



Basi di dati



Introduzione a PostgreSQL

Introduzione a PostgreSQL

PostgreSQL è un software relazionale e ad oggetti per la gestione di basi di dati

- PostgreSQL è Open-Source ed il suo sviluppo procede da 15 anni
 - il suo codice sorgente è quindi disponibile pubblicamente ed "aperto" ai contributi degli sviluppatori che volontariamente intendono partecipare alla sua implementazione
- In **PostgreSQL**, i dati vengono rappresentati attraverso delle tabelle e le tabelle sono gestite utilizzando un linguaggio di alto livello chiamato **SQL**.
- è scaricabile dalla rete all'indirizzo **<http://www.postgresql.org/download>**

Introduzione a PostgreSQL

Supporta una parte molto grande dello standard SQL e offre molte altre funzionalità:

- ▶ query complesse
- ▶ chiavi esterne
- ▶ trigger
- ▶ viste
- ▶ integrità transazionale
- ▶ controllo concorrente multiversione



Introduzione a PostgreSQL

L'utilizzo tipico di Postgres come database enterprise è quello di installare un server unico:

- tutti gli utenti operano su dati condivisi ed in modo concorrente



Introduzione a PostgreSQL

- PostgreSQL è suddiviso in due applicazioni principali:
 - **Server:**
 - **Postmaster**
 - su di esso gravano le incombenze relative al funzionamento del software; una volta avviato esso rimane in "ascolto" e attende le connessioni.
 - per ciascuna richiesta di connessione verso il servizio principale viene lanciato un corrispondente processo server.
 - **Client:**
 - **PGAdmin III** (interfaccia grafica)
 - **Psql** (interfaccia a linea di comando)
- ▶ Per le esercitazioni utilizzeremo il client PGAdmin III

Introduzione a PostgreSQL

Postmaster per poter funzionare ha bisogno di un *cluster* presente nel *file system*

- Un **cluster** è una directory di sistema formattata per accogliere una determinata istanza, all'interno di essa saranno reperibili il **catalogo** di sistema, i template e le basi di dati.
- Per consentire l'avvio dei processi "figli", il proprietario della directory dovrà essere lo stesso del **Postmaster**.
- Un **catalogo** di sistema si presenta come un insieme di tabelle in comune tra i database ospitati nel cluster, inoltre in esso è presente il *Data Dictionary*, un vero e proprio "vocabolario" per il parsing delle istruzioni scritte in linguaggio **SQL**.
- Il proprietario delle tabelle relative alle basi di dati nel cluster sarà l'utente **postgres** il quale, nello stesso tempo, sarà il *super user* della directory cluster.
- Parleremo spesso dell'utente **postgres** e ne descriveremo la fondamentale funzione.

Introduzione a PostgreSQL: file di configurazione

- Il server PostgreSQL è configurabile mediante tre file di configurazione, editabili dall'amministratore:
 - postgresql.conf
 - pg_hba.conf
 - pg_ident.conf
- I file di configurazione sono memorizzati nello stesso percorso del Data Cluster



Introduzione a PostgreSQL: postgresql.conf

Il file **postgresql.conf** contiene informazioni riguardanti:

- **Connessione e autenticazione**
 - Porta di connessione, massimo numero di connessioni, utilizzo di cifratura SSL, ecc ...
- **Utilizzo di risorse**
 - Buffer di memoria dedicato allo scambio di informazioni con i client, buffer di memoria dedicato ad operazioni di ordinamento, ecc ...
- **Logging:**
 - Dove effettuare il logging, quando e che cosa scrivere nei file di log.
- **Statistiche di utilizzo del server**



Introduzione a PostgreSQL: pg_hba.conf

Il file **pg_hba.conf** contiene indicazioni riguardanti da quali macchine è possibile collegarsi al server, a quali database, restringendo eventualmente il diritto di accesso a gruppi di utenti.

Sintassi:

TYPE	DATABASE	USER	IP-ADDRESS/CIDR-mask	METHOD
------	----------	------	----------------------	--------

Esempi:

host	all	all	127.0.0.1	md5
host	test_db	rossi	172.16.*.*	trust

Introduzione a PostgreSQL: pg_ident.conf

Il file **pg_ident.conf** permette di specificare un mapping tra i nomi utente del sistema operativo ed i nomi utente del DBMS.

Qualora si voglia permettere agli utenti di effettuare il login con lo username del sistema operativo.

