Esercitazioni di Basi di Dati

A.A. 2008-'09 Dispense del corso Utilizzo base di pgAdmin III

Lorenzo Sarti – sarti @dii.unisi.it

PgAdmin III

PgAdmin III è un sistema di progettazione e gestione grafica di database Postgres. E' gratuito e sviluppato in C++ utilizzando la libreria wxWidgets. La connessione al server Postgres è realizzata mediante la libreria di connessione libpq, distribuita insieme al DBMS.

pgAdmin finestra principale



La finestra principale di pgAdmin visualizza la struttura dei database gestiti dal server Postgres. Permette di creare, cancellare o modificare oggetti, se i privilegi dell'utente attualmente connesso lo permettono.

La parte sinistra della finestra mostra l'albero dei server Postgres raggiungibili e gli oggetti che i server contengono.

La parte superiore destra mostra i dettagli dell'oggetto corrente (selezionato). La parte inferiore destra visualizza il/i comando/i SQL che hanno generato l'oggetto attivo. I comandi SQL visualizzati sono realizzati attraverso un meccanismo di "Reverse Engineering".

La barra di stato mostra informazioni riguardanti l'ultima operazione effettuata, insieme al tempo necessario per compierla.

Aggiunta di un server

I New Server Re	egistration 🛛 🔀
Properties	
Name	
Host	
Port	5432 SSL 🔽
Maintenance DB	postgres 🗸
Username	postgres
Password	
Store password	
Restore env?	
DB restriction	
Service	
Connect now	
Help	OK Cancel
Please specify address	5.

Il campo nome contiene la descrizione che verrà visualizzata nell'albero della finestra principale di pgAdmin. Il campo host deve contenere l'indirizzo IP della macchina da contattare o il suo nome completo, se all'indirizzo IP è associato un "domain name". Il campo Maintenance DB è utilizzato per specificare il database che pgAdmin contatta durante la connessione. Il Maintenance DB può chiamarsi "postgres" o "template1", a seconda della versione di Postgres installata, e contiene la descrizione degli oggetti gestiti dal server.

Connettersi ad un server

Per aprire una connessione ad un server esistente, selezionare il server ed effettuare doppio click sul nome. La connessione verrà effettuata, eventualmente richiedendo la password di autenticazione.

Creazione di un database

🧻 New Da	tabase	
Properties	Variables	Privileges SQL
Name		DBexample
OID		
Owner		~
Encoding		SQL_ASCII 🔽
Template		×
Tablespace		~
Comment		
Help		OK Cancel

Nell'albero dei server selezionare la voce "Databases" e, facendo click con il tasto destro, scegliere la voce "New Database" dal menù contestuale. Apparirà la finestra "New Database".

Il campo Name contiene il nome del nuovo DB, la combobox Owner permette di specificare il proprietario del DB, ovvero l'utente che ha diritti di amministrazione. Se non specificato, il proprietario sarà l'utente attualmente autenticato.

La combobox Encoding permette di definire il tipo di codifica da utilizzare per le informazioni alfanumeriche.

La combobox Template indica a partire da quale modello di DB verrà creato il nuovo database. Attualmente Postgres crea un nuovo DB effettuando la copia di un modello. Il template predefinito è denominato "template1".

La combobox Tablespace indica qual è la locazione in cui il database verrà archiviato. Postgres mette a disposizione due Tablespace predefiniti, pg_global e pg_default, che corrispondono rispettivamente ai percorsi C:\Programmi\PostgreSQL\%Version%\data\pg_tblspc e

C:\Programmi\PostgreSQL\%Version%\data\global in ambiente Windows. La creazione di nuovi Tablespace da parte dell'utente è ammessa solo su filesystem che consentono la creazione di link simbolici (Linux, Solaris, MacOSX).

Infine la textbox Comment può essere utilizzata per inserire dei commenti che descrivano il contenuto del DB.

Gli altri Tab (Variables, Privileges) verranno descritti più avanti.

