizona.edu/tsql/tsql2/sql3/>`

Questa directory contiene le specifiche del linguaggio di un'estensione temporale del linguaggio SQL-92 standard. Questo nuovo linguaggio è denominato TSQL2.

Le specifiche del linguaggio qui citato costituiscono la versione finale del linguaggio.

La corrispondenza può essere diretta al Presidente del TSQL2 Language Design Committee, Richard T.Snodgrass, Department of Computer Science, University of Arizona, Tucson, AZ 85721,

- *rts@cs.arizona.edu*

Le affiliazioni, e gli indirizzi e-mail dei membri del TSQL2 Language Design Committee, possono essere trovati in una sezione separata, situata in coda alle specifiche del linguaggio. Il contenuto di questa directory è il seguente.

spec.dvi,.ps - Specifiche del Linguaggio TSQL2, pubblicate nel Settembre 1994

bookspec.ps - Specifiche del Linguaggio TSQL2, come nel libro del TSQL2, pubblicate nel Settembre 1995 (Vedere più sotto).

sql3 - proposte di modifiche sottoposte ai comitati ANSI e ISO SQL3.

Esiste, inoltre, una raccolta di commenti, abbinati alle specifiche del linguaggio, che dibattono le decisioni relative al progetto, forniscono esempi, e considerano come il linguaggio possa essere implementato. Questi commenti sono stati proposti originariamente al TSQL2 Language Design Committee. Adesso hanno un altro uso: fornire esempi dei costrutti, motivare le tante decisioni prese durante la progettazione del linguaggio, e comparare TSQL2 con i molti altri linguaggi proposti negli ultimi quindici anni. Si dovrebbe sottolineare che questi commenti non sono parte delle specifiche del linguaggio TSQL2, ma sono piuttosto un supplemento e un approfondimento delle stesse. Le specifiche del linguaggio sono le sole parole definitive su TSQL2.

I commenti, insieme alle specifiche del linguaggio, a svariati indici, ed altro materiale di supporto, sono stati pubblicati in un libro:

Snodgrass, R.T. redattore, The TSQL2 Temporal Query Language, Kluwer Academic Publishers, 1995, 674+xxiv pagine.

I commenti di valutazione, che sono presenti nel libro, sono in forma abbreviata; la forma integrale è contenuta nel file eval.ps, ospitato in questa directory

Il file tl2tsql2.pl è un programma in linguaggio prolog; esso traduce dichiarazioni logiche, consentite dal TSQL2, in costrutti validi per lo stesso TSQL2. Questo programma è stato scritto da Michael Boehlen

- *boehlen@iesd.auc.dk*

Può essere contattato per ottenere un testo descrittivo di questa traduzione. Quella trattata è, però, una versione alquanto datata di quel programma. Versioni più recenti sono disponibili presso

- `<http://www.cs.auc.dk/general/DBS/tdb/TimeCenter/Software>`

(the TimeDB and Tiger systems).

29.9 Parte 0 - Procurarsi Documenti su SQL ISO/ANSI

Questo documento illustra come procurarsi (legalmente) una copia dello standard SQL-92, e come procurarsi una copia dell'ultima bozza di SQL3.

Lo standard è protetto da copyright: lo standard ANSI è detenuto da ANSI, mentre lo standard ISO è detenuto da ISO.

Attualmente esistono due (2) standard SQL: una pubblicazione ANSI e una pubblicazione ISO. I due standard sono identici parola per parola, tranne che per alcune questioni banali come il titolo del documento, le intestazioni delle pagine, la frase "International Standard" contrapposta a "American Standard", e così via.

Acquistare lo standard SQL-92

Lo standard ISO, ISO/IEC 9075:1992, Information Technology - Database Languages - SQL, è attualmente (Marzo, 1993) disponibile alla vendita presso ANSI:

```
American National Standards Institute
1430 Broadway
New York, NY 10018 (USA)
Phone (sales): +1.212.642.4900
```

al costo di 230.00 $ USA. La versione ANSI, ANSI X3.135-1992, American National Standard for Information Systems - Database Language SQL, al momento della scrittura di questo documento, non era disponibile alla vendita, La disponibilità era attesa per il periodo tra la fine di marzo e l'inizio di maggio del 1993). Il costo previsto era di 225.00 $ USA.

Se acquistate entrambi i documenti da ANSI, avrete una maggiorazione del 7% (circa 9.10 $ USA). Le spedizioni all'estero costeranno sicuramente di più. ANSI richiede una copia stampata dell'ordine di acquisto; in alternativa, potete inviare un assegno bancario (in dollari USA ad una banca americana) che sarà sottoposto a controlli di validità e convertito in contanti prima di sbrigare l'ordine. (Esiste un'eccezione: Se la vostra organizzazione è un'azienda membro di ANSI, ANSI stessa vi spedirà la documentazione ed invierà il conto alla vostra società.)

Lo standard ISO è anche disponibile al di fuori degli USA presso le locali associazioni nazionali che sono membri sia di ISO (International Organization for Standardization), sia di IEC (International Electrotechnical Commission). Copie della lista di consociate nazionali, e dei loro indirizzi, sono disponibili presso ANSI o presso altre consociate nazionali. Sono disponibili anche presso ISO:

```
International Organization for Standardization
Central Secretariat
1, rue de Varembi
CH-1211 Genhve 20
Switzerland
```

Se preferite ordinare lo standard in un modo più comodo e rapido, dovrete pagare per avere questo privilegio. Potete ordinare ISO/IEC 9075:1992, Information Technology - Database Languages - SQL, presso:

```
Global Engineering Documents
2805 McGaw Ave
Irvine, CA 92714 (USA)
USA
Telefono (valido ovunque): +1.714.261.1455
Telefono (valido solo negli USA): (800)854-7179
```

per un costo di 308.00 $ USA. Non so se siano incluse le spese postali, ma posso immaginare che la spedizione internazionale costi di più (come minimo). In questo modo vi invieranno tempestivamente un documento, ed accetteranno anche pagamenti con le più note carte di credito. La Global non ha ancora la versione ANSI, e non dà né l'indicazione di un prezzo, né una data per la disponibilità (anche se mi aspetto che sia disponibile entro poche settimane dopo la pubblicazione di ANSI, e ad un prezzo vicino a 300.00 $ USA).

Acquistare una copia di SQL3 Working Draft (la bozza di lavoro)

Potete acquistare una copia stampata della bozza di lavoro di SQL3, presso il Segretariato ANSI X3, CBEMA (Computer and Business Equipment Manufacturers Association). Il loro proposito, è quello di conservare la "più recente" versione della bozza di lavoro di SQL3, e di venderla ad un prezzo tra i 60.00 ed i 65.00 $ USA. Potete contattare CBEMA all'indirizzo:

```
CBEMA, X3 Secretariat
Attn: Lynn Barra
1250 Eye St.
Suite 200
Washington, DC 20005 (USA)
```

Lynn Barra può essere raggiunta anche al numero di telefono +1.202.626.5738 per richiedere una copia, ma probabilmente il contatto via lettera è più gradito.

Accesso attraverso mezzi Elettronici alla SQL3 Working Draft

La versione più recente (al momento della redazione di questo documento) della bozza di lavoro (e tutte le sue Parti) di SQL3 (ANSI e ISO), è disponibile presso

```
gatekeeper.dec.com
```

```
nella directory
```

```
/pub/standards/sql/
```

In questa directory ci sono numerosi file. Ci sono file in formato PostScript. ed in formato "testo puro" (non gradevolmente formattato, ma leggibile a video senza software particolare).

In generale, potete trovare file con nomi come:

```
sql-bindings-mar94.ps
sql-bindings-mar94.txt
sql-cli-mar94.ps
sql-cli-mar94.txt
sql-foundation-mar94.ps
sql-foundation-mar94.txt
sql-framework-mar94.ps
sql-framework-mar94.txt
sql-psm-mar94.ps
sql-psm-mar94.txt
```

Quando una nuova versione dei documenti verrà prodotta, la stringa "mar94" cambierà per indicare la nuova data di pubblicazione (ad es., "aug94" è la data di nuova pubblicazione dopo "mar94").

In aggiunta, per quei lettori che non potessero ottenere un listato della directory a mezzo FTP, abbiamo posizionato un file di nome:

```
ls
```

nella stessa directory. Questo file (che sorpresa!) contiene un listato del contenuto della directory.

Reperire File Usando ftp Diretto

Questo è un esempio di come usare FTP. In particolare, viene mostrato come connettersi a gatekeeper.dec.com, come spostarsi nella directory in cui il documento base è ospitato, e come trasferire il documento al vostro nodo. Notate che il vostro nodo deve avere accesso ad Internet per fare quanto sarà detto. Il nome di login è 'ftp', e la password è il vostro indirizzo email (a questa modalità ci si riferisce spesso come quella di "anonymous ftp" o "ftp anonimo"). Il comando 'type binary' è usato per assicurarci che nessun bit venga rimosso dal/i file ricevuto/i. Il comando 'get' trasferisce un file alla volta. Nello script che segue, i commenti sono racchiusi nei simboli < e > .

```
% ftp gatekeeper.dec.com
Connected to gatekeeper.dec.com.
220- *** /etc/motd.ftp ***
     Gatekeeper.DEC.COM is an unsupported service of DEC Corporate Research.
     <... ora si potrebbe dover attendere...>
220 gatekeeper.dec.com FTP server (Version 5.83 Sat ... 1992) ready.
Name (gatekeeper.dec.com:<il vostro nome-login>): ftp  <va bene anche l'ftp anonimo>
331 Guest login ok, send ident as password.
Password: <inserite qui la vostra email >
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.  <o qualsiasi altro>
Using binary mode to transfer files.
ftp> cd pub/standards/sql
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 9529
-r--r--r--  1 root     system     357782 Feb 25 10:18 x3h2-93-081.ps
-r--r--r--  1 root     system     158782 Feb 25 10:19 x3h2-93-081.txt
-r--r--r--  1 root     system     195202 Feb 25 10:20 x3h2-93-082.ps
-r--r--r--  1 root     system      90900 Feb 25 10:20 x3h2-93-082.txt
-r--r--r--  1 root     system    5856284 Feb 25 09:55 x3h2-93-091.ps
-r--r--r--  1 root     system    3043687 Feb 25 09:57 x3h2-93-091.txt
226 Transfer complete.
ftp> type binary
200 Type set to I.
ftp> get x3h2-93-082.txt
200 PORT command successful.
150 Opening BINARY mode data connection for x3h2-93-082.txt (90900 bytes).
226 Transfer complete.
90900 bytes received in 0.53 seconds (166.11 Kbytes/s)
ftp> quit
% <il file è ora nella vostra directory col nome x3h2-93-082.txt>
```

Scaricare File Senza Supporto di ftp Diretto

Digital Equipment Corporation, al pari di molte altre compagnie, fornisce il servizio di ftp via email. La risposta può giungere dopo diversi giorni, ma il servizio fornito è equivalente al ftp, ed è per coloro che non hanno accesso diretto a ftp. L'indirizzo del server è:

ftpmail@decwrl.dec.com

Il seguente script permetterà di ottenere il file, in formato PostScript, relativo all'ultima versione del documento su SQL3:

```
reply joe.programmatore@azienda-immaginaria.com
connect gatekeeper.dec.com anonymous
binary
compress
uuencode
chdir /pub/standards/sql
get x3h2-93-091.ps
quit
```

La prima riga dello script istruisce il server ad inviarci i file richiesti; dovreste sostituire "joe.programmatore@azienda-immaginaria.com" col vostro indirizzo Internet. Il file di quest'esempio, x3h2-93-091.ps, viene inviato in formato compresso (con "compress"), codificato (con "uuencode"), e frantumato in 34 messaggi email separati. Se il vostro ambiente non fornisce strumenti per ricostruire file di questo tipo, potreste farvi inviare il file in forma di testo puro, con il seguente script:

```
reply joe.programmatore@azienda-immaginaria.com
connect gatekeeper.dec.com anonymous
chdir /pub/standards/sql
get x3h2-93-091.ps
quit
```

Ma fate attenzione: il file .ps arriverà a voi suddiviso in più di 70 parti!

Per recuperare un file particolare qualsiasi, che non sia x3h2-93-091.ps, sostituite semplicemente "x3h2-93-091.ps" con il nome del file desiderato. Per ottenere una lista di tutti i file disponibili, sostituite " get x3h2-93-091.ps" con "dir".

29.10 Parte 1 - Stato Attuale di SQL ISO/ANSI

Questo capitolo è una fonte di informazioni relativamente al processo di standardizzazione di SQL, e del suo stato attuale.

Stato attuale:

Progetti di sviluppo sono attualmente in corso per migliorare SQL, rendendolo un linguaggio completo dal punto di vista computazionale, e per la definizione e la gestione di oggetti persistenti e complessi. Ciò comporta: la generalizzazione e la specializzazione delle gerarchie, l'eredità multipla, i tipi di dati definiti dall'utente, i trigger e le assertion, il supporto per i "knowledge based system", espressioni con interrogazioni ricorsive, e strumenti addizionali per l'amministrazione dei dati. Vanno considerati anche le specifiche dei tipi di dati astratti (ADT), gli identificatori di oggetti, i metodi, l'ereditarietà, il polimorfismo, l'incapsulamento, e tutte le altre caratteristiche normalmente associate alla gestione dei dati ad oggetti.

Nell'autunno 1996, diverse parti di SQL3 sono state poste in votazione dal CD ISO. Queste porzioni erano: SQL/Framework, SQL/Foundation, e SQL/Bindings. Queste votazioni fallirono (come ci si aspettava) con circa 900 commenti di critica. Alla fine di gennaio, si svolse un incontro di elaborazione ISO DBL, che produsse un vasto numero di soluzioni ai problemi che erano stati acclusi come commenti ai voti, o sottoposti come testi separati. Poiché l'incontro di elaborazione DBL non fu in grado di trattare tutti gli argomenti discussi nei commenti, l'incontro stesso venne esteso. Il completamento dell'incontro di elaborazione, è fissato per la fine di Luglio del 1997, a Londra.

Esaminando l'incontro di elaborazione di Luglio, ci si può aspettare che una votazione finale del CD sarà richiesta per le parti di SQL nominate precedentemente. Il processo del CD Finale impiegherà circa 6 mesi, e richiederà un incontro elaborativo DBL; dopodiché ci sarà una votazione DIS ed una rapidissima votazione IS.

Le procedure ISO sono cambiate rispetto a quelle che hanno determinato SQL/92; per questo i comitati SQL sono ancora impegnati nello studio dei dettagli della procedura stessa.

Se tutto andrà bene, queste parti di SQL3 diverranno ufficialmente standard ISO/IEC alla fine del 1998, ma i tempi sono molto stretti.

Nel 1993, i comitati di sviluppo ANSI e ISO decisero di scindere lo sviluppo del futuro SQL in uno standard, composto di molte parti. Esse sono:

- Parte 1: Framework - Una descrizione non-tecnica della struttura del documento.

- Parte 2: Foundation - Le specifiche del nucleo, che includono tutte le nuove funzionalità ADT.
- Parte 3: SQL/CLI - Interfaccia al Livello di Chiamata
- Parte 4: SQL/PSM - Le specifiche per le stored procedure, che includono la completezza computazionale.
- Parte 5: SQL/Bindings - SQL Dinamico e binding di SQL Embedded, derivati da SQL-92.
- Parte 6: SQL/XA - Una specifica SQL della diffusa Interfaccia XA sviluppata da X/Open
- Parte 7:SQL/Temporal - Aggiunge proprietà correlate con il tempo a SQL standard.

Negli USA, la totalità di SQL3 viene trattata sia come progetto ANSI Domestico ("D"), sia come progetto ISO. Allo stato attuale, il completamento di SQL3 è atteso per il 1999.

Le Parti SQL/CLI e SQL/PSM sono in corso di elaborazione, come aggiuntive a SQL-92. Negli USA, queste caratteristiche vengono trattate solo come progetti Internazionali ("I"). SQL/CLI è stata completata nel 1995. SQL/PSM dovrebbe essere completata intorno alla fine del 1996.

Oltre al lavoro su SQL3, numerosi altri progetti sono in corsa:

- SQL/MM - Un crescente sforzo per definire pacchetti multi-media usando le funzionalità ADT di SQL3.
- Accesso Remoto ai Dati (RDA)

Comitato degli Standard e Procedimento

In tutto il mondo, i comitati per gli standard SQL sono sorprendentemente numerosi. Esiste un gruppo per gli standard internazionali SQL, ed è parte di ISO. Moltissimi paesi hanno comitati che si occupano specificatamente di SQL. Questi paesi (solitamente) inviano le loro rappresentative ai convegni ISO/IEC JTC1/SC 21/WG3 DBL. I paesi che partecipano attivamente alle procedure, per definire gli standard SQL ISO, sono:

- Australia
- Brasile
- Canada
- Francia
- Germania
- Giappone
- Corea
- Paesi Bassi
- Regno Unito
- Stati Uniti d'America

Convalida NIST

Le implementazioni di SQL sono convalidate (negli USA) dal National Institute of Standards and Training (NIST). Attualmente il NIST si serve di una suite di prova per nuovi prodotti SQL-92. I dettagli esatti dei requisiti, per la convalida del NIST, vengono definiti nel Federal Information Processing Standard (FIPS). I requisiti attuali per SQL sono definiti nel FIPS 127-2. Le versioni nei formati Postscript e testo di questo documento, possono essere reperite presso NIST. L'attuale Lista di Prodotti SQL Convalidati può essere ottenuta dal NIST.