Sintassi:

MAPNAME	IDENT-USERNAME	PG-USERNAME
Map1	rossi	postgres
Map2	bianchi	marco

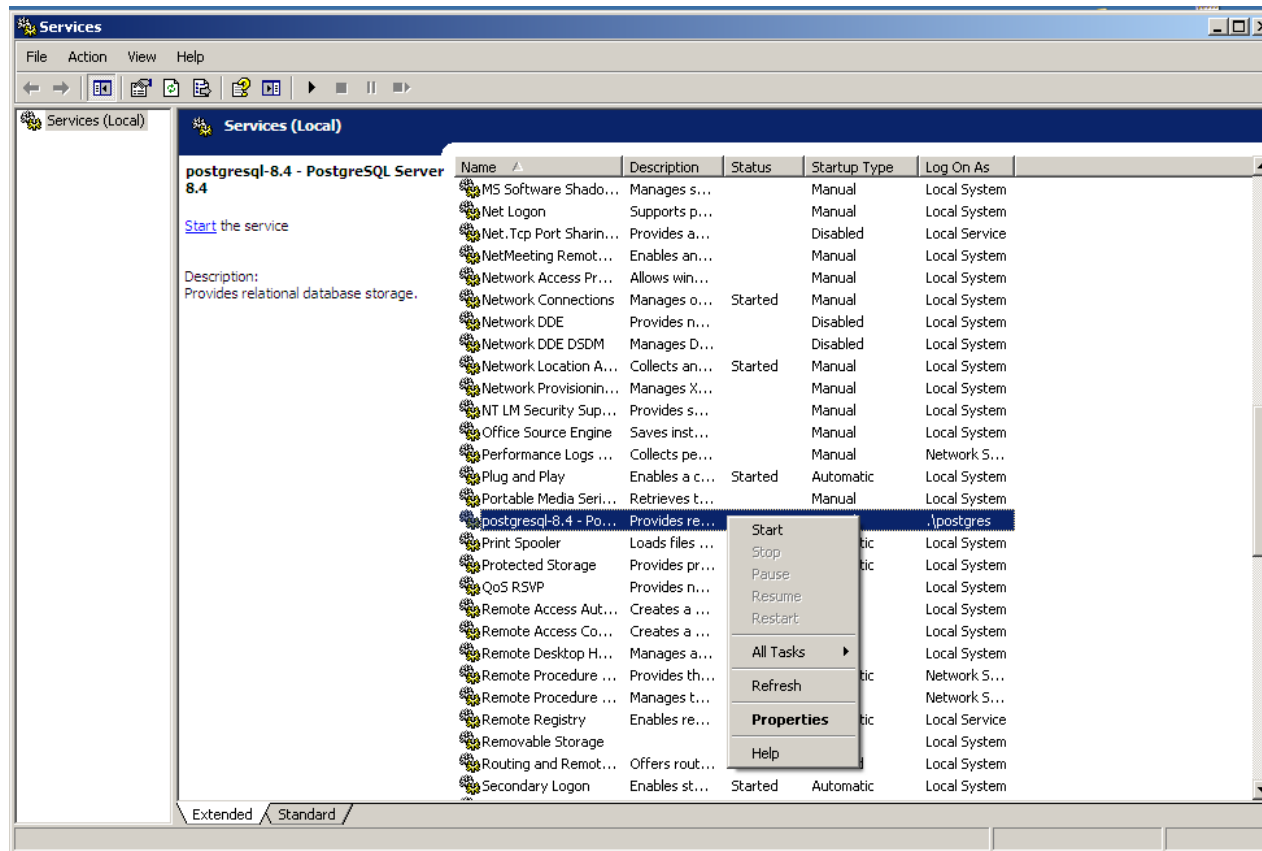
Introduzione a PostgreSQL

Avvio del processo Server

- ▶ In ambiente Windows il server è installato come servizio del sistema operativo.
- ▶ Il processo server PostgreSQL viene avviato automaticamente ad ogni riavvio dell'host server.
- Per disabilitare l'avvio automatico è necessario, accedere al Control Panel, Administrative tools, Services settando Startup Type a Manual.
- Sarà possibile avviare il processo manualmente dal menù programmi.