Il Tab SQL, invece, visualizza il comando SQL che equivale alle scelte operate dall'utente. Nel caso della scelta in figura l'istruzione SQL corrispondente è la seguente: CREATE TABLE "DBExample";

Creazione di una tabella

🖪 New Table		×
Properties Columns	Constraints Privileges SQL	
Name	example	
OID		
Owner	postgres	~
Tablespace	pg_default	~
Has OIDs		
Inherits from table	s	וה
Remove		
Add		•
Comment		
Use replication		~
Help		3

Per creare una tabella all'interno di un database esistente è necessario selezionare dall'albero di pgAdmin la voce "Table" sottostante al nome del DB al quale la tabella deve essere aggiunta. Cliccando con il tasto destro del mouse apparirà un menù contestuale, dal quale è necessario selezionare "New Table".

La finestra "New Table" è costituita da cinque Tab.

Il Tab Properties consente di specificare il nome della tabella, il proprietario dei diritti di amministrazione e il Tablespace.

La checkbox "Has OIDs" permette di attivare o disattivare la creazione automatica dell'Object Identification, strumento utilizzato dai DBMS object oriented per identificare univocamente i record della tabella.

Postgres permette di sfruttare l'ereditarietà. E' possibile quindi definire una tabella come "figlia" di un'altra tabella, in modo che ne erediti la definizione. L'eventuale tabella padre va indicata utilizzando il pulsante "Add".

Una volta definito il nome della tabella è possibile passare alla definizione delle sue colonne (dei suoi attributi), spostandosi sul Tab Columns.

Il Tab Columns riassume tutti gli attributi della tabella, per aggiungere un attributo si utilizza il tasto Add, il quale apre una finestra di dialogo denominata New Column.

📕 New Column		×
Properties		
Name		
Data type		*
Length	abstime aclitem	^
Default value	bit bool	
Sequence	bpchar bytea	=
Not NULL	char	
Statistics	cidr	
Comment	circle date float4 float8 inet	
	int2 int4	
	int8	
	interval macaddr	
Use replication	money	
Help	oid path polygon	

La finestra New Column consente di specificare il nome dell'attributo, il suo tipo, la sua eventuale lunghezza e precisione, il suo valore di default, e se sono ammessi valori nulli oppure no.

Postgres consente l'utilizzo di numerosi tipi di dato, sia semplici che sotto forma di array. Per alcuni tipi di dati (int, float) è possibile specificare il numero di byte utilizzati per la rappresentazione (2, 4, 6, 8). Alcuni tipi di dato non appartengono allo standard SQL. Tra questi, il più interessante è il tipo di dato serial, che consente di creare indici autoincrementanti, particolarmente utili per la definizione di codici univoci da utilizzare come chiavi primarie.

Terminata la specificazione delle colonne è possibile passare alla definizione dei vincoli interrelazionali e intrarelazionali, mediante il Tab Constraints.

Il Tab Constraints permette di specificare il vincolo di chiave primaria, di chiave straniera, il vincolo unique e check.

Per la creazione della chiave primaria è necessario selezionare dalla combobox in basso a sinistra la voce "Primary key" e cliccare sul tasto "Add". Apparirà una nuova finestra mediante la quale andranno specificati gli attributi appartenenti alla chiave. E' importante notare che Postgres considera anche i vincoli come oggetti a se stanti, quindi ogni nuovo vincolo dovrà avere un nome associato.

Per la creazione di una chiave straniera è necessario specificare la tabella riferita e gli attributi locali ed esterni che sono oggetto del vincolo.

La verifica di qualsiasi vincolo può essere effettuata da Postgres immediatamente (all'atto dell'inserimento di uno o più record) oppure può avvenire in modo differito ("Deferrable / Deferred"). L'utilizzo della verifica differita può essere utilizzato durante l'esecuzione di transazioni. Una transazione è un blocco di istruzioni SQL che vengono effettuate in sequenza ma atomicamente (o tutte le istruzioni vanno a buon fine oppure è necessario, generalmente, ripristinare la situazione iniziale in cui si trovava il DB). Se durante la creazione di un vincolo è specificato che il controllo può essere differito ("Deferrable") o è obbligatoriamente differito ("Deferred"), la verifica di integrità viene effettuata solo al termine della transazione.

🖬 New Table 🛛 🔀
Properties Columns Constraints Privileges SQL
Constraint name Definition
Primary Key Add Remove
Primary Key
Unique OK Cancel
Check

Come al solito il Tab SQL indica l'equivalente SQL delle operazioni effettuate dall'utente mediante l'interfaccia grafica.