Pubblicazioni di SQL Standard e articoli

Esistono due versioni dello standard SQL. Entrambe sono disponibili presso ANSI:

- ISO/IEC 9075:1992, "Information Technology — Database Languages — SQL"
- ANSI X3.135-1992, "Database Language SQL"

Le due versioni dello standard SQL sono identiche, tranne che per le intestazioni ed i riferimenti ad altri standard. Entrambe sono reperibili presso:

```
American National Standards Institute
1430 Broadway
New York, NY 10018
USA
Phone (sales): +1.212.642.4900
```

Oltre allo standard SQL-92, ora c'è un Technical Corrigendum (correzione dei bug):

```
* Technical Corrigendum 1:1994 to ISO/IEC 9075:1992
```

TC 1 dovrebbe essere reso disponibile da ANSI. C'è solo una versione ISO di TC 1 – Si applica sia alla versione ANSI che a quella ISO di SQL-92.

Oltre ai testi base, svariati libri sono stati scritti sullo standard SQL 1992. Questi libri forniscono una descrizione dello standard assai più comprensibile dei testi ufficiali veri e propri.

Standard Correlati

Numerosi altri standard sono oggetto di interesse per la comunità SQL. Questa sezione contiene alcuni puntatori alle informazioni su queste realizzazioni. Questi puntatori aumenteranno di numero, man mano che le informazioni aggiuntive diventeranno disponibili sul web.

- SQL Environments (FIPS 193)
- Next Generation Repository Systems (X3H4) - a News Release calling for partecipation in "Developing Standards for the Next Generation Repository Systems."

29.11 Parte 2 - Foundation - Fondamenti di SQL ISO/ANSI

Una quota significativa, nella realizzazione di SQL3, è quella del Documento dei Fondamenti di SQL:

- Proprietà Basilari SQL/PSM (ereditate da SQL/PSM-92)
- Nuovi tipi di dati
- Trigger
- Sottotabelle
- Tipi di Dati Astratti (Abstract Data Types:ADT)
- Funzionalità Orientate agli Oggetti

Ci sono diversi prerequisiti per le funzionalità orientate agli oggetti:

- Possibilità di definizione di operazioni complesse
- Operazioni complesse di archiviazione dei dati nel database
- Chiamate di procedure esterne Qualche operazione può non essere presente in SQL, o può richiedere interazioni con l'ambiente esterno

Queste funzionalità sono definite come parti di SQL/PSM

Gran parte del lavoro, viene attualmente svolta per affinare il modello a oggetti di SQL-3, e per allinearlo con il modello ad oggetti proposto da ODMG. Questo lavoro è descritto nel documento X3H2 e ISO DBL: "Accomodating SQL3 and ODMG". Esiste anche un recente aggiornamento sull'operazione di Merge (fusione) SQL3/OQL.

Tempi di lavoro di SQL3

Lo sviluppo di SQL3 è sulla buona strada, ma gli standard finali saranno pronti fra anni.

- Votazione Internazionale per far evolvere i Fondamenti SQL3 dalla bozza di lavoro alla Bozza del Comitato (Committee Draft - CD), autunno 1996.
- In seguito alla votazione si attendono numerosi commenti di critica
- Verosimilmente sarà necessaria una seconda votazione
- La votazione della bozza standard internazionale (Draft International Standard) avrà luogo probabilmente a metà del 1998
- Lo standard Internazionale dovrebbe essere completato a metà del 1999.

La versione ANSI dello standard avrà scadenze simili.

29.12 Parte 3 - Interfaccia ISO/ANSI SQL al Livello di Chiamata (Call Level Interface)

SQL/CLI è un'interfaccia a livello di chiamata ai database SQL. È progettata per agevolare l'accesso al database da parte di applicazioni "shrink-wrapped". La CLI è stata originariamente creata da un sottocomitato di SQL Access Group (SAG). Le specifiche SAG/CLI sono state pubblicate come le specifiche di Microsoft Open DataBase Connectivity (ODBC) del 1992. Nel 1993, il SAG sottomise la CLI ai comitati SQL ANSI e ISO SQL. (Ora SQL Access Group si è fuso con il consorzio X/Open.)

SQL/CLI fornisce uno standard internazionale per:

- CLI indipendenti dalla implementazione per accedere a database SQL
- Strumenti client-server che possono facilmente accedere ai database, attraverso Link dinamici alle Librerie
- Supporto ed incoraggiamento di un vasto numero di strumenti Client-server

Tempi di lavoro di SQL/CLI

Per essere un procedimento di standardizzazione, quello di SQL/CLI è stato realizzato in tempi rapidissimi.

- SQL/CLI è un'appendice allo standard SQL 1992 (SQL-92)
- È stato completato come standard ISO nel 1995
- ISO/IEC 9075-3:1995 Information technology – Database languages – SQL – Part 3: Call-Level Interface (SQL/CLI)
- Attualmente il lavoro su SQL/CLI è focalizzato sul supporto per le funzionalità di SQL3

29.13 Parte 4 - Persistent Stored Modules (Moduli Memorizzati Persistenti) SQL ISO/ANSI

SQL/PSM estende SQL aggiungendo:

- Estensioni di linguaggio Procedurale
- Istruzioni multiple e Stored Procedure
- Funzioni esterne e chiamate di procedura

Oltre ad essere un valido strumento di sviluppo di applicazioni, SQL/PSM fornisce il supporto fondamentale per le funzionalità orientate agli oggetti in SQL3.

Istruzioni multiple e Stored Procedure

Le istruzioni multiple e le Stored Procedure, offrono una varietà di vantaggi in ambiente client/server:

- Prestazioni - Poiché una stored procedure può eseguire istruzioni SQL multiple, l'interazione via rete col client è ridotta.
- Sicurezza - Ad un utente può essere conferito il diritto di richiamare una stored procedure che aggiorni una tabella od un insieme di tabelle, ma gli si può proibire l'aggiornamento diretto delle tabelle
- Codice Condiviso - Il codice in una stored procedure non deve essere riscritto e ritestato, per ogni strumento client che accede al database.
- Controllo - Fornisce un singolo punto di definizione e controllo per la logica dell'applicazione.

Estensioni di Linguaggio Procedurale

Il linguaggio procedurale aggiunge a SQL la potenza di un tradizionale linguaggio di programmazione, attraverso istruzioni per il controllo del flusso, ed una varietà di altri costrutti di programmazione.

Istruzioni per il controllo di Flusso

- If-then-else
- Costrutti per esecuzione di cicli
- Gestione delle Eccezioni
- Istruzione Case
- Blocchi Begin-End

Le estensioni di linguaggio procedurale comprendono anche altri costrutti:

- Dichiarazioni di variabile

- Istruzioni SET per assegnazione di valore
- GET DIAGNOSTICS per ottenere informazioni sul processo e sullo stato

Inoltre, tutte le tradizionali istruzioni SQL possono essere incluse nelle procedure con istruzioni multiple.

Procedure Esterne e Chiamate di Funzione

Esiste una funzionalità spesso citata tra quelle desiderate per molti database, ed implementate in qualcuno. Essa è la capacità di poter aggiungere, alle funzionalità interne, altre funzionalità, mediante chiamate a procedure scritte dall'utente, esterne al software del database stesso.

- È possibile aggiungere, a siti o ad applicazioni, specifiche funzioni al database con cui interagiscono
- Possono essere usate con qualsiasi applicazione che acceda al database

Il beneficio di queste capacità, è che permettono ad un database (e quindi anche alle applicazioni database) di poter usare un grande insieme di procedure e funzioni, troppo numerose per essere definite da un comitato per gli standard.

Tempi di sviluppo di SQL/PSM

SQL/PSM sta procedendo rapidamente:

- SQL/PSM è un'appendice a SQL-92
- Le votazioni internazionali, per far avanzare SQL/PSM a partire dalla prima bozza internazionale (Draft International Standard) verso uno Standard Internazionale, sono terminate nel Gennaio 1996.
- L'incontro elaborativo di Maggio 1996 non ha risolto tutte le questioni aperte
- La continuazione dell'incontro elaborativo PSM, è stabilita per il periodo dal 30 Settembre al 4 Ottobre del 1996
- I tempi sono ristretti, ma c'è la possibilità che PSM possa essere pubblicato entro il 1996.
- Il titolo ufficiale sarà: ISO/IEC DIS 9075-4:199? Information technology – Database languages – SQL – Part 4: SQL Persistent Stored Modules (SQL/PSM)
- Il lavoro di supporto SQL/PSM, alle funzionalità di SQL3, viaggia ben spedito.

29.14 Parte 5 - ISO/ANSI SQL/Binding

Per facilità di consultazione i binding con i linguaggi di programmazione, sono stati stralciati in un documento a parte. L'attuale versione è semplicemente un estratto dei binding incorporati e dinamici, tratti da SQL-92.

Svariate questioni, riguardanti i binding con i linguaggi di programmazione, rimangono irrisolte.

Per linguaggi di programmazione tradizionale, esiste la mappatura dei tipi di dati per SQL-92. Comunque, deve essere definita una mappatura tra gli oggetti SQL e le variabili del linguaggio di programmazione.

Nel caso dei linguaggi orientati agli oggetti, deve essere definita una mappatura per i tipi di dati SQL attuali, e tra il modello ad oggetti SQL ed il modello ad oggetti del linguaggio orientato agli oggetti.

Il modello ad oggetti deve essere reso stabile prima di essere utilizzato.

I binding coi linguaggi saranno completati come parte di SQL3.

29.15 Parte 6 - ISO/ANSI SQL XA Interface Specialization (SQL/XA)

Queste specifiche vorrebbero standardizzare un'interfaccia ai programmi applicativi (API), tra un Gestore di Transazione globale ed un Gestore di Risorse SQL. Sono volte a standardizzare le chiamate di funzione, basate sulle semantiche di ISO/IEC 10026, "Distributed Transaction Processing", che un Gestore di Risorse SQL dovrebbe poter supportare per commit in due-fasi. Il documento base è derivato da una pubblicazione X/Open, con il permesso di X/Open, che specifica esplicitamente i parametri e le semantiche di input ed output, in termini dei tipi di dati SQL, per le seguenti funzioni: xa_close, xa_commit, xa_complete, xa_end, xa_forget, xa_open, xa_prepare, xa_recover, xa_rollback, e xa_start.

Attualmente ISO sta tentando di effettuare il fast-track delle specifiche X/Open XA. Il processo di fast-track porta ad adottare specifiche industriali attualmente in uso senza apporre cambiamenti. Le votazioni relative al fast-track di XA al ISO SC21, livello JTC 1, sono iniziate il 27 aprile, 1995, e terminano il 27 ottobre 1995. Se le specifiche XA verranno approvate dal 75% dei votanti, e dai 2/3 dei membri di JTC 1, diverranno uno Standard Internazionale. Se la votazione sul fast-track sarà approvata, SQL/XA potrebbe diventare uno standard nel 1996.

29.16 Parte 7 - ISO/ANSI SQL Temporal

SQL Temporal riguarda i dati correlati con il tempo. Il concetto alla sua base, è l'utilità di effettuare interrogazioni dei dati per conoscere come erano in un momento particolare. SQL Temporal è raccolto in un testo di Rick Snodgrass, datato dicembre 1994, in cui sono descritti i concetti.

X3 Dichiara l'Approvazione di un Nuovo Progetto, ISO/IEC 9075 Parte 7: SQL/Temporal è un comunicato-stampa relativo a SQL/Temporal.

```
----------------------------------------------------------------------------
                              Temporal SQL
                              ************
Rick Snodgrass (presidente del comitato TSQL2)
31-Dic-1994
```

Diverse persone hanno contestato la necessità del supporto addizionale per il tempo in SQL3 (come proposto da DBL RIO-75, che ha richiesto una nuova parte di SQL per supportare i database temporali). Si sostiene che i tipi di dati astratti (ADT) siano sufficienti per il supporto temporale. In quest'articolo informativo sostengo, usando esempi concreti, che usare colonne definite con tipi di dati astratti, per interrogazioni temporali, è una soluzione inadeguata. In particolare, molte ordinarie interrogazioni temporali sono difficili da simulare in SQL, oppure richiedono l'incorporazione di SQL in un linguaggio procedurale. In TSQL2, un'estensione temporale di SQL-92, sono state proposte delle alternative.

29.16.1 INTRODUZIONE

Il supporto del tempo di validità surclassa quello di un ADT temporale. Secondo quest'ultimo, una colonna è specificata come parte di un dominio temporale, come DATE o INTERVAL (più avanti verranno forniti brevi esempi). Con la gestione del tempo di validità, le righe di una tabella variano nel tempo, al variare della realtà. Il timestamp associato ad una riga di una tabella con tempo di validità, è interpretato, dal linguaggio di interrogazione, come quel momento in cui la combinazione dei valori delle colonne nella riga era valida. Questo timestamp implicito permette di esprimere le interrogazioni in modo succinto ed intuitivo.

29.16.2 UN CASO DI STUDIO - ARCHIVIARE LE INFORMAZIONI CORRENTI

L'Ufficio Personale (UP) dell'Università dell'Arizona ha archiviato alcune informazioni in un database, tra cui il nome di ogni impiegato, il suo attuale salario, ed il suo attuale titolo. Ciò può essere rappresentato per mezzo di una semplice tabella.

```
Impiegato(Nome, Salario, Titolo)
```

Data questa tabella, trovare il salario di un impiegato è facile.

```
SELECT Salario
FROM Impiegato
WHERE Nome = 'Bob'
```

Adesso l'UP desidera registrare la data di nascita. Per fare ciò, viene aggiunta una colonna alla tabella, producendo il seguente schema.

```
Impiegato(Nome, Salario, Titolo, DatadiNascita DATE)
```

Trovare la data di nascita dell'impiegato è analogo al determinarne il salario.

```
SELECT DatadiNascita
FROM Impiegato
WHERE Nome = 'Bob'
```

29.16.3 UN CASO DI STUDIO - ARCHIVIARE INFORMAZIONI STORICHE

L'UP desidera gestire, via computer, l'archivio storico dei dipendenti dell'ufficio. Per farlo vengono aggiunte due colonne: la prima indica quando l'informazione della riga diventa valida, mentre l'altra indica quando l'informazione non è più valida.

Impiegato (Nome, Salario, Titolo, DatadiNascita, Inizio DATE, Fine DATE)

Secondo il modello dei dati, queste nuove colonne sono identiche a DatadiNascita. Tuttavia, la loro presenza ha conseguenze a largo raggio.

29.16.4 UN CASO DI STUDIO - PROIEZIONE

Per trovare il salario attuale di un impiegato, le cose si fanno più difficili.

```
SELECT Salario
FROM Impiegato
WHERE Nome = 'Bob' AND Inizio <= CURRENT_DATE AND CURRENT_DATE <= Fine
```

Questa query è più complicata della precedente. Le indiziate sono, ovviamente, le due nuove colonne. L'UP vuole fornire, ad ogni impiegato, la cronistoria del proprio salario. In particolare, devono essere determinati gli intervalli massimali per ogni salario. Sfortunatamente, questo non è possibile in SQL. Un impiegato potrebbe avere mutato il proprio titolo, pur mantenendo fisso il valore del salario.

```
Nome    Salario Titolo             DatadiNascita Inizio          Fine
----    ------- ------             ------------- ------          ----
Bob     60000   Assistente Rettore 1945-04-09    1993-01-01      1993-05-30
Bob     70000   Assistente Rettore 1945-04-09    1993-06-01      1993-09-30
```

```
Bob     70000   Rettore           1945-04-09    1993-10-01      1994-01-31
Bob     70000   Professore        1945-04-09    1994-02-01      1994-12-31
```

```
                                Figura 1
```

Notate che ci sono tre righe in cui il salario di Bob è rimasto costante a 70'000 $. Ne segue che il risultato dovrebbe essere di due righe per Bob.

```
Nome    Salario Inizio          Fine
----    ------- ------          ----
Bob     60000   1993-01-01      1993-05-30
Bob     70000   1993-06-01      1994-12-31
```

Un'alternativa è quella di fornire all'utente un estratto delle informazioni Salario e Titolo, così che egli possa determinare da sé quando il suo salario sia cambiato. Questa alternativa non è molto attraente, e neanche pratica. Una seconda alternativa è quella di usare SQL fin quando possibile.

```
CREATE TABLE Temp(Salario, Inizio, Fine)
AS      SELECT Salario, Inizio, Fine
        FROM Impiegato;
```

ripetete

```
        UPDATE Temp T1
        SET (T1.Fine) = (SELECT MAX(T2.Fine)
                         FROM Temp AS T2
                         WHERE T1.Salario = T2.Salario AND T1.Inizio < T2.Inizio
                                AND T1.Fine >= T2.Inizio AND T1.Fine < T2.Fine)
        WHERE EXISTS (SELECT *
                      FROM Temp AS T2
                      WHERE T1.Salario = T2.Salario AND T1.Inizio < T2.Inizio
                                AND T1.Fine >= T2.Inizio AND T1.Fine < T2.Fine)
```

finché più nessuna riga viene aggiornata;

```
DELETE FROM Temp T1

WHERE EXISTS (SELECT *
              FROM Temp AS T2
              WHERE T1.Salario = T2.Salario
                        AND ((T1.Inizio > T2.Inizio AND T1.Fine <= T2.Fine)
                        OR (T1.Inizio >= T2.Inizio AND T1.Fine < T2.Fine))
```

Il ciclo trova quegli intervalli che si sovrappongono o sono adiacenti, e che quindi dovrebbero essere fusi. Nella peggiore delle ipotesi, il ciclo viene eseguito log N volte, dove N è il numero di righe comprese in una serie di sovrapposizioni o di adiacenze. Il lettore può simulare l'interrogazione sulla tabella d'esempio, per convincersi della sua correttezza.