Introduzione a PostgreSQL

Avvio del processo Server



PostgreSQL – interazione Client/Server

PostgreSQL si basa su un sistema **client/server**:

- **Un processo server** (si occupa della gestione/memorizzazione dei dati)
 - gestisce i file del database
 - accetta connessioni al database da applicazioni client
 - esegue azioni per mezzo dei client
- **Applicazione client** (fornisce un'interfaccia di consultazione e/o inserimento)
 - applicazione grafica
 - server web (accesso al database per visualizzare pagine web)
 - strumento testuale
- Il client e il server possono essere su host diversi
- Il server PostgreSQL può gestire connessioni multiple concorrenti dai client

PostgreSQL – interazione Client/Server

- Per ogni connessione client viene creato un processo server
- Il numero massimo di connessioni è configurabile
- Maggiore è il numero di connessioni massime, maggiore sarà il carico computazionale dell'host server
- Connessioni possono essere effettuate da client dedicati o da applicazioni utente

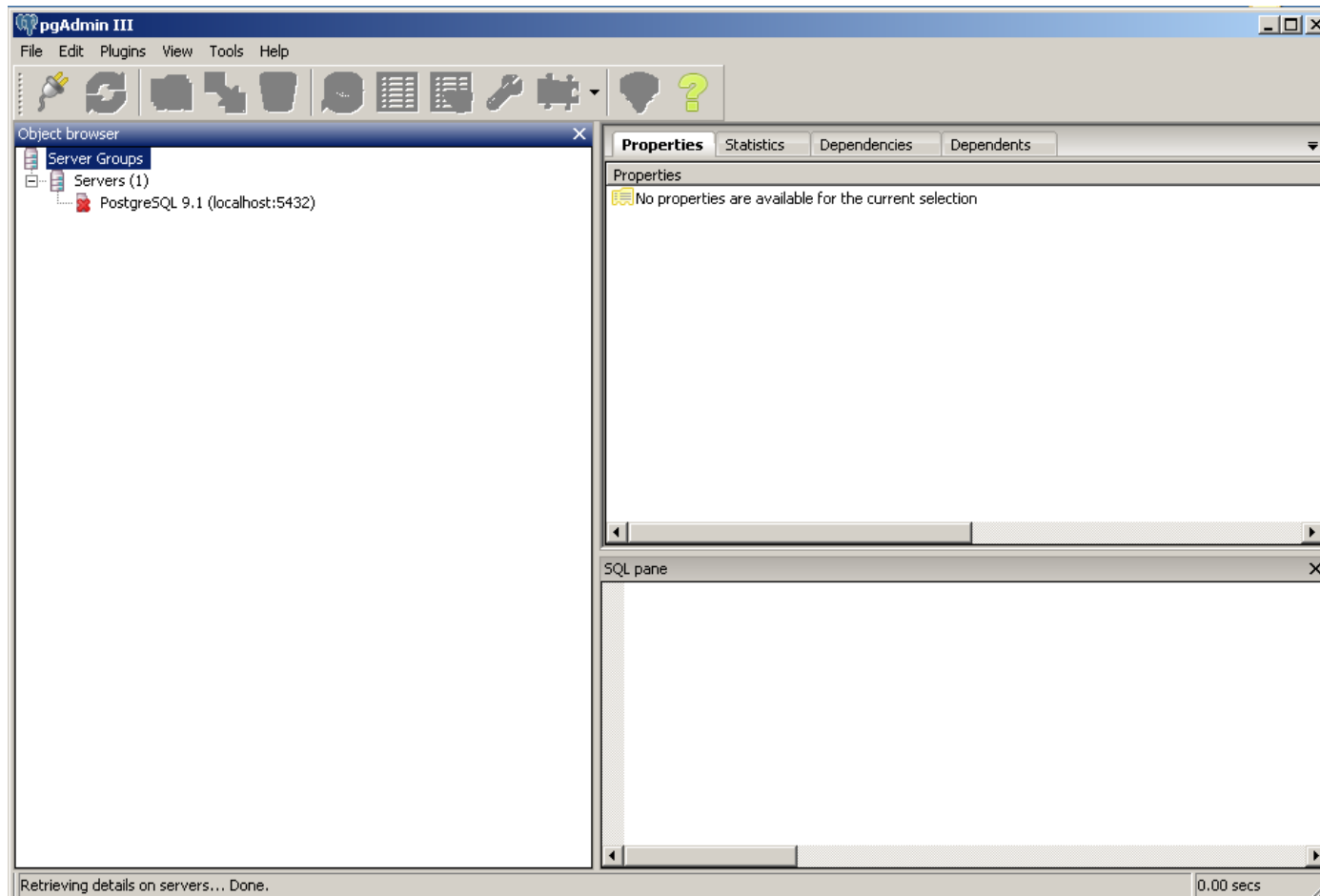
PostgreSQL – PgAdmin III

PgAdmin è un' interfaccia grafica che consente di amministrare in modo semplificato DataBase di PostgreSQL