Visualizzazione contenuto tabella e inserimento dati

Postgres permette la visualizzazione e l'inserimento di dati attraverso una visualizzazione stile foglio di calcolo.

🔲 pgA dmin	III Edit Data -	PostgreSQL	Database Serv	/er 8.1 (loca	lhost:5432	2) - pos)[>	<
🔳 / 🍠 🐢	🖿 🗑 🏹	7 8						
	codice [PK] int4	nome varchar	nazione varchar					^
1	1	Firenze	Italy					
2	2	Siena	Italy				-	
3	3	Roma	Italy					
4	4	Milano	Italy					
5	5	Torino	Italy					
6	6	Bologna	Italy					
7	7	Napoli	Italy					
*								
								~
<							>	
7 rows.								

L'inserimento può essere effettuato mediante la riga contrassegnata dal simbolo asterisco (*). In questa modalità è anche possibile modificare una tupla o eliminarla.

Per inserire una singola tupla in SQL è disponibile il comando INSERT INTO. Postgres mette a disposizione anche un comando non standard, COPY, che consente di inserire una serie di tuple lette da un file di testo.

Supponendo di dover copiare un elenco di città in una tabella denominata citta, è necessario:

1. predisporre un file (altrecitta.txt, per esempio) con il seguente contenuto:

10, Parigi, France 11, Londra, England 12, Mosca, Russia 13, Berlino, Germany

2. eseguire la seguente istruzione SQL: COPY citta from 'C:\\Documents and Settings\\ ... \\altrecitta.txt' WITH DELIMITER ','

Query tool

Il query tool permette di eseguire comandi SQL. Query tool è attivato dalla finestra principale di pgAdmin utilizzando il tasto contrassegnato dalla matita.

La parte superiore della finestra query tool consente di inserire il comando SQL da eseguire. Le query possono essere lette da file o salvate su file. Per eseguire una istruzione SQL è possibile selezionare il comando Execute dal Query menù, oppure utilizzare il tasto Play sulla barra di comando.

L'eventuale risultato dell'esecuzione dell'istruzione viene visualizzato nella parte bassa della finestra, nel Tab Data Output. Il risultato può essere salvato su file di testo, utilizzando il tasto Play&Save. Verrà richiesta la codifica da utilizzare, il tipo di separatore di colonna (, ; TAB, ecc...), ed il nome del file di output.

🛯 pg	Admi	n III Qu	ery - postgres	on localhos	t:5432 *			
<u>F</u> ile <u>E</u>	idit g	uery <u>H</u>	elp					
6	I N		🖉 🔊 🗖	🖉 🔎 postg	res on localhost:5	5432 🔽 🕨 🕨	🍋 🔚 🔳	2
SEL	ECT	* FROM	citta, treni	WHERE citt	a.nome='Fire	nze'		
<								>
Data C	Dutout	Explain	Messages His	tory				
P	codic	e (int4)	nome (var	pazione (v	codice (ipt4)	stazionena	stazionear	orapartenz
1	1	e (incry	Firenze	Italy	1	1	2	11:00:00
2	1		Firenze	Italy	2	2	1	12:00:00
<								>
ΟК.				Lr	1 Col 53	2 rows.	20+50) ms

Interessante è l'utilizzo della funzione Explain, attivabile mediante il tasto F7, che descrive come la query è stata ottimizzata ed eseguita graficamente, specificando i tempi necessari per lo svolgimento della varie operazioni.

Image: PgAdmin III Query - postgres on localhost:5432 ★	
<u>File E</u> dit Query <u>H</u> elp	
📂 🔚 🐰 🗎 🖷 🧼 🧔 🐢 🔎 postgres on localhost:5432 🛛 💌 🕨 🍖 🔳 💡	2
SELECT * FROM citta, treni WHERE citta.nome='Firenze'	
	>
Data Output Explain Messages History	
treni Nested Loop	
citta Materialize	~
OK. Ln 1 Col 53 5 rows. 10 ms	

Creazione di un dominio

La definizione di un dominio è simile alla definizione di un tipo di dati astratto. Un dominio non può però essere definito come un tipo strutturato.