Una terza alternativa è quella di usare SQL solo per aprire un cursore sulla tabella. Viene mantenuta una lista collegata di periodi, ognuno di essi associato ad un salario. Questa lista collegata dovrebbe essere inizializzata come lista vuota.

```
DECLARE emp_cursor CURSOR FOR
        SELECT Salario, Titolo, Inizio, Fine
        FROM Impiegato;
OPEN emp_cursor;
ciclo:
FETCH emp_cursor INTO :salario, :inizio, :fine;
        se nessun-dato restituito goto finito;
        trova posizione in lista collegata per inserire quest'informazione;
        goto ciclo;
finito:
CLOSE emp_cursor;
```

itera attraverso la lista collegata, stampando date e salari

In questo caso, la lista collegata potrebbe non essere necessaria, se il cursore è realizzato con ORDER BY Inizio.

In ogni caso l'interrogazione, anche la più semplice, è piuttosto difficile da esprimere usando le funzionalità presenti in SQL-92. L'interrogazione è banale in TSQL2.

```
SELECT Salario
FROM Impiegato
```

29.16.5 UN CASO DI STUDIO - JOIN

Un approccio più drastico è quello di evitare il problema dell'estrazione della cronistoria del salario, riorganizzando lo schema per separare le informazioni di salario, titolo, e data di nascita (in seguito ignoreremo la data di nascita, per semplicità).

```
Impiegato1 (Nome, Salario, DATA Inizio, DATA Fine)
Impiegato2 (Nome, Titolo, DATA Inizio, DATA Fine)
```

La tabella dell'Impiegato1 è la seguente.

```
Nome    Salario Inizio          Fine
----    ------- ------          ----
Bob     60000   1993-01-01      1993-05-30
Bob     70000   1993-06-01      1993-12-31
```

Segue la tabella d'esempio Impiegato2.

```
Nome    Titolo                  Inizio          Fine
----    ------                  ------          ----
Bob     Assistente Rettore      1993-01-01      1993-09-30
Bob     Rettore                 1993-10-01      1994-01-31
Bob     Professore              1994-02-01      1994-12-31
```

Ora, con questa variazione, è facile ottenere l'informazione del salario di un impiegato.

```
SELECT Salario, Inizio, Fine
FROM Impiegato1
WHERE Nome = 'Bob'
```

Ma che fare se l'Ufficio Personale volesse una tabella che contenga intervalli di salario e titoli ? (cioè, supponete che l'UP voglia una tabella simile a quella in Figura 1) Una possibilità è quella di stampare due tabelle, lasciando all'utente l'onere di immaginare le giuste combinazioni. Una seconda alternativa è quella di usare esclusivamente SQL. Sfortunatamente, quest'interrogazione deve riguardare un caso d'analisi relativamente a come ogni riga di Impiegato1 si sovrappone ad ogni riga di Impiegato2; sono possibili quattro casi.

```
SELECT Impiegato1.Nome, Salario, Dipartimento, Impiegato1.Inizio, Impiegato1.Fine
FROM Impiegato1, Impiegato2
WHERE Impiegato1.Nome = Impiegato2.Nome
     AND Impiegato2.Inizio <= Impiegato1.Inizio AND Impiegato1.Fine < Impiegato2.Fine
UNION
SELECT Impiegato1.Nome, Salario, Dipartimento, Impiegato1.Inizio, Impiegato2.Fine
FROM Impiegato1, Impiegato2
WHERE Impiegato1.Nome = Impiegato2.Nome
     AND Impiegato1.Inizio >= Impiegato2.Inizio AND Impiegato2.Fine < Impiegato1.Fine
        AND Impiegato1.Inizio < Impiegato2.Fine
UNION
SELECT Impiegato1.Name, Salario, Dipartimento, Impiegato2.Inizio, Impiegato1.Fine
FROM Impiegato1, Impiegato2
WHERE Impiegato1.Nome = Impiegato2.Nome
     AND Impiegato2.Inizio > Impiegato1.Inizio AND Impiegato1.Fine < Impiegato2.Fine
        AND Impiegato2.Inizio < Impiegato1.Fine
UNION
SELECT Impiegato1.Nome, Salario, Dipartimento, Impiegato2.Inizio, Impiegato2.Fine
FROM Impiegato1, Impiegato2
WHERE Impiegato1.Nome = Impiegato2.Nome
     AND Impiegato2.Inizio > Impiegato1.Inizio AND Impiegato2.Fine  < Impiegato1.Fine
```

Ottenere tutti i casi giusti è un compito arduo. In TSQL2, effettuando una join temporale, si ottiene proprio quanto ci si attende.

```
SELECT Impiegato1.Nome, Salario, Dipartimento
FROM Impiegato1, Impiegato2
WHERE Impiegato1.Nome = Impiegato2.Nome
```

29.16.6 UN CASO DI STUDIO - FUNZIONI AGGREGATE

La prossima richiesta dell'Ufficio Personale è: qual è il salario massimo? Prima di aggiungere il tempo, ciò era facile.

```
SELECT MAX(Salario)
FROM Impiegato
```

Ora, dal momento che è stata archiviata la cronistoria del salario, vorremmo una cronistoria del massimo salario. Il problema, naturalmente, è che SQL non fornisce funzioni aggregate temporali. Il modo facile per farlo è quello di stampare l'informazione, e scorrerla manualmente alla ricerca dei massimi. Un'alternativa più scaltra è quella di convertire l'interrogazione istantanea aggregata in una non-aggregata, e poi convertire quest'ultima in un'interrogazione temporale. L'interrogazione non-aggregata trova quei salari per i quali non esiste un salario maggiore degli altri.

```
SELECT Salario
FROM Impiegato AS I1
WHERE NOT EXISTS (SELECT *
                  FROM Impiegato AS I2
                  WHERE I2.Salario > I1.Salario)
```

Convertire questa query in un'interrogazione temporale non è affatto banale. Quella che segue è una delle strade possibili.

```
CREATE TABLE Temp (Salario, Inizio, Fine )
AS      SELECT Salario, Inizio, Fine
        FROM Impiegato;
INSERT INTO Temp
        SELECT T.Salario, T.Inizio, I.Inizio
        FROM Temp AS T, Impiegato AS I
        WHERE I.Inizio >= T.Inizio AND I.Inizio < T.Fine  AND I.Salario > T.Salario;
INSERT INTO Temp
        SELECT T.Salario, T.Fine , I.Fine
        FROM Temp AS T, Impiegato AS I
        WHERE I.Fine > T.Inizio AND I.Fine <= T.Fine AND I.Salario > T.Salario;
DELETE FROM Temp T
WHERE EXISTS (SELECT *
              FROM Impiegato AS I
              WHERE ((T.Inizio => I.Inizio AND T.Inizio < I.Fine )
                        OR (I.Inizio >= T.Inizio AND I.Inizio < T.Fine ))
                    AND I.Salario > T.Salario;
```

Seguendo questa strada creiamo una tabella ausiliaria. Aggiungiamo a questa tabella il minor periodo di una sottrazione di periodi, ed il maggior periodo di una sottrazione di periodi. Poi cancelliamo tutti i periodi che si sovrappongono a qualche riga definita dalla subquery, e in seguito utilizziamo la precedente interrogazione con l'espressione NOT EXISTS. Infine generiamo il periodo massimo a partire dalla tabella ausiliaria, e lo facciamo nello stesso modo in cui è stato calcolato in precedenza. Come si può immaginare, questo codice SQL è estremamente inefficiente da eseguire, viste le complesse interrogazioni nidificate con predicati irregolari.

Una terza possibilità è quella di impiegare il meno possibile SQL, e calcolare invece il desiderato valore massimo della cronistoria, servendosi dei cursori in un linguaggio ospite.

La query in TSQL2 è ancora una volta lineare ed intuitiva.

```
        SELECT MAX(Salario)
        FROM Impiegato
```

29.16.7 RIASSUNTO

I dati variabili nel tempo sono gestiti dalla maggior parte delle applicazioni database. Il supporto del tempo di validità è assente in SQL. Molte ordinarie interrogazioni temporali sono difficilmente simulabili in SQL, oppure richiedono l'incorporazione di SQL in un linguaggio procedurale, a causa della mancanza di supporto alle tabelle con tempo di validità nel modello di dati di SQL, e nei suoi costrutti interrogativi.

D'altra parte, abbiamo dimostrato come l'aggiunta del supporto del tempo di validità richieda poche variazioni all'implementazione del DBMS, e come possa semplificare in modo drammatico alcune interrogazioni e consentirne altre; abbiamo visto, inoltre, come ciò possa consentire ottimizzazioni delle strutture di archiviazione, dei metodi di indicizzazione, e delle strategie di ottimizzazione che possono produrre significativi miglioramenti delle prestazioni.

Con una nuova parte di SQL3, che gestisce la variazione nel tempo dell'informazione, possiamo iniziare a realizzare applicazioni mirate, consentendo a SQL3 di gestire i dati temporali in modo più accorto.

```
---------------------------------------------------------------------------
           Accredited Standards Committee* X3, Information Technology
```

```
NEWS RELEASE

Doc. No.:       PR/96-0002

Reply to:       Barbara Bennett presso bbennett@itic.nw.dc.us

              X3 Annuncia l'approvazione di un Nuovo Progetto, ISO/IEC

                        9075 Parte 7:  SQL/Temporal

Washington D.C., Gennaio 1996
----------------------------------------------------------------------------
```

– Accredited Standards Committee X3, Information Technology, annuncia l'approvazione di un nuovo progetto di Supporto a SQL/Temporal, ISO/IEC 9075 Parte 7, con il lavoro realizzato dal Technical Committee X3H2, Database. L'intenzione, di questa proposta di standard, è quella di stabilire una nuova Parte dello standard SQL3 in costruzione, e cioè la Parte 7, SQL Temporale, che vuole estendere il linguaggio SQL per fornire supporto all'archiviazione, estrazione, e manipolazione dei dati temporali in un ambiente di database SQL. Il prossimo meeting X3H2 è fissato per 11-14 Marzo 1996 in Kansas.

Le richieste di informazioni, riguardo a questo progetto, dovrebbero essere inviate al

```
Presidente di X3H2,
Dr. Donald R. Deutsch,
Sybase, Inc., Suite 800,
6550 Rock Spring
Drive, Bethesda, MD  20817.
Email: deutsch@sybase.com.
```

Una primo giro di consultazioni, su possibili licenze e su altre questioni pertinenti (copyright, marchi), è in corso. Si prega di fornire le informazioni del caso al

```
Segretariato X3 presso
1250 Eye Street
NW, Suite 200,
Washington DC  20005.
Email: x3sec@itic.nw.dc.us
FAX:  (202)638-4922.
```

29.17 Parte 8 - ISO/ANSI SQL MULTIMEDIA (SQL/MM)

Un nuovo progetto di standardizzazione internazionale ISO/IEC, volto allo sviluppo di una libreria di classi SQL per applicazioni multimediali, è stato approvato agli inizi del 1993. Questa nuova attività di standardizzazione, denominata SQL Multimedia (SQL/MM), mira a stabilire l'insieme delle definizioni dei tipi di dati astratti (ADT) di SQL, usando gli strumenti per la specifica e l'invocazione degli ADT, i quali sono forniti dalle specifiche del costruendo SQL3. SQL/MM si propone di standardizzare librerie di classi ad uso di ambienti scientifici ed ingegneristici, e per elaborazione di documenti e di metodi per la gestione di oggetti multimediali come immagini, suoni, animazione, musica, e video. Verosimilmente SQL/MM fornirà, al linguaggio SQL, un collegamento agli oggetti multimediali definiti da altri organismi di standardizzazione JTC1 (ad es. SC18 per i documenti, SC24 per le immagini, e SC29 per fotografie e filmati).

Il Piano del Progetto di SQL/MM, indica che sarà uno standard composito, consistente di un certo numero di parti sviluppate indipendentemente. La Parte 1 sarà una struttura di base, che specifica come le altre parti devono essere realizzate. Ognuna delle altre parti sarà dedicata ad uno specifico pacchetto applicativo SQL. La seguente struttura delle Parti SQL/MM esiste dall'Agosto 1994:

- Parte 1: Framework (Struttura), Una descrizione non tecnica di come il documento è strutturato.
- Parte 2: Metodi Full Text e ADT per elaborazione di dati in forma di testo. Circa 45 pagine.
- Parte 3: Metodi Spaziali e ADT per gestione di dati spaziali. Sono circa 200 pagine, realizzate con la collaborazione attiva di esperti in elaborazione dei Dati Spaziali, provenienti da 3 organismi nazionali.
- Parte 4: Metodi di Impiego Generale e ADT per numeri complessi; Le funzionalità descritte comprendono funzioni trigonometriche ed esponenziali, vettori, insiemi, ecc. Attualmente consta di circa 90 pagine.

Esiste un certo numero di tentativi di standardizzazione nell'area delle Informazioni Spaziali e Geografiche:

- ANSI X3L1 - Sistemi di Informazione Geografica. Mark Ashworth, di Unisys, è il raccordo tra X3L1 e ANSI X3H2. È anche curatore delle parti 1, 3, e 4 della bozza di SQL/MM.
- ISO TC 211 - Informazione Geografica/Geomatica

30 Supporto Tecnico per PostgreSQL

Questo è l'ordine da seguire per la risoluzione di problemi:

- Alle vostre domande potete trovare risposta presso gli *online manuals* `<http://www.postgresql.org/users-lounge>`
- Inserite una parola chiave nel *search box* `<http://www.postgresql.org/search.cgi>`
- Pubblicate le vostre domande nella mailing list

Se dovete porre domande di natura tecnica, o incontrate qualsiasi problema, potete inviare un'e-mail a:

- *pgsql-questions@postgresql.org*
- Newsgroup `<comp.databases.postgresql.general>`
- Newsgroup `<comp.databases.postgresql.hackers>`
- Newsgroup `<comp.databases.postgresql.doc>`
- Newsgroup `<comp.databases.postgresql.bugs>`
- Newsgroup `<linux.postgres>`
- Altre Mailing list `<http://www.postgresql.org>`

Otterrete risposte via e-mail entro un giorno. La base di utenti dei prodotti internet è molto vasta, inoltre gli utenti assistono altri utenti, perciò internet sarà in grado di fornire facilmente assistenza tecnica a miliardi di utenti. L'assistenza via Email è molto più pratica di quella telefonica, poiché potete fare il "copia e incolla" dei messaggi di errore, dell'output del programma ecc.. e potete facilmente comunicare con mailing list/newsgroup.

30.1 Supporto Commerciale

L'organizzazione di PostgreSQL fornisce, dietro pagamento, assistenza tecnica alle aziende; le entrate saranno usate per il mantenimento di vari siti mirror (web e ftp) intorno al mondo. Inoltre i proventi saranno anche impiegati per la produzione di documentazione stampata, guide, manuali, che aiuteranno gli utenti. Il sito dell'organizzazione è `<http://www.postgresql.org>`

Un'altra azienda, chiamata 'Great Bridge Corporation', effettua sviluppo, vendita e assistenza di PostgreSQL. Il loro sito è `<http://www.greatbridge.com>`. È una società per azioni fondata della 'Landmark Communications corp' ed altre imprese di capitale, volta esclusivamente alla vendita di PostgreSQL ed alla fronitura di assistenza a grandi imprese ed aziende intorno al mondo.

Potete ottenere assistenza da aziende di consulenza professionale come RedHat, Anderson, WGS (Work Group Solutions). Contattateli per richiedere assistenza, poiché hanno una buonissima esperienza in "C", "C++" (PostgreSQL è scritto in "C") -

- Redhat Corp - Database consulting division `<http://www.redhat.com>`
- Work Group Solutions `<http://www.wgs.com>`
- Anderson Consulting `<http://www.ac.com>`

31 Aspetti imprenditoriali ed economici

I produttori di database commerciali pagano molti tributi come la tassa federale, quella di stato, di vendita, di impiego, di sicurezza sociale, di assistenza medica agli impiegati, oltre ai benefici per gli impiegati, e i costi di marketing e pubblicitari. Tutti questi costi non sono diretti allo sviluppo del database, e non incrementano la qualità o la tecnologia del database. Quanto acquistate un database commerciale, buona parte delle entrate è destinata a spese extra come le tasse, e le spese di bilancio per i costi R&S del database (Ricerca e Sviluppo, n.d.t.).

Le aziende dei database commerciali devono pagare anche per i loro edifici/beni immobiliari, e per l'acquisto di macchine Unix, per la loro installazione e manutenzione. Tutti questi costi vengono scaricati sui clienti.

PostgreSQL è superiore ai database commerciali perché non c'è alcuna tassa per un prodotto realizzato in internet. Un vastissimo gruppo di persone contribuisce allo sviluppo di PostgreSQL. Facciamo il caso ipotetico, ad esempio, che negli U.S.A. esista un milione di aziende, e che ogni azienda versi circa 10 $ (contribuendo così allo sviluppo di PostgreSQL); così facendo ogni compagnia ricaverebbe dieci milioni di dollari!! Questa è la **GRANDE MAGIA** dello sviluppo di software su internet.