- ▶ L'applicazione è utilizzata sia agli amministratori del database, sia agli utenti. Gestisce i permessi prelevandoli dal database PostgreSQL.
- ▶ **PgAdmin** permette di creare un database da zero, creare le tabelle ed eseguire operazioni di ottimizzazione sulle stesse. Presenta un feedback sulla creazione delle tabelle per evitare eventuali errori. Sono previste delle funzionalità per l'inserimento dei dati (*popolazione del database*), per le query, per il backup dei dati, ecc..
- ▶ L'interfaccia grafica permette la gestione degli utenti: l'inserimento di un nuovo utente, la modifica della relativa password e la gestione dei permessi che l'utente ha sul database, utilizzando lo standard SQL.

PostgreSQL – PgAdmin III

Finestra principale:

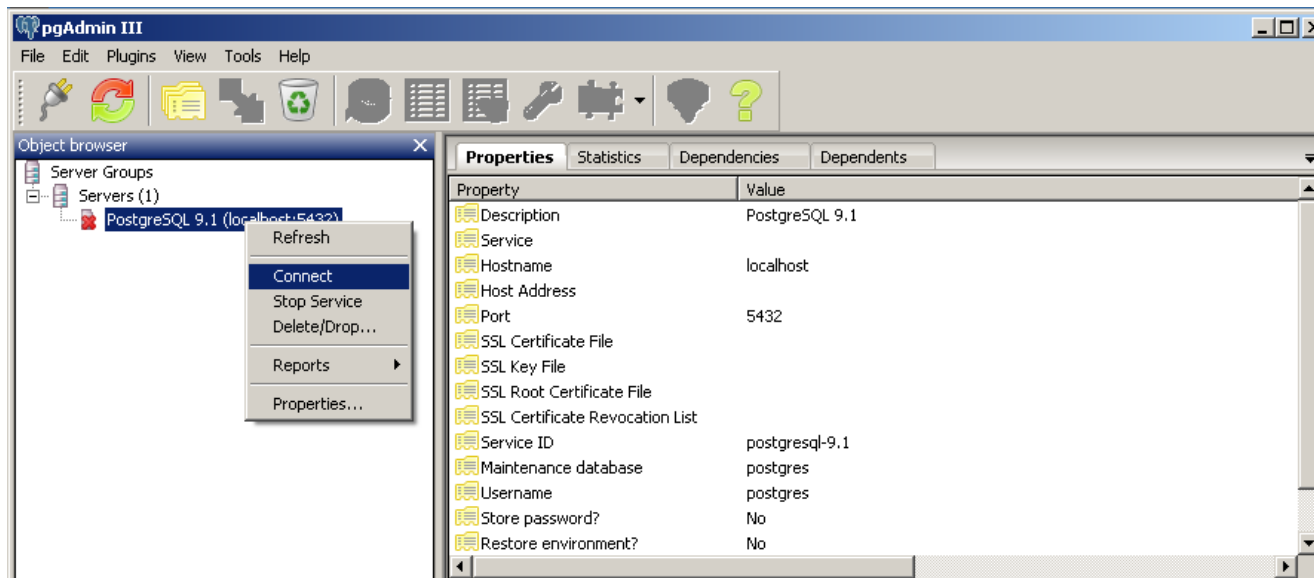


PostgreSQL – PgAdmin III

Cliccando con il tasto destro del mouse sulla **connessione già configurata**, PostgreSQL Database Server 9.1 (localhost:5432), e selezionando *Connect* ci verrà Chiesta la password di postgres (l'amministratore del RDBMS).

Possiamo anche configurare una nuova connessione ad un database remoto.

Una volta eseguito il **login** con l'utente amministratore troveremo già presente un database di test, creato al momento dell'installazione, il cui nome è postgres.