Attualmente il codice sorgente di PostgreSQL è di circa 250'000 righe di codice in "C", "C++". Se il costo di ogni riga di codice "C" è di 2 $, il valore di PostgreSQL è di circa 500'000 $ (mezzo milione di dollari!).

Molte aziende hanno già sviluppato per proprio conto grandi quantità di codice "C", "C++". Se incorporassero questo codice nel sorgente di PostgreSQL, e collaborassero con altre aziende su internet, si otterrebbe un grande beneficio per tutti, e ciò permetterebbe di risparmiare tempo e fatica.

32 Lista di Altri Database

Segue un elenco di altri database SQL per Unix, Linux.

- Fate click e andate alle Applicazioni->database. `<http://www.caldera.com/tech-ref/linuxapps/linapps.html>`
- Fate click e andate alle Applicazioni->database. `<http://www.xnet.com/~blatura/linapps.shtml>`

- Risorse su Database <http://linas.org/linux/db.html>. Sono state scritte da Linas Vepstas: *linas@fc.net*
- Lista di Database Liberi/Gratuiti <http://cuiwww.unige.ch:80/~scg/FreeDB/FreeDB.list.html>
- Lista di RDBMS di Browne <http://www.hex.net/~cbbrowne/rdbms.html> scritta da Christopher B. Browne *cbbrowne@hex.net*
- Lista SAL di DBMS Relazionali <http://SAL.KachinaTech.COM/H/1/>
- Lista SAL di DBMS Orientati agli Oggetti <http://SAL.KachinaTech.COM/H/2/>
- Lista SAL di Utilità e Altri Database <http://SAL.KachinaTech.COM/H/3/>
- ACM SIGMOD Indice di Software Database Disponibile Pubblicamente <http://bunny.cs.uiuc.edu/sigmod/databaseSoftware/>

33 Suggerimenti per la Ricerca nel World Wide Web di Internet

Internet è vastissima, contiene una grande quantità di software, ed un oceano di informazioni sommerse. Sta crescendo col tasso del 300% annuale in tutto il mondo. Si stima che esistano circa 10 milioni di siti Web in tutto il mondo!

Per ricercare un'informazione dovreste utilizzare i motori di ricerca come "Yahoo", "Netscape", "Lycos" ecc. Recatevi su Yahoo, e fate click su search. Usate opzioni filtranti per restringere i vostri criteri di ricerca. L'azione di ricerca predefinita è "Intelligent search", che è la più generale e lista tutte le possibilità. Fate click su "Options" per scegliere ricerche per frase esatta "EXACT phrase", ricerche con "AND", "OR", ecc.. In questo modo dovreste trovare molto più rapidamente l'informazione di cui avete bisogno. Inoltre, nel menu search, ci sono pulsanti opzionali per le ricerche in Usenet, sui siti Web ed i siti Yahoo.

34 Conclusione

Dopo aver ricercato tra tutti i database **gratis** disponibili, ed il cui codice sorgente sia libero, SOLO PostgreSQL è risultato il PIÚ maturo, il più largamente usato e robusto database RDBMS SQL (orientato agli oggetti) libero/gratuito del mondo.

PostgreSQL esercita una forte atttrazione, poiché una gran mole di lavoro è già stata fatta. Esso dispone di ODBC e JDBC, usando i quali è possibile scrivere applicazioni indipendenti dai database. Le applicazioni scritte in PostgreSQL, usando i driver ODBC e JDBC, sono facilmente portabili ad altri database come Oracle, Sybase e Informix, e viceversa.

Potreste domandarvi "Ma perché PostgreSQL ?" La risposta è che, poiché sviluppare un sistema di database da zero richiede molto tempo, è più sensato scegliere un sistema di database che soddisfi le seguenti condizioni:

Un sistema di database

- Il cui codice sorgente sia disponibile: Deve essere un sistema con 'Codice Open Source'
- Che non abbia licenza vincolante, con nessun vincolo di proprietà connesso
- Che possa essere distribuito su internet
- Che sia stato sviluppato per svariati anni.
- Che soddisfi standard come SQL 92 (e SQL 89) ISO/ANSI

- Che possa soddisfare necessità future come SQL 3 (SQL 98)
- Che abbia funzionalità avanzate

"PostgreSQL" soddisfa proprio tutte queste condizioni, ed è un software adatto per questa situazione. Potreste dire che 'PostgreSQL' è un nome davvero insolito (Si pronuncia Post-gres-chiu-el, e non Postgre-es-chiu-el. È un nome davvero poco usuale, ed è molto difficile da pronunciare). Ma il mio argomento è: perché cambiare il nome? Questo mondo sarà per sempre innamorato di "PostgreSQL" !! e tutta la gente del mondo ama questo nome!!

35 FAQ - Domande su PostgreSQL

Riferitevi all'ultima versione delle FAQ per argomenti generali, Linux e Irix presso

- `<http://www.postgresql.org/docs/faq-english.shtml>`

36 Altri Formati di questo Documento

Questo documento è diffuso in 11 diversi formati, e precisamente: DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU-info, HTML, RTF(Rich Text Format), testo-puro, pagine di manuale Unix e SGML.

- Potete scaricare questo documento HOWTO come semplice file tar nei formati HTML, DVI, Postscript o SGML, da: `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/>` e `<http://www.linuxdoc.org/docs.html#howto>`
- Il formato di solo testo è presso: `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>` e `<http://www.linuxdoc.org/docs.html#howto>`
- Tutto il documento, in formato HTML, si trova in: `<http://www.linuxdoc.org/docs.html#howto>`
- Traduzioni in altri linguaggi come Francese, Tedesco, Spagnolo, Cinese, Giapponese, sono in `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>` e `<http://www.linuxdoc.org/docs.html#howto>` Qualsiasi vostro contributo, per la traduzione in altri linguaggi, è benvenuto.

Il documento è stato scritto usando uno strumento chiamato "SGML-Tools", che può essere trovato presso: `<http://www.sgmltools.org>` . Compilando il sorgente potrete dare comandi come i seguenti

- sgml2html databasehowto.sgml (per generare file html)
- sgml2rtf databasehowto.sgml (per generare file RTF)
- sgml2latex databasehowto.sgml (per generare file latex)

I documenti LaTeX possono essere convertiti in file PDF semplicemente; dapprima si produce un output Postscript, usando **sgml2latex** (e dvips), poi si ridirige l'output attraverso il comando **distill** di Acrobat (`<http://www.adobe.com>`), come segue:

```
bash$ man sgml2latex
bash$ sgml2latex nomefile.sgml
bash$ man dvips
```

```
bash$ dvips -o nomefile.ps nomefile.dvi
bash$ distill nomefile.ps
bash$ man ghostscript
bash$ man ps2pdf
bash$ ps2pdf input.ps output.pdf
bash$ acroread output.pdf &
```

Oppure potete usare il comando Ghostscript **ps2pdf**. ps2pdf è un prodotto con quasi tutte le funzionalità di Adobe Acrobat Distiller: converte file PostScript in file Portable Document Format (PDF). **ps2pdf** è implementato come un piccolissimo comando script (un file batch) che aziona Ghostscript, selezionando un "dispositivo di output" speciale chiamato **pdfwrite**. Per poter usare ps2pdf, il dispositivo pdfwrite deve essere stato incluso nel makefile quando Ghostscript è stato compilato; leggete la documentazione sulla compilazione di Ghostscript per conoscere i dettagli.

Questo documento howto è posizionato presso:

- `<http://sunsite.unc.edu/LDP/HOWTO/PostgreSQL-HOWTO.html>`

Potete trovare questo documento anche presso i seguenti siti mirror:

- `<http://www.caldera.com/LDP/HOWTO/PostgreSQL-HOWTO.html>`
- `<http://www.WGS.com/LDP/HOWTO/PostgreSQL-HOWTO.html>`
- `<http://www.cc.gatech.edu/linux/LDP/HOWTO/PostgreSQL-HOWTO.html>`
- `<http://www.redhat.com/linux-info/ldp/HOWTO/PostgreSQL-HOWTO.html>`
- Altri siti mirror a voi vicini (secondo il vostro indirizzo di rete) possono essere trovati presso `<http://sunsite.unc.edu/LDP/hmirrors.html>` scegliete un sito e recatevi direttamente nella directory /LDP/HOWTO/PostgreSQL-HOWTO.html

Per visualizzare il documento in formato dvi, usate il programma xdvi. Il programma xdvi è posizionato nel pacchetto tetex-xdvi*.rpm di Redhat Linux, che si trova seguendo il percorso dei bottoni di menu ControlPanel | Applications | Publishing | TeX . Per leggere il documento dvi date il comando:

```
xdvi -geometry 80x90 howto.dvi
man xdvi
```

E ridimensionate la finestra con il mouse. Per navigare nel documento usate i tasti freccia, i tasti Page Up, Page Down keys, e potete usare anche i tasti delle lettere 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' per muovervi su, giù, al centro, nella pagina successiva, nella precedente, ecc. Per disattivare il menu degli utenti esperti premete 'x'.

Potete leggere i file postscript usando il programma 'gv' (ghostview), oppure potete usare 'ghostscript'. Il programma ghostscript è nel pacchetto ghostscript*.rpm; il programma gv è nel pacchetto gv*.rpm di Redhat Linux; Possono essere trovati seguendo il percorso dei bottoni di menu ControlPanel | Applications | Graphics . Il programma gv è di uso molto più immediato di ghostscript. Inoltre ghostscript e gv sono disponibili su altre piattaforme come OS/2, Windows 95 e NT, e potete quindi visualizzare questi documenti anche su queste piattaforme.

- Prelevate ghostscript per Windows 95, OS/2, e per tutti i Sistemi Operativi da `<http://www.cs.wisc.edu/~ghost>`

Per leggere i documenti postscript date il comando:

```
gv howto.ps
ghostscript howto.ps
```

ATTENZIONE: Questo documento è voluminoso e, se stampato, risulterà di circa 113 pagine (postscript).

Potete leggere il documento in formato HTML usando Netscape Navigator, Microsoft Internet explorer, il Web browser Redhat Baron, o uno qualsiasi degli altri 10 browser web.

Potete leggere l'output latex e LyX usando LyX, un'interfaccia per X-Window al latex.

37 Diritto d'autore e Licenza

37.1 Copyright and License (in inglese)

37.2 Copyright and License (in italiano)

n.d.t. - L'unica licenza valida è quella originale in lingua inglese. La traduzione che segue non ha, quindi, un reale valore legale.

A Appendice A - Sintassi di ANSI/ISO SQL 1992

```
Questo file contiene una rappresentazione, trasversale ed alquanto profonda,
degli alberi sintattici del BNF per il linguaggio realizzato intorno al
27-AUG-1992 11:03:41.64.
La versione specifica del BNF qui acclusa è: solo ANSI, solo SQL2.


<SQL terminal character> ::=
      <SQL language character>
    | <SQL embedded language character>

<SQL language character> ::=
      <simple Latin letter>
    | <digit>
    | <SQL special character>

<simple Latin letter> ::=
      <simple Latin upper case letter>
    | <simple Latin lower case letter>

<simple Latin upper case letter> ::=
          A | B | C | D | E | F | G | H | I | J | K | L | M | N | O
    | P | Q | R | S | T | U | V | W | X | Y | Z

<simple Latin lower case letter> ::=
          a | b | c | d | e | f | g | h | i | j | k | l | m | n | o
    | p | q | r | s | t | u | v | w | x | y | z

<digit> ::=
    0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<SQL special character> ::=
      <space>
    | <double quote>
    | <percent>
    | <ampersand>
    | <quote>
    | <left paren>
    | <right paren>
    | <asterisk>
    | <plus sign>
    | <comma>
    | <minus sign>
    | <period>
    | <solidus>
    | <colon>
    | <semicolon>
```

```
    | <less than operator>
    | <equals operator>
    | <greater than operator>
    | <question mark>
    | <underscore>
    | <vertical bar>

<space> ::= !! <EMPHASIS>(space character in character set in use)

<double quote> ::= "

<percent> ::= %

<ampersand> ::= &

<quote> ::= '

<left paren> ::= (

<right paren> ::= )

<asterisk> ::= *

<plus sign> ::= +

<comma> ::= ,

<minus sign> ::= -

<period> ::= .

<solidus> ::= /

<colon> ::= :

<semicolon> ::= ;

<less than operator> ::= <

<equals operator> ::= =

<greater than operator> ::= >

<question mark> ::= ?

<underscore> ::= _

<vertical bar> ::= |

<SQL embedded language character> ::=
      <left bracket>
    | <right bracket>

<left bracket> ::= [
```

```
<right bracket> ::= ]

<token> ::=
      <nondelimiter token>
    | <delimiter token>

<nondelimiter token> ::=
      <regular identifier>
    | <key word>
    | <unsigned numeric literal>
    | <national character string literal>
    | <bit string literal>
    | <hex string literal>

<regular identifier> ::= <identifier body>

<identifier body> ::=
    <identifier start> [ ( <underscore> | <identifier part> )... ]

<identifier start> ::= <EMPHASIS>(!! See the Syntax Rules)

<identifier part> ::=
      <identifier start>
    | <digit>

<key word> ::=
      <reserved word>
    | <non-reserved word>

<reserved word> ::=
      ABSOLUTE | ACTION | ADD | ALL
    | ALLOCATE | ALTER | AND
    | ANY | ARE
    | AS | ASC
    | ASSERTION | AT
    | AUTHORIZATION | AVG
    | BEGIN | BETWEEN | BIT | BIT_LENGTH
    | BOTH | BY
    | CASCADE | CASCADED | CASE | CAST
    | CATALOG
    | CHAR | CHARACTER | CHAR_LENGTH
    | CHARACTER_LENGTH | CHECK | CLOSE | COALESCE
    | COLLATE | COLLATION
    | COLUMN | COMMIT
    | CONNECT
    | CONNECTION | CONSTRAINT
    | CONSTRAINTS | CONTINUE
    | CONVERT | CORRESPONDING | COUNT | CREATE | CROSS
    | CURRENT
    | CURRENT_DATE | CURRENT_TIME
    | CURRENT_TIMESTAMP | CURRENT_USER | CURSOR
    | DATE | DAY | DEALLOCATE | DEC
    | DECIMAL | DECLARE | DEFAULT | DEFERRABLE
    | DEFERRED | DELETE | DESC | DESCRIBE | DESCRIPTOR
    | DIAGNOSTICS
```

```
    | DISCONNECT | DISTINCT | DOMAIN | DOUBLE | DROP
    | ELSE | END | END-EXEC | ESCAPE
    | EXCEPT | EXCEPTION
    | EXEC | EXECUTE | EXISTS
    | EXTERNAL | EXTRACT
    | FALSE | FETCH | FIRST | FLOAT | FOR
    | FOREIGN | FOUND | FROM | FULL
    | GET | GLOBAL | GO | GOTO
    | GRANT | GROUP
    | HAVING | HOUR
    | IDENTITY | IMMEDIATE | IN | INDICATOR
    | INITIALLY | INNER | INPUT
    | INSENSITIVE | INSERT | INT | INTEGER | INTERSECT
    | INTERVAL | INTO | IS
    | ISOLATION
    | JOIN
    | KEY
    | LANGUAGE | LAST | LEADING | LEFT
    | LEVEL | LIKE | LOCAL | LOWER
    | MATCH | MAX | MIN | MINUTE | MODULE
    | MONTH
    | NAMES | NATIONAL | NATURAL | NCHAR | NEXT | NO
    | NOT | NULL
    | NULLIF | NUMERIC
    | OCTET_LENGTH | OF
    | ON | ONLY | OPEN | OPTION | OR
    | ORDER | OUTER
    | OUTPUT | OVERLAPS
    | PAD | PARTIAL | POSITION | PRECISION | PREPARE
    | PRESERVE | PRIMARY
    | PRIOR | PRIVILEGES | PROCEDURE | PUBLIC
    | READ | REAL | REFERENCES | RELATIVE | RESTRICT
    | REVOKE | RIGHT
    | ROLLBACK | ROWS
    | SCHEMA | SCROLL | SECOND | SECTION
    | SELECT
    | SESSION | SESSION_USER | SET
    | SIZE | SMALLINT | SOME | SPACE | SQL | SQLCODE
    | SQLERROR | SQLSTATE
    | SUBSTRING | SUM | SYSTEM_USER
    | TABLE | TEMPORARY
    | THEN | TIME | TIMESTAMP
    | TIMEZONE_HOUR | TIMEZONE_MINUTE
    | TO | TRAILING | TRANSACTION
    | TRANSLATE | TRANSLATION | TRIM | TRUE
    | UNION | UNIQUE | UNKNOWN | UPDATE | UPPER | USAGE
    | USER | USING
    | VALUE | VALUES | VARCHAR | VARYING | VIEW
    | WHEN | WHENEVER | WHERE | WITH | WORK | WRITE
    | YEAR
    | ZONE

<non-reserved word> ::=

      ADA
```

```
    | C | CATALOG_NAME
    | CHARACTER_SET_CATALOG | CHARACTER_SET_NAME
    | CHARACTER_SET_SCHEMA | CLASS_ORIGIN | COBOL | COLLATION_CATALOG
    | COLLATION_NAME | COLLATION_SCHEMA | COLUMN_NAME | COMMAND_FUNCTION
    | COMMITTED
    | CONDITION_NUMBER | CONNECTION_NAME | CONSTRAINT_CATALOG | CONSTRAINT_NAME
    | CONSTRAINT_SCHEMA | CURSOR_NAME
    | DATA | DATETIME_INTERVAL_CODE
    | DATETIME_INTERVAL_PRECISION | DYNAMIC_FUNCTION
    | FORTRAN
    | LENGTH
    | MESSAGE_LENGTH | MESSAGE_OCTET_LENGTH | MESSAGE_TEXT | MORE | MUMPS
    | NAME | NULLABLE | NUMBER
    | PASCAL | PLI
    | REPEATABLE | RETURNED_LENGTH | RETURNED_OCTET_LENGTH | RETURNED_SQLSTATE
    | ROW_COUNT
    | SCALE | SCHEMA_NAME | SERIALIZABLE | SERVER_NAME | SUBCLASS_ORIGIN
    | TABLE_NAME | TYPE
    | UNCOMMITTED | UNNAMED

<unsigned numeric literal> ::=
      <exact numeric literal>
    | <approximate numeric literal>

<exact numeric literal> ::=
      <unsigned integer> [ <period> [ <unsigned integer> ] ]
    | <period> <unsigned integer>

<unsigned integer> ::= <digit>...

<approximate numeric literal> ::= <mantissa> E <exponent>

<mantissa> ::= <exact numeric literal>

<exponent> ::= <signed integer>

<signed integer> ::= [ <sign> ] <unsigned integer>

<sign> ::= <plus sign> | <minus sign>

<national character string literal> ::=
    N <quote> [ <character representation>... ] <quote>
      [ ( <separator>... <quote> [ <character representation>... ] <quote> )... ]

<character representation> ::=
      <nonquote character>
    | <quote symbol>

<nonquote character> ::= !! <EMPHASIS>(See the Syntax Rules.)

<quote symbol> ::= <quote><quote>

<separator> ::= ( <comment> | <space> | <newline> )...

<comment> ::=
```

```
    <comment introducer> [ <comment character>... ] <newline>

<comment introducer> ::= <minus sign><minus sign>[<minus sign>...]

<comment character> ::=
      <nonquote character>
    | <quote>

<newline> ::= !! <EMPHASIS>(implementation-defined end-of-line indicator)

<bit string literal> ::=
    B <quote> [ <bit>... ] <quote>
      [ ( <separator>... <quote> [ <bit>... ] <quote> )... ]

<bit> ::= 0 | 1

<hex string literal> ::=
    X <quote> [ <hexit>... ] <quote>
      [ ( <separator>... <quote> [ <hexit>... ] <quote> )... ]

<hexit> ::= <digit> | A | B | C | D | E | F | a | b | c | d | e | f

<delimiter token> ::=
      <character string literal>
    | <date string>
    | <time string>
    | <timestamp string>
    | <interval string>
    | <delimited identifier>
    | <SQL special character>
    | <not equals operator>
    | <greater than or equals operator>
    | <less than or equals operator>
    | <concatenation operator>
    | <double period>
    | <left bracket>
    | <right bracket>

<character string literal> ::=
    [ <introducer><character set specification> ]
    <quote> [ <character representation>... ] <quote>
      [ ( <separator>... <quote> [ <character representation>... ] <quote> )... ]

<introducer> ::= <underscore>

<character set specification> ::=
      <standard character repertoire name>
    | <implementation-defined character repertoire name>
    | <user-defined character repertoire name>
    | <standard universal character form-of-use name>
    | <implementation-defined universal character form-of-use name>

<standard character repertoire name> ::= <character set name>

<character set name> ::= [ <schema name> <period> ]
```

```
      <SQL language identifier>

<schema name> ::=
    [ <catalog name> <period> ] <unqualified schema name>

<catalog name> ::= <identifier>

<identifier> ::=
    [ <introducer><character set specification> ] <actual identifier>

<actual identifier> ::=
      <regular identifier>
    | <delimited identifier>

<delimited identifier> ::=
    <double quote> <delimited identifier body> <double quote>

<delimited identifier body> ::= <delimited identifier part>...

<delimited identifier part> ::=
      <nondoublequote character>
    | <doublequote symbol>

<nondoublequote character> ::= <EMPHASIS>(!! See the Syntax Rules)

<doublequote symbol> ::= <double quote><double quote>

<unqualified schema name> ::= <identifier>

<SQL language identifier> ::=
    <SQL language identifier start>
       [ ( <underscore> | <SQL language identifier part> )... ]

<SQL language identifier start> ::= <simple Latin letter>

<SQL language identifier part> ::=
      <simple Latin letter>
    | <digit>

<implementation-defined character repertoire name> ::=
    <character set name>

<user-defined character repertoire name> ::= <character set name>

<standard universal character form-of-use name> ::=
    <character set name>

<implementation-defined universal character form-of-use name> ::=
    <character set name>

<date string> ::=
    <quote> <date value> <quote>

<date value> ::=
    <years value> <minus sign> <months value>
```

```
        <minus sign> <days value>

<years value> ::= <datetime value>

<datetime value> ::= <unsigned integer>

<months value> ::= <datetime value>

<days value> ::= <datetime value>

<time string> ::=
    <quote> <time value> [ <time zone interval> ] <quote>

<time value> ::=
    <hours value> <colon> <minutes value> <colon> <seconds value>

<hours value> ::= <datetime value>

<minutes value> ::= <datetime value>

<seconds value> ::=
      <seconds integer value> [ <period> [ <seconds fraction> ] ]

<seconds integer value> ::= <unsigned integer>

<seconds fraction> ::= <unsigned integer>

<time zone interval> ::=
    <sign> <hours value> <colon> <minutes value>

<timestamp string> ::=
    <quote> <date value> <space> <time value>
        [ <time zone interval> ] <quote>

<interval string> ::=
    <quote> ( <year-month literal> | <day-time literal> ) <quote>

<year-month literal> ::=
      <years value>
    | [ <years value> <minus sign> ] <months value>

<day-time literal> ::=
      <day-time interval>
    | <time interval>

<day-time interval> ::=
    <days value>
      [ <space> <hours value> [ <colon> <minutes value>
        [ <colon> <seconds value> ] ] ]

<time interval> ::=
      <hours value> [ <colon> <minutes value> [ <colon> <seconds value> ] ]
    | <minutes value> [ <colon> <seconds value> ]
    | <seconds value>
```

```
<not equals operator> ::= <>

<greater than or equals operator> ::= >=

<less than or equals operator> ::= <=

<concatenation operator> ::= ||

<double period> ::= ..

<module> ::=
    <module name clause>
    <language clause>
    <module authorization clause>
    [ <temporary table declaration>... ]
    <module contents>...

<module name clause> ::=
    MODULE [ <module name> ]
      [ <module character set specification> ]

<module name> ::= <identifier>

<module character set specification> ::=
    NAMES ARE <character set specification>

<language clause> ::=
    LANGUAGE <language name>

<language name> ::=
    ADA | C | COBOL | FORTRAN | MUMPS | PASCAL | PLI

<module authorization clause> ::=
      SCHEMA <schema name>
    | AUTHORIZATION <module authorization identifier>
    | SCHEMA <schema name>
          AUTHORIZATION <module authorization identifier>

<module authorization identifier> ::=
    <authorization identifier>

<authorization identifier> ::= <identifier>

<temporary table declaration> ::=
    DECLARE LOCAL TEMPORARY TABLE
        <qualified local table name>
      <table element list>
      [ ON COMMIT ( PRESERVE | DELETE ) ROWS ]

<qualified local table name> ::=
    MODULE <period> <local table name>

<local table name> ::= <qualified identifier>

<qualified identifier> ::= <identifier>
```

```
<table element list> ::=
      <left paren> <table element> [ ( <comma> <table element> )... ] <right paren>

<table element> ::=
      <column definition>
    | <table constraint definition>

<column definition> ::=
    <column name> ( <data type> | <domain name> )
    [ <default clause> ]
    [ <column constraint definition>... ]
    [ <collate clause> ]

<column name> ::= <identifier>

<data type> ::=
      <character string type>
           [ CHARACTER SET <character set specification> ]
    | <national character string type>
    | <bit string type>
    | <numeric type>
    | <datetime type>
    | <interval type>

<character string type> ::=
      CHARACTER [ <left paren> <length> <right paren> ]
    | CHAR [ <left paren> <length> <right paren> ]
    | CHARACTER VARYING <left paren> <length> <right paren>
    | CHAR VARYING <left paren> <length> <right paren>
    | VARCHAR <left paren> <length> <right paren>

<length> ::= <unsigned integer>

<national character string type> ::=
      NATIONAL CHARACTER [ <left paren> <length> <right paren> ]
    | NATIONAL CHAR [ <left paren> <length> <right paren> ]
    | NCHAR [ <left paren> <length> <right paren> ]
    | NATIONAL CHARACTER VARYING <left paren> <length> <right paren>
    | NATIONAL CHAR VARYING <left paren> <length> <right paren>
    | NCHAR VARYING <left paren> <length> <right paren>

<bit string type> ::=
      BIT [ <left paren> <length> <right paren> ]
    | BIT VARYING <left paren> <length> <right paren>

<numeric type> ::=
      <exact numeric type>
    | <approximate numeric type>

<exact numeric type> ::=
      NUMERIC [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | DECIMAL [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | DEC [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | INTEGER
```

```
    | INT
    | SMALLINT

<precision> ::= <unsigned integer>

<scale> ::= <unsigned integer>

<approximate numeric type> ::=
      FLOAT [ <left paren> <precision> <right paren> ]
    | REAL
    | DOUBLE PRECISION

<datetime type> ::=
      DATE
    | TIME [ <left paren> <time precision> <right paren> ]
          [ WITH TIME ZONE ]
    | TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]
          [ WITH TIME ZONE ]

<time precision> ::= <time fractional seconds precision>

<time fractional seconds precision> ::= <unsigned integer>

<timestamp precision> ::= <time fractional seconds precision>

<interval type> ::= INTERVAL <interval qualifier>

<interval qualifier> ::=
      <start field> TO <end field>
    | <single datetime field>

<start field> ::=
    <non-second datetime field>
        [ <left paren> <interval leading field precision> <right paren> ]

<non-second datetime field> ::= YEAR | MONTH | DAY | HOUR
    | MINUTE

<interval leading field precision> ::= <unsigned integer>

<end field> ::=
      <non-second datetime field>
    | SECOND [ <left paren> <interval fractional seconds precision> <right paren> ]

<interval fractional seconds precision> ::= <unsigned integer>

<single datetime field> ::=
      <non-second datetime field>
          [ <left paren> <interval leading field precision> <right paren> ]
    | SECOND [ <left paren> <interval leading field precision>
          [ <comma> <interval fractional seconds precision> ] <right paren> ]

<domain name> ::= <qualified name>

<qualified name> ::=
```

```
    [ <schema name> <period> ] <qualified identifier>

<default clause> ::=
      DEFAULT <default option>

<default option> ::=
      <literal>
    | <datetime value function>
    | USER
    | CURRENT_USER
    | SESSION_USER
    | SYSTEM_USER
    | NULL

<literal> ::=
      <signed numeric literal>
    | <general literal>

<signed numeric literal> ::=
    [ <sign> ] <unsigned numeric literal>

<general literal> ::=
      <character string literal>
    | <national character string literal>
    | <bit string literal>
    | <hex string literal>
    | <datetime literal>
    | <interval literal>

<datetime literal> ::=
      <date literal>
    | <time literal>
    | <timestamp literal>

<date literal> ::=
    DATE <date string>

<time literal> ::=
    TIME <time string>

<timestamp literal> ::=
    TIMESTAMP <timestamp string>

<interval literal> ::=
    INTERVAL [ <sign> ] <interval string> <interval qualifier>

<datetime value function> ::=
      <current date value function>
    | <current time value function>
    | <current timestamp value function>

<current date value function> ::= CURRENT_DATE

<current time value function> ::=
      CURRENT_TIME [ <left paren> <time precision> <right paren> ]
```

```
<current timestamp value function> ::=
      CURRENT_TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]

<column constraint definition> ::=
    [ <constraint name definition> ]
    <column constraint>
      [ <constraint attributes> ]

<constraint name definition> ::= CONSTRAINT <constraint name>

<constraint name> ::= <qualified name>

<column constraint> ::=
      NOT NULL
    | <unique specification>
    | <references specification>
    | <check constraint definition>

<unique specification> ::=
    UNIQUE | PRIMARY KEY

<references specification> ::=
    REFERENCES <referenced table and columns>
      [ MATCH <match type> ]
      [ <referential triggered action> ]

<referenced table and columns> ::=
     <table name> [ <left paren> <reference column list> <right paren> ]

<table name> ::=
      <qualified name>
    | <qualified local table name>

<reference column list> ::= <column name list>

<column name list> ::=
    <column name> [ ( <comma> <column name> )... ]

<match type> ::=
      FULL
    | PARTIAL

<referential triggered action> ::=
      <update rule> [ <delete rule> ]
    | <delete rule> [ <update rule> ]

<update rule> ::= ON UPDATE <referential action>

<referential action> ::=
      CASCADE
    | SET NULL
    | SET DEFAULT
    | NO ACTION
```

```
<delete rule> ::= ON DELETE <referential action>

<check constraint definition> ::=
    CHECK
        <left paren> <search condition> <right paren>

<search condition> ::=
      <boolean term>
    | <search condition> OR <boolean term>

<boolean term> ::=
      <boolean factor>
    | <boolean term> AND <boolean factor>

<boolean factor> ::=
    [ NOT ] <boolean test>

<boolean test> ::=
    <boolean primary> [ IS [ NOT ]
          <truth value> ]

<boolean primary> ::=
      <predicate>
    | <left paren> <search condition> <right paren>

<predicate> ::=
      <comparison predicate>
    | <between predicate>
    | <in predicate>
    | <like predicate>
    | <null predicate>
    | <quantified comparison predicate>
    | <exists predicate>
    | <unique predicate>
    | <match predicate>
    | <overlaps predicate>

<comparison predicate> ::=
    <row value constructor> <comp op>
        <row value constructor>

<row value constructor> ::=
       <row value constructor element>
    | <left paren> <row value constructor list> <right paren>
    | <row subquery>

<row value constructor element> ::=
      <value expression>
    | <null specification>
    | <default specification>

<value expression> ::=
      <numeric value expression>
    | <string value expression>
    | <datetime value expression>
```

```
    | <interval value expression>

<numeric value expression> ::=
      <term>
    | <numeric value expression> <plus sign> <term>
    | <numeric value expression> <minus sign> <term>

<term> ::=
      <factor>
    | <term> <asterisk> <factor>
    | <term> <solidus> <factor>

<factor> ::=
    [ <sign> ] <numeric primary>

<numeric primary> ::=
      <value expression primary>
    | <numeric value function>

<value expression primary> ::=
      <unsigned value specification>
    | <column reference>
    | <set function specification>
    | <scalar subquery>
    | <case expression>
    | <left paren> <value expression> <right paren>
    | <cast specification>

<unsigned value specification> ::=
      <unsigned literal>
    | <general value specification>

<unsigned literal> ::=
      <unsigned numeric literal>
    | <general literal>

<general value specification> ::=
      <parameter specification>
    | <dynamic parameter specification>
    | <variable specification>
    | USER
    | CURRENT_USER
    | SESSION_USER
    | SYSTEM_USER
    | VALUE

<parameter specification> ::=
    <parameter name> [ <indicator parameter> ]

<parameter name> ::= <colon> <identifier>

<indicator parameter> ::=
    [ INDICATOR ] <parameter name>

<dynamic parameter specification> ::= <question mark>
```

```
<variable specification> ::=
    <embedded variable name> [ <indicator variable> ]

<embedded variable name> ::=
    <colon><host identifier>

<host identifier> ::=
      <Ada host identifier>
    | <C host identifier>
    | <COBOL host identifier>
    | <Fortran host identifier>
    | <MUMPS host identifier>
    | <Pascal host identifier>
    | <PL/I host identifier>

<Ada host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<C host identifier> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<COBOL host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<Fortran host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<MUMPS host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<Pascal host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<PL/I host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<indicator variable> ::=
    [ INDICATOR ] <embedded variable name>

<column reference> ::= [ <qualifier> <period> ] <column name>

<qualifier> ::=
      <table name>
    | <correlation name>

<correlation name> ::= <identifier>

<set function specification> ::=
      COUNT <left paren> <asterisk> <right paren>
    | <general set function>

<general set function> ::=
      <set function type>
          <left paren> [ <set quantifier> ] <value expression> <right paren>

<set function type> ::=
    AVG | MAX | MIN | SUM | COUNT

<set quantifier> ::= DISTINCT | ALL
```

```
<scalar subquery> ::= <subquery>