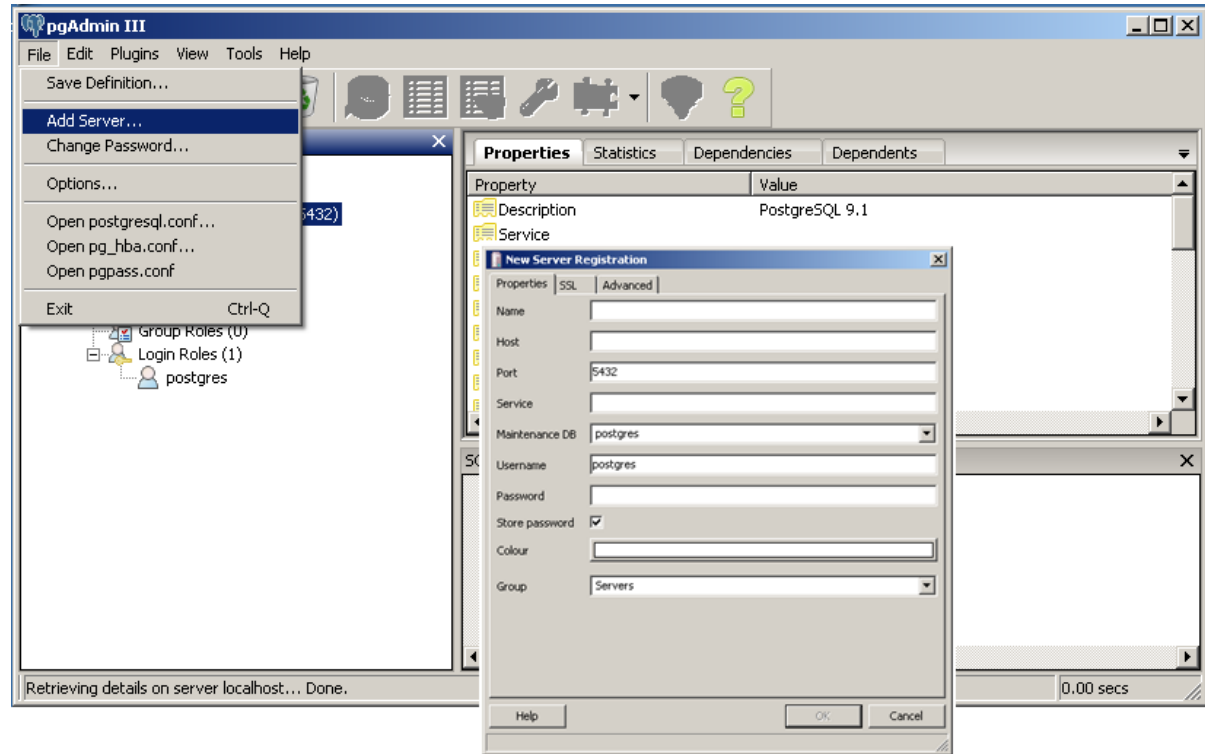
PostgreSQL – PgAdmin III

Aggiunta di un server:

Name: contiene la descrizione che verrà visualizzata nel Object Browser.

Host: contiene l'indirizzo IP della macchina da contattare o il suo nome completo.

Maintenance DB: è utilizzato per specificare il database che pgAdmin contatta durante la connessione.



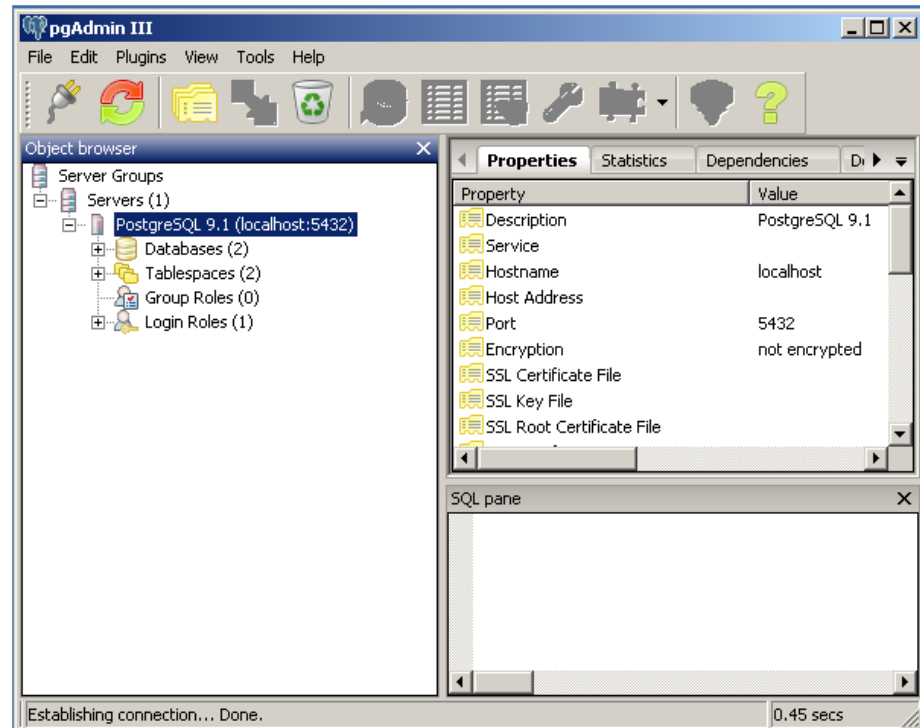
PostgreSQL – PgAdmin III

Nella parte sinistra della finestra viene visualizzato un albero con tutti gli oggetti presenti sul server.

Elenco dei database presenti (ogni Server ne può contenere molti), un elenco di tablespaces ed un elenco di Group e Login.

I **Tablespace** servono per gestire la memorizzazione fisica dei dati; ad esempio grandi moli di dati possono essere partizionate in più tablespaces per aumentare l'efficienza.

La gestione dei **gruppi e dei login** permette di regolare i livelli di accesso ad ogni risorsa, in presenza di molti utilizzatori.



PostgreSQL – PgAdmin III

The screenshot displays the PgAdmin III interface. On the left, the Object browser shows a tree view of the PostgreSQL 9.1 server. The 'cinema' database is selected under 'Databases (2)'. The Properties window on the right shows the following details:

Property	Value
Name	cinema
OID	16393
Owner	postgres
ACL	
Tablespace	pg_default
Default tablespace	pg_default
Encoding	UTF8
Collation	English_United States.1252
Character type	English_United States.1252
Default schema	public
Default table ACL	
Default sequence ACL	
Default function ACL	
Allow connections?	Yes
Connected?	Yes

The SQL pane at the bottom contains the following SQL script:

```
-- Database: cinema
-- DROP DATABASE cinema;

CREATE DATABASE cinema
  WITH OWNER = postgres
  ENCODING = 'UTF8'
  TABLESPACE = pg_default
  LC_COLLATE = 'English_United States.1252'
  LC_CTYPE = 'English_United States.1252'
  CONNECTION LIMIT = -1;
```

A context menu is open over the Properties window, showing options such as 'Refresh', 'New Object', 'Delete/Drop...', 'Drop cascaded...', 'CREATE Script', 'Reports', 'Backup...', 'Restore...', 'Grant Wizard...', and 'Properties...'. The 'New Object' submenu is also visible, listing options like 'New Schema...', 'New Collation...', 'New Domain...', 'New Function...', 'New Sequence...', 'New Table...', 'New FTS Configuration...', 'New FTS Dictionary...', 'New FTS Parser...', 'New FTS Template...', 'New Trigger Function...', and 'New View...'.

PostgreSQL – PgAdmin III

Extensions:

Si tratta dei linguaggi con cui è possibile creare le Stored procedures e di cui abbiamo parlato in precedenza. Di default e' PL/pgSQL.

Schema:

questi sono gli schemi in cui un db puo essere suddiviso. Corrispondono alle cartelle di un disco. Il db contiene lo schema di default, che si chiama public. Gli schemi sono una risposta ad una limitazione di PostgreSQL: poichè non è possibile effettuare query che coinvolgano più database (ad esempio un'istruzione Select che esegua una Join tra tabelle di database differenti) è previsto che un database sia suddivisibile in schemi.

- Ogni schema rappresenta una diversa entità all'interno del database (ha le proprie tabelle, le proprie stored procedures e così via) ma si possono eseguire query che coinvolgano tabelle di schemi differenti.
- Ogni database contiene almeno uno schema di default denominato *public*.

PostgreSQL – PgAdmin III

Collations:

è una nuova funzionalità di PostgreSQL 9.1 che permette di specificare il tipo di ordinamento per colonne o per operazione.

Domains:

PostgreSQL supporta i vincoli (constraints) di colonna e di tabella. Quando un vincolo è utilizzabile da più tabelle, anzichè definirlo per la singola tabella, possiamo assegnarlo come domain in modo che diventi universale.