<subquery> ::= <left paren> <query expression> <right paren>

<query expression> ::=
      <non-join query expression>
    | <joined table>

<non-join query expression> ::=
      <non-join query term>
    | <query expression> UNION  [ ALL ]
          [ <corresponding spec> ] <query term>
    | <query expression> EXCEPT [ ALL ]
          [ <corresponding spec> ] <query term>

<non-join query term> ::=
      <non-join query primary>
    | <query term> INTERSECT [ ALL ]
          [ <corresponding spec> ] <query primary>

<non-join query primary> ::=
      <simple table>
    | <left paren> <non-join query expression> <right paren>

<simple table> ::=
      <query specification>
    | <table value constructor>
    | <explicit table>

<query specification> ::=
    SELECT [ <set quantifier> ] <select list> <table expression>

<select list> ::=
      <asterisk>
    | <select sublist> [ ( <comma> <select sublist> )... ]

<select sublist> ::=
      <derived column>
    | <qualifier> <period> <asterisk>

<derived column> ::= <value expression> [ <as clause> ]

<as clause> ::= [ AS ] <column name>

<table expression> ::=
    <from clause>
    [ <where clause> ]
    [ <group by clause> ]
    [ <having clause> ]

<from clause> ::= FROM <table reference>
    [ ( <comma> <table reference> )... ]

<table reference> ::=
      <table name> [ [ AS ] <correlation name>
```

```
          [ <left paren> <derived column list> <right paren> ] ]
    | <derived table> [ AS ] <correlation name>
          [ <left paren> <derived column list> <right paren> ]
    | <joined table>

<derived column list> ::= <column name list>

<derived table> ::= <table subquery>

<table subquery> ::= <subquery>

<joined table> ::=
      <cross join>
    | <qualified join>
    | <left paren> <joined table> <right paren>

<cross join> ::=
    <table reference> CROSS JOIN <table reference>

<qualified join> ::=
    <table reference> [ NATURAL ] [ <join type> ] JOIN
      <table reference> [ <join specification> ]

<join type> ::=
      INNER
    | <outer join type> [ OUTER ]
    | UNION

<outer join type> ::=
      LEFT
    | RIGHT
    | FULL

<join specification> ::=
      <join condition>
    | <named columns join>

<join condition> ::= ON <search condition>

<named columns join> ::=
    USING <left paren> <join column list> <right paren>

<join column list> ::= <column name list>

<where clause> ::= WHERE <search condition>

<group by clause> ::=
    GROUP BY <grouping column reference list>

<grouping column reference list> ::=
    <grouping column reference>
        [ ( <comma> <grouping column reference> )... ]

<grouping column reference> ::=
    <column reference> [ <collate clause> ]
```

```
<collate clause> ::= COLLATE <collation name>

<collation name> ::= <qualified name>

<having clause> ::= HAVING <search condition>

<table value constructor> ::=
    VALUES <table value constructor list>

<table value constructor list> ::=
    <row value constructor> [ ( <comma> <row value constructor> )... ]

<explicit table> ::= TABLE <table name>

<query term> ::=
      <non-join query term>
    | <joined table>

<corresponding spec> ::=
    CORRESPONDING [ BY <left paren> <corresponding column list> <right paren> ]

<corresponding column list> ::= <column name list>

<query primary> ::=
      <non-join query primary>
    | <joined table>

<case expression> ::=
      <case abbreviation>
    | <case specification>

<case abbreviation> ::=
      NULLIF <left paren> <value expression> <comma>
            <value expression> <right paren>
    | COALESCE <left paren> <value expression>
            ( <comma> <value expression> )... <right paren>

<case specification> ::=
      <simple case>
    | <searched case>

<simple case> ::=
    CASE <case operand>
      <simple when clause>...
      [ <else clause> ]
    END

<case operand> ::= <value expression>

<simple when clause> ::= WHEN <when operand> THEN <result>

<when operand> ::= <value expression>

<result> ::= <result expression> | NULL
```

```
<result expression> ::= <value expression>

<else clause> ::= ELSE <result>

<searched case> ::=
    CASE
      <searched when clause>...
      [ <else clause> ]
    END

<searched when clause> ::= WHEN <search condition> THEN <result>

<cast specification> ::=
    CAST <left paren> <cast operand> AS
        <cast target> <right paren>

<cast operand> ::=
      <value expression>
    | NULL

<cast target> ::=
      <domain name>
    | <data type>

<numeric value function> ::=
      <position expression>
    | <extract expression>
    | <length expression>

<position expression> ::=
    POSITION <left paren> <character value expression>
        IN <character value expression> <right paren>

<character value expression> ::=
      <concatenation>
    | <character factor>

<concatenation> ::=
    <character value expression> <concatenation operator>
        <character factor>

<character factor> ::=
    <character primary> [ <collate clause> ]

<character primary> ::=
      <value expression primary>
    | <string value function>

<string value function> ::=
      <character value function>
    | <bit value function>

<character value function> ::=
      <character substring function>
```

```
    | <fold>
    | <form-of-use conversion>
    | <character translation>
    | <trim function>

<character substring function> ::=
    SUBSTRING <left paren> <character value expression> FROM <start position>
                [ FOR <string length> ] <right paren>

<start position> ::= <numeric value expression>

<string length> ::= <numeric value expression>

<fold> ::= ( UPPER | LOWER )
     <left paren> <character value expression> <right paren>

<form-of-use conversion> ::=
    CONVERT <left paren> <character value expression>
        USING <form-of-use conversion name> <right paren>

<form-of-use conversion name> ::= <qualified name>

<character translation> ::=
    TRANSLATE <left paren> <character value expression>
        USING <translation name> <right paren>

<translation name> ::= <qualified name>

<trim function> ::=
    TRIM <left paren> <trim operands> <right paren>

<trim operands> ::=
    [ [ <trim specification> ] [ <trim character> ] FROM ] <trim source>

<trim specification> ::=
      LEADING
    | TRAILING
    | BOTH

<trim character> ::= <character value expression>

<trim source> ::= <character value expression>

<bit value function> ::=
    <bit substring function>

<bit substring function> ::=
    SUBSTRING <left paren> <bit value expression> FROM <start position>
        [ FOR <string length> ] <right paren>

<bit value expression> ::=
      <bit concatenation>
    | <bit factor>

<bit concatenation> ::=
```

```
    <bit value expression> <concatenation operator> <bit factor>

<bit factor> ::= <bit primary>

<bit primary> ::=
      <value expression primary>
    | <string value function>

<extract expression> ::=
    EXTRACT <left paren> <extract field>
        FROM <extract source> <right paren>

<extract field> ::=
      <datetime field>
    | <time zone field>

<datetime field> ::=
      <non-second datetime field>
    | SECOND

<time zone field> ::=
      TIMEZONE_HOUR
    | TIMEZONE_MINUTE

<extract source> ::=
      <datetime value expression>
    | <interval value expression>

<datetime value expression> ::=
      <datetime term>
    | <interval value expression> <plus sign> <datetime term>
    | <datetime value expression> <plus sign> <interval term>
    | <datetime value expression> <minus sign> <interval term>

<interval term> ::=
      <interval factor>
    | <interval term 2> <asterisk> <factor>
    | <interval term 2> <solidus> <factor>
    | <term> <asterisk> <interval factor>

<interval factor> ::=
    [ <sign> ] <interval primary>

<interval primary> ::=
      <value expression primary> [ <interval qualifier> ]

<interval term 2> ::= <interval term>

<interval value expression> ::=
      <interval term>
    | <interval value expression 1> <plus sign> <interval term 1>
    | <interval value expression 1> <minus sign> <interval term 1>
    | <left paren> <datetime value expression> <minus sign>
        <datetime term> <right paren> <interval qualifier>
```

```
<interval value expression 1> ::= <interval value expression>

<interval term 1> ::= <interval term>

<datetime term> ::=
      <datetime factor>

<datetime factor> ::=
      <datetime primary> [ <time zone> ]

<datetime primary> ::=
      <value expression primary>
    | <datetime value function>

<time zone> ::=
    AT <time zone specifier>

<time zone specifier> ::=
      LOCAL
    | TIME ZONE <interval value expression>

<length expression> ::=
      <char length expression>
    | <octet length expression>
    | <bit length expression>

<char length expression> ::=
    ( CHAR_LENGTH | CHARACTER_LENGTH )
        <left paren> <string value expression> <right paren>

<string value expression> ::=
      <character value expression>
    | <bit value expression>

<octet length expression> ::=
    OCTET_LENGTH <left paren> <string value expression> <right paren>

<bit length expression> ::=
    BIT_LENGTH <left paren> <string value expression> <right paren>

<null specification> ::=
    NULL

<default specification> ::=
    DEFAULT

<row value constructor list> ::=
    <row value constructor element>
        [ ( <comma> <row value constructor element> )... ]

<row subquery> ::= <subquery>

<comp op> ::=
      <equals operator>
    | <not equals operator>
```

```
    | <less than operator>
    | <greater than operator>
    | <less than or equals operator>
    | <greater than or equals operator>

<between predicate> ::=
    <row value constructor> [ NOT ] BETWEEN
      <row value constructor> AND <row value constructor>

<in predicate> ::=
    <row value constructor>
      [ NOT ] IN <in predicate value>

<in predicate value> ::=
      <table subquery>
    | <left paren> <in value list> <right paren>

<in value list> ::=
    <value expression> ( <comma> <value expression> )...

<like predicate> ::=
    <match value> [ NOT ] LIKE <pattern>
      [ ESCAPE <escape character> ]

<match value> ::= <character value expression>

<pattern> ::= <character value expression>

<escape character> ::= <character value expression>

<null predicate> ::= <row value constructor>
    IS [ NOT ] NULL

<quantified comparison predicate> ::=
    <row value constructor> <comp op> <quantifier> <table subquery>

<quantifier> ::= <all> | <some>

<all> ::= ALL

<some> ::= SOME | ANY

<exists predicate> ::= EXISTS <table subquery>

<unique predicate> ::= UNIQUE <table subquery>

<match predicate> ::=
    <row value constructor> MATCH [ UNIQUE ]
        [ PARTIAL | FULL ] <table subquery>

<overlaps predicate> ::=
    <row value constructor 1> OVERLAPS <row value constructor 2>

<row value constructor 1> ::= <row value constructor>
```

```
<row value constructor 2> ::= <row value constructor>

<truth value> ::=
      TRUE
    | FALSE
    | UNKNOWN

<constraint attributes> ::=
      <constraint check time> [ [ NOT ] DEFERRABLE ]
    | [ NOT ] DEFERRABLE [ <constraint check time> ]

<constraint check time> ::=
      INITIALLY DEFERRED
    | INITIALLY IMMEDIATE

<table constraint definition> ::=
    [ <constraint name definition> ]
    <table constraint> [ <constraint attributes> ]

<table constraint> ::=
      <unique constraint definition>
    | <referential constraint definition>
    | <check constraint definition>

<unique constraint definition> ::=
            <unique specification> even in SQL3)
    <unique specification>
      <left paren> <unique column list> <right paren>

<unique column list> ::= <column name list>

<referential constraint definition> ::=
    FOREIGN KEY
        <left paren> <referencing columns> <right paren>
      <references specification>

<referencing columns> ::=
    <reference column list>

<module contents> ::=
      <declare cursor>
    | <dynamic declare cursor>
    | <procedure>

<declare cursor> ::=
    DECLARE <cursor name> [ INSENSITIVE ] [ SCROLL ] CURSOR
      FOR <cursor specification>

<cursor name> ::= <identifier>

<cursor specification> ::=
    <query expression> [ <order by clause> ]
      [ <updatability clause> ]

<order by clause> ::=
```

```
    ORDER BY <sort specification list>

<sort specification list> ::=
    <sort specification> [ ( <comma> <sort specification> )... ]

<sort specification> ::=
    <sort key> [ <collate clause> ] [ <ordering specification> ]

<sort key> ::=
      <column name>
    | <unsigned integer>

<ordering specification> ::= ASC | DESC

<updatability clause> ::=
    FOR
        ( READ ONLY |
          UPDATE [ OF <column name list> ] )

<dynamic declare cursor> ::=
    DECLARE <cursor name> [ INSENSITIVE ] [ SCROLL ] CURSOR
        FOR <statement name>

<statement name> ::= <identifier>

<procedure> ::=
    PROCEDURE <procedure name>
        <parameter declaration list> <semicolon>
      <SQL procedure statement> <semicolon>

<procedure name> ::= <identifier>

<parameter declaration list> ::=
      <left paren> <parameter declaration>
          [ ( <comma> <parameter declaration> )... ] <right paren>
    | <parameter declaration>...

<parameter declaration> ::=
      <parameter name> <data type>
    | <status parameter>

<status parameter> ::=
    SQLCODE | SQLSTATE

<SQL procedure statement> ::=
      <SQL schema statement>
    | <SQL data statement>
    | <SQL transaction statement>
    | <SQL connection statement>
    | <SQL session statement>
    | <SQL dynamic statement>
    | <SQL diagnostics statement>

<SQL schema statement> ::=
      <SQL schema definition statement>
```

```
    | <SQL schema manipulation statement>

<SQL schema definition statement> ::=
      <schema definition>
    | <table definition>
    | <view definition>
    | <grant statement>
    | <domain definition>
    | <character set definition>
    | <collation definition>
    | <translation definition>
    | <assertion definition>

<schema definition> ::=
    CREATE SCHEMA <schema name clause>
      [ <schema character set specification> ]
      [ <schema element>... ]

<schema name clause> ::=
      <schema name>
    | AUTHORIZATION <schema authorization identifier>
    | <schema name> AUTHORIZATION
          <schema authorization identifier>

<schema authorization identifier> ::=
    <authorization identifier>

<schema character set specification> ::=
    DEFAULT CHARACTER
        SET <character set specification>

<schema element> ::=
      <domain definition>
    | <table definition>
    | <view definition>
    | <grant statement>
    | <assertion definition>
    | <character set definition>
    | <collation definition>
    | <translation definition>

<domain definition> ::=
    CREATE DOMAIN <domain name>
        [ AS ] <data type>
      [ <default clause> ]
      [ <domain constraint>... ]
      [ <collate clause> ]

<domain constraint> ::=
    [ <constraint name definition> ]
    <check constraint definition> [ <constraint attributes> ]

<table definition> ::=
    CREATE [ ( GLOBAL | LOCAL ) TEMPORARY ] TABLE
        <table name>
```

```
      <table element list>
      [ ON COMMIT ( DELETE | PRESERVE ) ROWS ]

<view definition> ::=
    CREATE VIEW <table name> [ <left paren> <view column list>
                              <right paren> ]
      AS <query expression>
      [ WITH [ <levels clause> ] CHECK OPTION ]

<view column list> ::= <column name list>

<levels clause> ::=
    CASCADED | LOCAL

<grant statement> ::=
   GRANT <privileges> ON <object name>
     TO <grantee> [ ( <comma> <grantee> )... ]
       [ WITH GRANT OPTION ]

<privileges> ::=
      ALL PRIVILEGES
    | <action list>

<action list> ::= <action> [ ( <comma> <action> )... ]

<action> ::=
      SELECT
    | DELETE
    | INSERT [ <left paren> <privilege column list> <right paren> ]
    | UPDATE [ <left paren> <privilege column list> <right paren> ]
    | REFERENCES [ <left paren> <privilege column list> <right paren> ]
    | USAGE

<privilege column list> ::= <column name list>

<object name> ::=
      [ TABLE ] <table name>
    | DOMAIN <domain name>
    | COLLATION <collation name>
    | CHARACTER SET <character set name>
    | TRANSLATION <translation name>

<grantee> ::=
      PUBLIC
    | <authorization identifier>

<assertion definition> ::=
    CREATE ASSERTION <constraint name> <assertion check>
      [ <constraint attributes> ]

<assertion check> ::=
    CHECK
        <left paren> <search condition> <right paren>

<character set definition> ::=
```

```
    CREATE CHARACTER SET <character set name>
        [ AS ]
      <character set source>
      [ <collate clause> | <limited collation definition> ]

<character set source> ::=
      GET <existing character set name>

<existing character set name> ::=
      <standard character repertoire name>
    | <implementation-defined character repertoire name>
    | <schema character set name>

<schema character set name> ::= <character set name>

<limited collation definition> ::=
    COLLATION FROM <collation source>

<collation source> ::=
      <collating sequence definition>
    | <translation collation>

<collating sequence definition> ::=
      <external collation>
    | <schema collation name>
    | DESC <left paren> <collation name> <right paren>
    | DEFAULT

<external collation> ::=
    EXTERNAL <left paren> <quote> <external collation name> <quote> <right paren>

<external collation name> ::=
      <standard collation name>
    | <implementation-defined collation name>

<standard collation name> ::= <collation name>

<implementation-defined collation name> ::= <collation name>

<schema collation name> ::= <collation name>

<translation collation> ::=
    TRANSLATION <translation name>
        [ THEN COLLATION <collation name> ]

<collation definition> ::=
    CREATE COLLATION <collation name> FOR
        <character set specification>
      FROM <collation source>
        [ <pad attribute> ]

<pad attribute> ::=
      NO PAD
    | PAD SPACE
```

```
<translation definition> ::=
    CREATE TRANSLATION <translation name>
      FOR <source character set specification>
        TO <target character set specification>
      FROM <translation source>

<source character set specification> ::= <character set specification>

<target character set specification> ::= <character set specification>

<translation source> ::=
      <translation specification>

<translation specification> ::=
      <external translation>
    | IDENTITY
    | <schema translation name>

<external translation> ::=
    EXTERNAL <left paren> <quote> <external translation name> <quote> <right paren>

<external translation name> ::=
      <standard translation name>
    | <implementation-defined translation name>

<standard translation name> ::= <translation name>

<implementation-defined translation name> ::= <translation name>

<schema translation name> ::= <translation name>

<SQL schema manipulation statement> ::=
      <drop schema statement>
    | <alter table statement>
    | <drop table statement>
    | <drop view statement>
    | <revoke statement>
    | <alter domain statement>
    | <drop domain statement>
    | <drop character set statement>
    | <drop collation statement>
    | <drop translation statement>
    | <drop assertion statement>

<drop schema statement> ::=
    DROP SCHEMA <schema name> <drop behavior>

<drop behavior> ::= CASCADE | RESTRICT

<alter table statement> ::=
    ALTER TABLE <table name> <alter table action>

<alter table action> ::=
      <add column definition>
    | <alter column definition>
```

```
    | <drop column definition>
    | <add table constraint definition>
    | <drop table constraint definition>

<add column definition> ::=
    ADD [ COLUMN ] <column definition>

<alter column definition> ::=
    ALTER [ COLUMN ] <column name> <alter column action>

<alter column action> ::=
      <set column default clause>
    | <drop column default clause>

<set column default clause> ::=
    SET <default clause>

<drop column default clause> ::=
    DROP DEFAULT

<drop column definition> ::=
    DROP [ COLUMN ] <column name> <drop behavior>

<add table constraint definition> ::=
    ADD <table constraint definition>

<drop table constraint definition> ::=
    DROP CONSTRAINT <constraint name> <drop behavior>

<drop table statement> ::=
    DROP TABLE <table name> <drop behavior>

<drop view statement> ::=
    DROP VIEW <table name> <drop behavior>

<revoke statement> ::=
    REVOKE [ GRANT OPTION FOR ]
        <privileges>
        ON <object name>
      FROM <grantee> [ ( <comma> <grantee> )... ] <drop behavior>

<alter domain statement> ::=
    ALTER DOMAIN <domain name> <alter domain action>

<alter domain action> ::=
      <set domain default clause>
    | <drop domain default clause>
    | <add domain constraint definition>
    | <drop domain constraint definition>

<set domain default clause> ::= SET <default clause>

<drop domain default clause> ::= DROP DEFAULT

<add domain constraint definition> ::=
```

```
        ADD <domain constraint>

    <drop domain constraint definition> ::=
        DROP CONSTRAINT <constraint name>

    <drop domain statement> ::=
        DROP DOMAIN <domain name> <drop behavior>

    <drop character set statement> ::=
        DROP CHARACTER SET <character set name>

    <drop collation statement> ::=
        DROP COLLATION <collation name>

    <drop translation statement> ::=
        DROP TRANSLATION <translation name>

    <drop assertion statement> ::=
        DROP ASSERTION <constraint name>

    <SQL data statement> ::=
          <open statement>
        | <fetch statement>
        | <close statement>
        | <select statement: single row>
        | <SQL data change statement>

    <open statement> ::=
        OPEN <cursor name>

    <fetch statement> ::=
        FETCH [ [ <fetch orientation> ] FROM ]
          <cursor name> INTO <fetch target list>

    <fetch orientation> ::=
          NEXT
        | PRIOR
        | FIRST
        | LAST
        | ( ABSOLUTE | RELATIVE ) <simple value specification>

    <simple value specification> ::=
          <parameter name>
        | <embedded variable name>
        | <literal>

    <fetch target list> ::=
        <target specification> [ ( <comma> <target specification> )... ]

    <target specification> ::=
          <parameter specification>
        | <variable specification>

    <close statement> ::=
        CLOSE <cursor name>
```

```
<select statement: single row> ::=
    SELECT [ <set quantifier> ] <select list>
      INTO <select target list>