Se ad esempio dovessimo utilizzare in più tabelle un campo atto a contenere codici di avviamento postale (Cap), anzichè effettuare un controllo di validità sul Cap assegnandolo alla singola tabella, posso creare un tipo di dato Cap e contestualmente assegnarvi il controllo di validità.



PostgreSQL – PgAdmin III

Functions, Operators, Operator Classes:

Sono funzionalità aggiuntive con cui è possibile estendere il RDBMS. È possibile aggiungere funzioni di vario tipo e nuovi operatori.

Aggregates:

In questo "contenitore" finiscono le funzioni di aggregazione aggiuntive che l'utente può creare per estendere quelle già presenti nel database, esempi di funzioni di aggregazione predefinite sono **SUM()**, **AVG()**, **COUNT()** e così via.

Conversions:

Funzioni di conversione personalizzate tra character sets differenti.



PostgreSQL – PgAdmin III

Procedures e Triggers:

Le stored procedures sono moduli o routine che incapsulano codice in modo da poterlo riutilizzare, i Triggers sono stored procedures particolari che vengono innescate automaticamente al verificarsi di determinati eventi.

Sequences:

PostgreSQL non prevede il tipo di dato "autoincrement" come MySQL, in origine quindi non era possibile definire un campo il cui valore si incrementava automaticamente all'aggiunta di un nuovo record. L'unico modo era definire una sequenza, ovvero una sorta di generatore automatico che doveva essere richiamato quando si inseriva un valore nel campo da incrementare. A partire dalla release 7.3 tuttavia è stato introdotto il tipo di dato **Serial** che pur non essendo un vero "autoincrement" genera automaticamente la sequenza e la assegna come valore di default del campo.

PostgreSQL – PgAdmin III

Tables:

Sono l'unico aspetto che dovrebbe essere già familiare agli utenti di MySQL, tuttavia ogni database nasce già con delle tabelle predefinite che non devono essere toccate.

Types:

Sono i tipi di dati che vanno ad integrare o estendere quelli predefiniti, possono essere creati in svariati linguaggi, solitamente in C. Si tratta di un modo diverso di definire nuovi tipi di dati rispetto all'utilizzo dei domains.

Views:

Le Viste sono una sorta di query memorizzate in tabelle virtuali, per questo vengono definite anche "pseudo-tabelle". Si rivelano utili in diverse situazioni, in modo particolare quando si vuole concedere agli utenti l'accesso a determinati dati e solo a quelli.



PostgreSQL – PgAdmin III

La finestra di comandi SQL: Query Tool

Una volta che avete selezionato il database che vi interessa cliccate sullo strumento SQL. Si aprirà la finestra con l' editor SQL.

Nella parte sinistra della finestra c' è l'editor vero e proprio, nella parte destra c' è una specie di notepad, utile per copiare ed incollare testi.

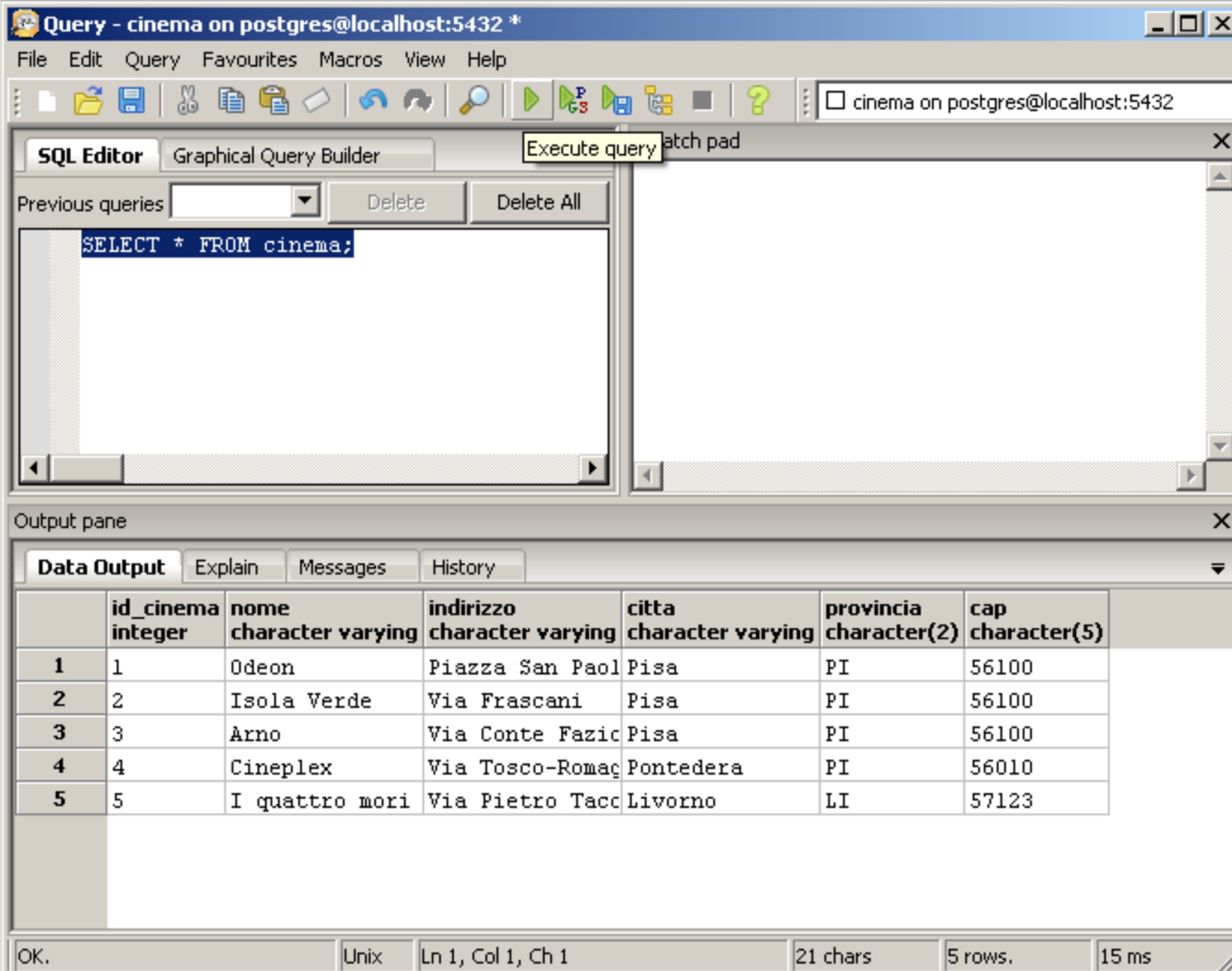
Nella parte sottostante invece vengono visualizzati i risultati delle operazioni ed altre cose.

L'utilizzo della finestra è il seguente:

Si scrive la query SQL nella finestra di sinistra, si preme il tasto Execute (icona play) e si legge il risultato nella parte sottostante.



PostgreSQL – PgAdmin III



The screenshot shows the PgAdmin III interface. The title bar reads "Query - cinema on postgres@localhost:5432 *". The menu bar includes File, Edit, Query, Favourites, Macros, View, and Help. The toolbar contains various icons for file operations and query execution. The "SQL Editor" tab is active, showing the query: `SELECT * FROM cinema;`. The "Output pane" is visible below, showing the results of the query in a table format. The table has 7 columns: `id_cinema` (integer), `nome` (character varying), `indirizzo` (character varying), `citta` (character varying), `provincia` (character(2)), and `cap` (character(5)). The results are as follows:

	<code>id_cinema</code> integer	<code>nome</code> character varying	<code>indirizzo</code> character varying	<code>citta</code> character varying	<code>provincia</code> character(2)	<code>cap</code> character(5)
1	1	Odeon	Piazza San Paolo	Pisa	PI	56100
2	2	Isola Verde	Via Frascani	Pisa	PI	56100
3	3	Arno	Via Conte Fazio	Pisa	PI	56100
4	4	Cineplex	Via Tosco-Romagnolo	Pontedera	PI	56010
5	5	I quattro mori	Via Pietro Tacchini	Livorno	LI	57123

The status bar at the bottom indicates: OK, Unix, Ln 1, Col 1, Ch 1, 21 chars, 5 rows., 15 ms.

PostgreSQL – PgAdmin III

La sintassi SQL deve essere esatta!

Ricordatevi di:

- non inserire spazi all'interno dei comandi
- di non confondere zero con la lettera o
- di non confondere l'apicetto singolo con le doppie virgolette
- di non confondere le virgole, punti e punti e virgola

Per fortuna, SQL **non è case sensitive**, vale a dire che non distingue maiuscole e minuscole (potete scrivere indifferentemente `SELECT`, `select` o `Select`).