        <table expression>

<select target list> ::=
    <target specification> [ ( <comma> <target specification> )... ]

<SQL data change statement> ::=
      <delete statement: positioned>
    | <delete statement: searched>
    | <insert statement>
    | <update statement: positioned>
    | <update statement: searched>

<delete statement: positioned> ::=
    DELETE FROM <table name>
      WHERE CURRENT OF <cursor name>

<delete statement: searched> ::=
    DELETE FROM <table name>
      [ WHERE <search condition> ]

<insert statement> ::=
    INSERT INTO <table name>
      <insert columns and source>

<insert columns and source> ::=
      [ <left paren> <insert column list> <right paren> ]
            <query expression>
    | DEFAULT VALUES

<insert column list> ::= <column name list>

<update statement: positioned> ::=
    UPDATE <table name>
      SET <set clause list>
        WHERE CURRENT OF <cursor name>

<set clause list> ::=
    <set clause> [ ( <comma> <set clause> )... ]

<set clause> ::=
    <object column> <equals operator> <update source>

<object column> ::= <column name>

<update source> ::=
      <value expression>
    | <null specification>
    | DEFAULT

<update statement: searched> ::=
    UPDATE <table name>
```

```
      SET <set clause list>
      [ WHERE <search condition> ]

<SQL transaction statement> ::=
      <set transaction statement>
    | <set constraints mode statement>
    | <commit statement>
    | <rollback statement>

<set transaction statement> ::=
    SET TRANSACTION <transaction mode>
        [ ( <comma> <transaction mode> )... ]

<transaction mode> ::=
      <isolation level>
    | <transaction access mode>
    | <diagnostics size>

<isolation level> ::=
    ISOLATION LEVEL <level of isolation>

<level of isolation> ::=
      READ UNCOMMITTED
    | READ COMMITTED
    | REPEATABLE READ
    | SERIALIZABLE

<transaction access mode> ::=
      READ ONLY
    | READ WRITE

<diagnostics size> ::=
    DIAGNOSTICS SIZE <number of conditions>

<number of conditions> ::= <simple value specification>

<set constraints mode statement> ::=
    SET CONSTRAINTS <constraint name list>
        ( DEFERRED | IMMEDIATE )

<constraint name list> ::=
      ALL
    | <constraint name> [ ( <comma> <constraint name> )... ]

<commit statement> ::=
    COMMIT [ WORK ]

<rollback statement> ::=
    ROLLBACK [ WORK ]

<SQL connection statement> ::=
      <connect statement>
    | <set connection statement>
    | <disconnect statement>
```

```
<connect statement> ::=
    CONNECT TO <connection target>

<connection target> ::=
      <SQL-server name>
        [ AS <connection name> ]
          correspondence with Tony Gordon)
        [ USER <user name> ]
    | DEFAULT

<SQL-server name> ::= <simple value specification>

<connection name> ::= <simple value specification>

<user name> ::= <simple value specification>

<set connection statement> ::=
    SET CONNECTION <connection object>

<connection object> ::=
      DEFAULT
    | <connection name>

<disconnect statement> ::=
    DISCONNECT <disconnect object>

<disconnect object> ::=
      <connection object>
    | ALL
    | CURRENT

<SQL session statement> ::=
      <set catalog statement>
    | <set schema statement>
    | <set names statement>
    | <set session authorization identifier statement>
    | <set local time zone statement>

<set catalog statement> ::=
    SET CATALOG <value specification>

<value specification> ::=
      <literal>
    | <general value specification>

<set schema statement> ::=
    SET SCHEMA <value specification>

<set names statement> ::=
    SET NAMES <value specification>

<set session authorization identifier statement> ::=
    SET SESSION AUTHORIZATION
        <value specification>
```

```
<set local time zone statement> ::=
    SET TIME ZONE
        <set time zone value>

<set time zone value> ::=
      <interval value expression>
    | LOCAL

<SQL dynamic statement> ::=
      <system descriptor statement>
    | <prepare statement>
    | <deallocate prepared statement>
    | <describe statement>
    | <execute statement>
    | <execute immediate statement>
    | <SQL dynamic data statement>

<system descriptor statement> ::=
      <allocate descriptor statement>
    | <deallocate descriptor statement>
    | <set descriptor statement>
    | <get descriptor statement>

<allocate descriptor statement> ::=
    ALLOCATE DESCRIPTOR <descriptor name>
       [ WITH MAX <occurrences> ]

<descriptor name> ::=
    [ <scope option> ] <simple value specification>

<scope option> ::=
      GLOBAL
    | LOCAL

<occurrences> ::= <simple value specification>

<deallocate descriptor statement> ::=
    DEALLOCATE DESCRIPTOR <descriptor name>

<set descriptor statement> ::=
    SET DESCRIPTOR <descriptor name>
        <set descriptor information>

<set descriptor information> ::=
      <set count>
    | VALUE <item number>
        <set item information> [ ( <comma> <set item information> )... ]

<set count> ::=
    COUNT <equals operator> <simple value specification 1>

<simple value specification 1> ::= <simple value specification>

<item number> ::= <simple value specification>
```

```
<set item information> ::=
    <descriptor item name> <equals operator> <simple value specification 2>

<descriptor item name> ::=
      TYPE
    | LENGTH
    | OCTET_LENGTH
    | RETURNED_LENGTH
    | RETURNED_OCTET_LENGTH
    | PRECISION
    | SCALE
    | DATETIME_INTERVAL_CODE
    | DATETIME_INTERVAL_PRECISION
    | NULLABLE
    | INDICATOR
    | DATA
    | NAME
    | UNNAMED
    | COLLATION_CATALOG
    | COLLATION_SCHEMA
    | COLLATION_NAME
    | CHARACTER_SET_CATALOG
    | CHARACTER_SET_SCHEMA
    | CHARACTER_SET_NAME

<simple value specification 2> ::= <simple value specification>

<item number> ::= <simple value specification>

<get descriptor statement> ::=
    GET DESCRIPTOR <descriptor name> <get descriptor information>

<get descriptor information> ::=
      <get count>
    | VALUE <item number>
        <get item information> [ ( <comma> <get item information> )... ]

<get count> ::=
    <simple target specification 1> <equals operator>
         COUNT

<simple target specification 1> ::= <simple target specification>

<simple target specification> ::=
      <parameter name>
    | <embedded variable name>

<get item information> ::=
    <simple target specification 2> <equals operator> <descriptor item name>>

<simple target specification 2> ::= <simple target specification>

<prepare statement> ::=
    PREPARE <SQL statement name> FROM <SQL statement variable>
```

```
<SQL statement name> ::=
      <statement name>
    | <extended statement name>

<extended statement name> ::=
    [ <scope option> ] <simple value specification>

<SQL statement variable> ::= <simple value specification>

<deallocate prepared statement> ::=
    DEALLOCATE PREPARE <SQL statement name>

<describe statement> ::=
      <describe input statement>
    | <describe output statement>

<describe input statement> ::=
    DESCRIBE INPUT <SQL statement name> <using descriptor>

<using descriptor> ::=
    ( USING | INTO ) SQL DESCRIPTOR <descriptor name>

<describe output statement> ::=
    DESCRIBE [ OUTPUT ] <SQL statement name> <using descriptor>

<execute statement> ::=
    EXECUTE <SQL statement name>
      [ <result using clause> ]
      [ <parameter using clause> ]

<result using clause> ::= <using clause>

<using clause> ::=
      <using arguments>
    | <using descriptor>

<using arguments> ::=
    ( USING | INTO ) <argument> [ ( <comma> <argument> )... ]

<argument> ::= <target specification>

<parameter using clause> ::= <using clause>

<execute immediate statement> ::=
    EXECUTE IMMEDIATE <SQL statement variable>

<SQL dynamic data statement> ::=
      <allocate cursor statement>
    | <dynamic open statement>
    | <dynamic fetch statement>
    | <dynamic close statement>
    | <dynamic delete statement: positioned>
    | <dynamic update statement: positioned>

<allocate cursor statement> ::=
```

```
    ALLOCATE <extended cursor name> [ INSENSITIVE ]
        [ SCROLL ] CURSOR
      FOR <extended statement name>

<extended cursor name> ::=
    [ <scope option> ] <simple value specification>

<dynamic open statement> ::=
    OPEN <dynamic cursor name> [ <using clause> ]

<dynamic cursor name> ::=
      <cursor name>
    | <extended cursor name>

<dynamic fetch statement> ::=
    FETCH [ [ <fetch orientation> ] FROM ] <dynamic cursor name>
        <using clause>

<dynamic close statement> ::=
    CLOSE <dynamic cursor name>

<dynamic delete statement: positioned> ::=
    DELETE FROM <table name>
      WHERE CURRENT OF
          <dynamic cursor name>

<dynamic update statement: positioned> ::=
    UPDATE <table name>
      SET <set clause>
          [ ( <comma> <set clause> )... ]
        WHERE CURRENT OF
            <dynamic cursor name>

<SQL diagnostics statement> ::=
    <get diagnostics statement>

<get diagnostics statement> ::=
    GET DIAGNOSTICS <sql diagnostics information>

<sql diagnostics information> ::=
      <statement information>
    | <condition information>

<statement information> ::=
    <statement information item> [ ( <comma> <statement information item> )... ]

<statement information item> ::=
    <simple target specification> <equals operator> <statement information item name>

<statement information item name> ::=
      NUMBER
    | MORE
    | COMMAND_FUNCTION
    | DYNAMIC_FUNCTION
    | ROW_COUNT
```

```
<condition information> ::=
    EXCEPTION <condition number>
      <condition information item> [ ( <comma> <condition information item> )... ]

<condition number> ::= <simple value specification>

<condition information item> ::=
    <simple target specification> <equals operator> <condition information item name>

<condition information item name> ::=
      CONDITION_NUMBER
    | RETURNED_SQLSTATE
    | CLASS_ORIGIN
    | SUBCLASS_ORIGIN
    | SERVER_NAME
    | CONNECTION_NAME
    | CONSTRAINT_CATALOG
    | CONSTRAINT_SCHEMA
    | CONSTRAINT_NAME
    | CATALOG_NAME
    | SCHEMA_NAME
    | TABLE_NAME
    | COLUMN_NAME
    | CURSOR_NAME
    | MESSAGE_TEXT
    | MESSAGE_LENGTH
    | MESSAGE_OCTET_LENGTH

<embedded SQL host program> ::=
      <embedded SQL Ada program>
    | <embedded SQL C program>
    | <embedded SQL COBOL program>
    | <embedded SQL Fortran program>
    | <embedded SQL MUMPS program>
    | <embedded SQL Pascal program>
    | <embedded SQL PL/I program>

<embedded SQL Ada program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL C program> ::=
      !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL COBOL program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL Fortran program> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL MUMPS program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL Pascal program> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL PL/I program> ::= !! <EMPHASIS>(See the Syntax Rules.)
```

```
<embedded SQL declare section> ::=
      <embedded SQL begin declare>
        [ <embedded character set declaration> ]
        [ <host variable definition>... ]
      <embedded SQL end declare>
    | <embedded SQL MUMPS declare>

<embedded SQL begin declare> ::=
    <SQL prefix> BEGIN DECLARE SECTION
        [ <SQL terminator> ]

<SQL prefix> ::=
      EXEC SQL
    | <ampersand>SQL<left paren>

<SQL terminator> ::=
      END-EXEC
    | <semicolon>
    | <right paren>

<embedded character set declaration> ::=
    SQL NAMES ARE <character set specification>

<host variable definition> ::=
      <Ada variable definition>
    | <C variable definition>
    | <COBOL variable definition>
    | <Fortran variable definition>
    | <MUMPS variable definition>
    | <Pascal variable definition>
    | <PL/I variable definition>

<Ada variable definition> ::=
    <Ada host identifier> [ ( <comma> <Ada host identifier> )... ] :
    <Ada type specification> [ <Ada initial value> ]

<Ada type specification> ::=
      <Ada qualified type specification>
    | <Ada unqualified type specification>

<Ada qualified type specification> ::=
      SQL_STANDARD.CHAR [ CHARACTER SET
         [ IS ] <character set specification> ]
          <left paren> 1 <double period> <length> <right paren>
    | SQL_STANDARD.BIT
          <left paren> 1 <double period> <length> <right paren>
    | SQL_STANDARD.SMALLINT
    | SQL_STANDARD.INT
    | SQL_STANDARD.REAL
    | SQL_STANDARD.DOUBLE_PRECISION
    | SQL_STANDARD.SQLCODE_TYPE
    | SQL_STANDARD.SQLSTATE_TYPE
    | SQL_STANDARD.INDICATOR_TYPE

<Ada unqualified type specification> ::=
```

```
      CHAR
          <left paren> 1 <double period> <length> <right paren>
    | BIT
          <left paren> 1 <double period> <length> <right paren>
    | SMALLINT
    | INT
    | REAL
    | DOUBLE_PRECISION
    | SQLCODE_TYPE
    | SQLSTATE_TYPE
    | INDICATOR_TYPE

<Ada initial value> ::=
    <Ada assignment operator> <character representation>...

<Ada assignment operator> ::= <colon><equals operator>

<C variable definition> ::=
      [ <C storage class> ]
      [ <C class modifier> ]
      <C variable specification>
    <semicolon>

<C storage class> ::=
      auto
    | extern
    | static

<C class modifier> ::= const | volatile

<C variable specification> ::=
      <C numeric variable>
    | <C character variable>
    | <C derived variable>

<C numeric variable> ::=
    ( long | short | float | double )
      <C host identifier> [ <C initial value> ]
            [ ( <comma> <C host identifier> [ <C initial value> ] )... ]

<C initial value> ::=
    <equals operator> <character representation>...

<C character variable> ::=
    char [ CHARACTER SET
             [ IS ] <character set specification> ]
      <C host identifier>
        <C array specification> [ <C initial value> ]
        [ ( <comma> <C host identifier>
          <C array specification>
                 [ <C initial value> ] )... ]

<C array specification> ::=
    <left bracket> <length> <right bracket>
```

```
<C derived variable> ::=
      <C VARCHAR variable>
    | <C bit variable>

<C VARCHAR variable> ::=
    VARCHAR [ CHARACTER SET [ IS ]
        <character set specification> ]
        <C host identifier>
            <C array specification> [ <C initial value> ]
          [ ( <comma> <C host identifier>
              <C array specification>
                      [ <C initial value> ] )... ]

<C bit variable> ::=
    BIT <C host identifier>
        <C array specification> [ <C initial value> ]
      [ ( <comma> <C host identifier>
        <C array specification>
                   [ <C initial value> ] )... ]

<COBOL variable definition> ::=
    (01|77) <COBOL host identifier> <COBOL type specification>
      [ <character representation>... ] <period>

<COBOL type specification> ::=
      <COBOL character type>
    | <COBOL bit type>
    | <COBOL numeric type>
    | <COBOL integer type>

<COBOL character type> ::=
    [ CHARACTER SET [ IS ]
          <character set specification> ]
    ( PIC | PICTURE ) [ IS ] ( X [ <left paren> <length> <right paren> ] )...

<COBOL bit type> ::=
    ( PIC | PICTURE ) [ IS ]
        ( B [ <left paren> <length> <right paren> ] )...

<COBOL numeric type> ::=
    ( PIC | PICTURE ) [ IS ]
      S <COBOL nines specification>
    [ USAGE [ IS ] ] DISPLAY SIGN LEADING SEPARATE

<COBOL nines specification> ::=
      <COBOL nines> [ V [ <COBOL nines> ] ]
    | V <COBOL nines>

<COBOL nines> ::= ( 9 [ <left paren> <length> <right paren> ] )...

<COBOL integer type> ::=
      <COBOL computational integer>
    | <COBOL binary integer>

<COBOL computational integer> ::=
```

```
    ( PIC | PICTURE ) [ IS ] S<COBOL nines>
      [ USAGE [ IS ] ] ( COMP | COMPUTATIONAL )

<COBOL binary integer> ::=
    ( PIC | PICTURE ) [ IS ] S<COBOL nines>
      [ USAGE [ IS ] ] BINARY

<Fortran variable definition> ::=
    <Fortran type specification>
    <Fortran host identifier>
        [ ( <comma> <Fortran host identifier> )... ]

<Fortran type specification> ::=
      CHARACTER [ <asterisk> <length> ]
          [ CHARACTER SET [ IS ]
                <character set specification> ]
    | BIT [ <asterisk> <length> ]
    | INTEGER
    | REAL
    | DOUBLE PRECISION

<MUMPS variable definition> ::=
    ( <MUMPS numeric variable> | <MUMPS character variable> )
        <semicolon>

<MUMPS numeric variable> ::=
    <MUMPS type specification>
      <MUMPS host identifier> [ ( <comma> <MUMPS host identifier> )... ]

<MUMPS type specification> ::=
      INT
    | DEC
          [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | REAL

<MUMPS character variable> ::=
    VARCHAR <MUMPS host identifier> <MUMPS length specification>
      [ ( <comma> <MUMPS host identifier> <MUMPS length specification> )... ]

<MUMPS length specification> ::=
    <left paren> <length> <right paren>

<Pascal variable definition> ::=
    <Pascal host identifier> [ ( <comma> <Pascal host identifier> )... ] <colon>
      <Pascal type specification> <semicolon>

<Pascal type specification> ::=
      PACKED ARRAY
          <left bracket> 1 <double period> <length> <right bracket>
        OF CHAR
          [ CHARACTER SET [ IS ]
                <character set specification> ]
    | PACKED ARRAY
          <left bracket> 1 <double period> <length> <right bracket>
        OF BIT
```

```
    | INTEGER
    | REAL
    | CHAR [ CHARACTER SET
                               [ IS ] <character set specification> ]
    | BIT

<PL/I variable definition> ::=
    (DCL | DECLARE)
        (   <PL/I host identifier>
          | <left paren> <PL/I host identifier>
                [ ( <comma> <PL/I host identifier> )... ] <right paren> )
    <PL/I type specification>
    [ <character representation>... ] <semicolon>

<PL/I type specification> ::=
      ( CHAR | CHARACTER ) [ VARYING ]
          <left paren><length><right paren>
          [ CHARACTER SET
                [ IS ] <character set specification> ]
    | BIT [ VARYING ] <left paren><length><right paren>
    | <PL/I type fixed decimal> <left paren> <precision>
          [ <comma> <scale> ] <right paren>
    | <PL/I type fixed binary> [ <left paren> <precision> <right paren> ]
    | <PL/I type float binary> <left paren> <precision> <right paren>

<PL/I type fixed decimal> ::=
      ( DEC | DECIMAL ) FIXED
    | FIXED ( DEC | DECIMAL )

<PL/I type fixed binary> ::=
      ( BIN | BINARY ) FIXED
    | FIXED ( BIN | BINARY )

<PL/I type float binary> ::=
      ( BIN | BINARY ) FLOAT
    | FLOAT ( BIN | BINARY )

<embedded SQL end declare> ::=
    <SQL prefix> END DECLARE SECTION
        [ <SQL terminator> ]

<embedded SQL MUMPS declare> ::=
    <SQL prefix>
      BEGIN DECLARE SECTION
        [ <embedded character set declaration> ]
        [ <host variable definition>... ]
      END DECLARE SECTION
    <SQL terminator>

<embedded SQL statement> ::=
    <SQL prefix>
      <statement or declaration>
    [ <SQL terminator> ]

<statement or declaration> ::=
```

```
      <declare cursor>
    | <dynamic declare cursor>
    | <temporary table declaration>
    | <embedded exception declaration>
    | <SQL procedure statement>

<embedded exception declaration> ::=
    WHENEVER <condition> <condition action>

<condition> ::=
    SQLERROR | NOT FOUND

<condition action> ::=
    CONTINUE | <go to>

<go to> ::=
    ( GOTO | GO TO ) <goto target>

<goto target> ::=
      <host label identifier>
    | <unsigned integer>
    | <host PL/I label variable>

<host label identifier> ::= !!<EMPHASIS>(See the Syntax Rules.)

<host PL/I label variable> ::= !!<EMPHASIS>(See the Syntax Rules.)

<preparable statement> ::=
      <preparable SQL data statement>
    | <preparable SQL schema statement>
    | <preparable SQL transaction statement>
    | <preparable SQL session statement>
    | <preparable implementation-defined statement>

<preparable SQL data statement> ::=
      <delete statement: searched>
    | <dynamic single row select statement>
    | <insert statement>
    | <dynamic select statement>
    | <update statement: searched>
    | <preparable dynamic delete statement: positioned>
    | <preparable dynamic update statement: positioned>

<dynamic single row select statement> ::= <query specification>

<dynamic select statement> ::= <cursor specification>

<preparable dynamic delete statement: positioned> ::=
   DELETE [ FROM <table name> ]
      WHERE CURRENT OF <cursor name>

<preparable dynamic update statement: positioned> ::=
   UPDATE [ <table name> ]
      SET <set clause list>
      WHERE CURRENT OF <cursor name>
```

```
<preparable SQL schema statement> ::=
      <SQL schema statement>

<preparable SQL transaction statement> ::=
      <SQL transaction statement>

<preparable SQL session statement> ::=
      <SQL session statement>

<preparable implementation-defined statement> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<direct SQL statement> ::=
    <directly executable statement> <semicolon>

<directly executable statement> ::=
      <direct SQL data statement>
    | <SQL schema statement>
    | <SQL transaction statement>
    | <SQL connection statement>
    | <SQL session statement>
    | <direct implementation-defined statement>

<direct SQL data statement> ::=
      <delete statement: searched>
    | <direct select statement: multiple rows>
    | <insert statement>
    | <update statement: searched>
    | <temporary table declaration>

<direct select statement: multiple rows> ::=
    <query expression> [ <order by clause> ]

<direct implementation-defined statement> ::=
    !!<EMPHASIS>(See the Syntax Rules)

<SQL object identifier> ::=
    <SQL provenance> <SQL variant>

<SQL provenance> ::= <arc1> <arc2> <arc3>

<arc1> ::= iso | 1 | iso <left paren> 1 <right paren>

<arc2> ::= standard | 0 | standard <left paren> 0 <right paren>

<arc3> ::= 9075

<SQL variant> ::= <SQL edition> <SQL conformance>

<SQL edition> ::= <1987> | <1989> | <1992>

<1987> ::= 0 | edition1987 <left paren> 0 <right paren>

<1989> ::= <1989 base> <1989 package>
```

```
<1989 base> ::= 1 | edition1989 <left paren> 1 <right paren>

<1989 package> ::= <integrity no> | <integrity yes>

<integrity no> ::= 0 | IntegrityNo <left paren> 0 <right paren>

<integrity yes> ::= 1 | IntegrityYes <left paren> 1 <right paren>

<1992> ::= 2 | edition1992 <left paren> 2 <right paren>

<SQL conformance> ::= <low> | <intermediate> | <high>

<low> ::= 0 | Low <left paren> 0 <right paren>

<intermediate> ::= 1 | Intermediate <left paren> 1 <right paren>

<high> ::= 2 | High <left paren> 2 <right paren>
```

B Appendice B - Tutorial su SQL per principianti

B.1 Tutorial per PostgreSQL

Il tutorial su SQL è distribuito insieme a PostgreSQL. Gli script del tutorial su SQL sono nella directory src/tutorial

B.2 Puntatori ad URL in Internet

Tutorial su SQL per principianti possono essere trovati presso

- tutorial di Jim Hoffman `<http://w3.one.net/~jhoffman/sqltut.htm>`
- Carnegie Mellon Univ `<http://www.heinz.cmu.edu/project/dbms>` . Arrivati qui, fate click su 'technical'->'SQL_examples.html' e sul resto.
- Concord Univ `<http://www.cs.concordia.ca/Course_Notes/oracle/browser/node1.html>`

Commenti o suggerimenti? Spedite una mail a

- Jim Hoffman *jhoffman@one.net*

Seguono i siti suggeriti da John Hoffman:

- SQL Reference `<http://www.contrib.andrew.cmu.edu/~shadow/sql.html>`
- Ask the SQL Pro `<http://www.inquiry.com/techtips/thesqlpro/>`
- Siti Utili sui DB Relazionali di SQL Pro `<http://www.inquiry.com/techtips/thesqlpro/usefulsites.html>`
- Sorgenti del Programmatore `<http://infoweb.magi.com/~steve/develop.html>`
- Siti sui DBMS `<http://info.itu.ch/special/wwwfiles>` Arrivati qui leggete il file comp_db.html
- DB Ingredients `<http://www.compapp.dcu.ie/databases/f017.html>`

- Web Authoring <http://www.stars.com/Tutorial/CGI/>
- Computing Dictionary <http://wfn-shop.princeton.edu/cgi-bin/foldoc>
- DBMS Lab/Links <http://www-ccs.cs.umass.edu/db.html>
- FAQ SQL <http://epoch.CS.Berkeley.EDU:8000/sequoia/dba/montage/FAQ> Arrivati qui, leggete il file SQL_TOC.html
- Database SQL <http://chaos.mur.csu.edu.au/itc125/cgi/sqldb.html>
- Pagina RIT sulla Progettazione di Database <http://www.it.rit.edu/~wjs/IT/199602/icsa720/icsa720postings.html>
- Sito Jump Database <http://www.pcslink.com/~ej/dbweb.html>
- Tutorial sulla Programmazione nel Web <http://www.eng.uc.edu/~jtilley/tutorial.html>
- Risorse per lo Sviluppo <http://www.ndev.com/ndc2/support/resources.htp>
- Lista di Query <http://ashok.pair.com/sql.htm>
- IMAGE SQL Miscellaneous <http://jazz.external.hp.com/training/sqltables/main.html>
- Lista di Risorse in Internet <http://www.eit.com/web/netservices.html>

B.3 Tutorial On-line su SQL

Visitate i seguenti siti con tutorial on-line su SQL

- Corso per principianti su SQL <http://sqlcourse.com>
- Corso avanzato su SQL <http://sqlcourse2.com>

C Appendice C - Istruzioni per un'Installazione Rapida di Linux

Se avete in mente di usare PostgreSQL su Linux, e vi serve una mano per installare Linux, visitate i puntatori suggeriti in quest'Appendice. Vi troverete trattati i seguenti argomenti:

- Caratteristiche Rilevanti di Linux: Perché Linux risulta migliore, come server database, se confrontato con Windows 95/NT
- Istruzioni per una Rapida Installazione di Linux in 10 minuti
- Lista di Analogie Microsoft-Linux
- Ricompilazione del Kernel Linux in Pochi Passi

- Sito principale presso <http://www.aldev.8m.com> siti mirror: *webjump* <http://aldev.webjump.com>, *angelfire* <http://www.angelfire.com/nv/aldev>, *geocities* <http://www.geocities.com/alavoor/index.html>, *virtualave* <http://aldev.virtualave.net>, *bizland* <http://aldev.bizland.com>, *theglobe* <http://members.theglobe.com/aldev/index.html>, *spree* <http://members.spree.com/technology/aldev>, *infoseek* <http://homepages.infoseek.com/~aldev1/index.html>, *bcity* <http://www3.bcity.com/aldev>, *50megs* <http://aldev.50megs.com>

D Appendice C - Installazione di Midgard

Attualmente gli RPM di Midgard, ottenibili dalla locazione <http://www.midgard-project.org/download/binaries>, non comprendono PostgreSQL, e quindi dovrete installare a partire dall'archivio tar dei sorgenti.

Scaricate l'archivio tar dei sorgenti di Midgard, e leggete il file INSTALL.REDHAT :

```
bash# cd midgard-lib-1.4beta6
bash# ./configure --prefix=/usr/local --with-mysql=/usr/local --includedir=/usr/include/mysql --with-midgard=/usr/local --with-pgsql=/var/lib/pgsql --includedir=/usr/include/pgsql
bash# make
bash# make install
bash# ldconfig -v | grep -i midga
Copiate i file header, nel caso in cui questo non venisse fatto dal make install..
bash# cp *.h /usr/local/include


bash# cd ../mod_midgard-1.4beta5c
bash# ./configure --prefix=/usr/local --with-mysql=/usr/local --includedir=/usr/include/mysql --with-midgard=/usr --with-pgsql=/var/lib/pgsql --includedir=/usr/include/pgsql
bash# make
bash# make install
#modificate una riga di apache per correggere /usr/.....
bash# vi /etc/httpd/conf/httpd.conf   (oppure /etc/apache/httpd.conf)
bash# /etc/init.d/apache restart
#ora apache dovrebbe ripartire!!!


bash# cd ../midgard-php-1.4beta6
bash# ./configure '--with-apxs' '--with-mysql' '--with-pgsql' '--with-midgard' --prefix=/usr/local --with-midgard=/usr/local

bash# gvim Makefile
Aggiungete -I/usr/include/pgsql per la variabile INCLUDE.

Aggiungete anche $(INCLUDE) al comando $(APXS) come segue:
libphp3.so: mod_php3.c libmodphp3-so.a  pcrelib/libpcre.a midgard/libphpmidgard.a
        -@test -f ./mod_php3.c || test -L ./mod_php3.c || $(LN_S) $(srcdir)/mod_php3.c ./mod_php3.c
        -@test -f ./mod_php3.c || test -h ./mod_php3.c || $(LN_S) $(srcdir)/mod_php3.c ./mod_php3.c
        $(APXS) -c -o libphp3.so  -I$(srcdir) $(INCLUDE) -I. -I/usr/local/include -I/usr/lib/glib/include  -Wl,'-rpath /usr/local/lib' ./mod_php3.c libmodphp3-so.a $(APXS_LDFLAGS)

bash# make
bash# make install
#modificate una riga di apache per correggere /usr/.....
# e aggiungete righe come queste:
        LoadModule php4_module        modules/libphp4.so
        AddModule mod_php4.c
        LoadModule php4_module        lib/apache/libphp4.so

        <IfModule mod_php4.c>
                AddType application/x-httpd-php4 .php4
                AddType application/x-httpd-php4 .php
                AddType application/x-httpd-php4-source .phps
                AddType application/x-httpd-php .php
```

```
        </IfModule>

bash# vi /etc/httpd/conf/httpd.conf   (oppure /etc/apache/httpd.conf)

bash# /etc/init.d/apache restart
#ora apache dovrebbe ripartire!!!
```

D.1 Provare il Server PHP Midgard

Per provare l'installazione fate quanto segue: Create un file nella vostra directory root per i documenti. Di solito io lo chiamo info.php, e ci inserisco dentro quest'unica riga:

```
<?phpinfo()?>
```

Poi caricatela nel vostro browser: http://localhost/info.php

Dovreste vedere una bella pagina che riassume tutte le informazioni possibili riguardo alle vostre impostazioni. Probabilmente non dovreste lasciare questo file su un server di produzione, ma è molto utile a fini di debugging e per conoscere le informazioni generali nella fase di sviluppo.

D.2 Security OpenSSL

Potreste anche dover prelevare il pacchetto RSA, che consente la cifratura SSL, dal sito <ftp://ftp.deva.net/pub/sources/crypto/rsaref20.1996.tar.Z> Vedere anche il pacchetto RPM di OpenSSL sul cdrom di Linux (<http://www.openssl.org>).

Se non volete abilitare SSL (o se incontrate qualche difficoltà), scaricate il pacchetto RPM del sorgente di Apache-Midgard, modificate il file *.spec, commentando la parte riguardante SSL, e ricostruite l'RPM.