Per definire un nuovo dominio è possibile cliccare con il tasto destro sulla voce Domains dell'albero di pgAdmin e scegliere la voce "New Domain". I domini appartengono ad una base di dati specifica, quindi sarà necessario definirli all'interno della base di dati in cui dovranno essere utilizzati. Apparirà la seguente finestra:

🏠 New Domain	N 1997
Properties SQL	
Name	Voto
OID	
Owner	postgres 🛛 👻
Base type	int2
Default	18
Check	VALUE>=18 AND VALUE<=30
Length	Precision
Not NULL	
Comment	
Use replication	×
Help	OK Cancel

Come tutti gli oggetti Postgres sarà necessario specificare un nome, all'interno della casella di testo denominata "Name", un proprietario, il tipo di base dal quale si definisce il dominio, un valore di Deault (obbligatorio) ed una eventuale condizione booleana che i valori appartenenti al dominio devono soddisfare. Per riferirsi al valore assunto da un elemento appartenente al dominio, nell'espressione booleana deve essere utilizzata la parola chiave VALUE.

Vacuum

Quando una tupla viene modificata, in realtà PostgreSQL mantiene una copia della vecchia tupla nel database e ne crea una nuova. La vecchia tupla è contrassegnata come invalida. La stessa cosa avviene durante la cancellazione di una tupla o di un altro elemento del database. La cancellazione è solo virtuale, in realtà una copia dell'oggetto rimosso viene mantenuta nel database. Questa ridondanza può essere utile per effettuare procedure di ripristino di dati. E' comunque evidente che con il passare del tempo PostgreSQL può diventare inefficiente. Per liberare definitivamente lo spazio occupato dagli oggetti contrassegnati invalidi è possibile utilizzare lo strumento "Vacuum" (contrassegnato da un aspirapolvere o da una chiave inglese nella barra degli strumenti di pgAdmin, a seconda della versione). Attivando il Vacuum apparirà la seguente finestra:

🥕 Maintain Database Studenti 💦 🔲 🔀
Options Messages
Maintenance operation O VACUUM ANALYZE REINDEX
VACUUM options FULL FREEZE ANALYZE Verbose messages
Help OK Cancel

Il Vacuum è uno degli strumenti di manutenzione messi a disposizione da pgAdmin. Ovviamente il Vacuum viene attivato sull'oggetto selezionato nel frame di sinistra dell'applicazione. Se si vuole effettuare la pulizia dell'intero database sarà necessario selezionare l'oggetto database.

L'attivazione dell'opzione "Analyze" permette la pulizia dello spazio inutilizzato e la raccolta di statistiche che l'ottimizzatore di query utilizzerà per svolgere i suoi compiti. L'esecuzione di Vacuum con l'opzione Analyze non richiede l'accesso esclusivo al database.

L'attivazione dell'opzione "Full", invece, non solo elimina le tuple inutilizzate, ma compatta le altre in modo da ottimizzare l'accesso. Questa opzione acquisisce l'accesso esclusivo al database, durante l'esecuzione.

L'opzione "Freeze" riguarda la manutenzione di database che sono oggetto di continue transazioni, la sua spiegazione non è necessaria agli obiettivi del corso.

Durante l'esecuzione di "Vacuum", il DBMS visualizza una finestra di logging con l'avanzamento della procedura e le statistiche raccolte.

Backup

Per effettuare il backup di un database o parte di esso è necessario selezionare l'oggetto da salvare nell'albero di pgAdmin, utilizzando il pulsante destro, e scegliere la voce "Backup". Apparirà la seguente finestra:

📑 Backup Table Esami	
Options Messages	
Filename	
Format	
⊙ COMPRESS ○ TAR	○ PLAIN
 Blobs With OIDs Insert commands Disable \$ quoting 	PLAIN options Only data Only schema No owner Create DB Drop DB Disable Trigger
Verbose messages	
	OK <u>C</u> ancel

E' necessario selezionare il nome del file sul quale effettuare il dump del database. L'output può essere effettuato in tre modalità: file compresso (Compress), file di archivio (Tar) o puro testo (PLAIN). L'opzione "Plain" è particolarmente utile a livello didattico, il salvataggio viene infatti generato sotto forma di script SQL. E' possibile scegliere di salvare solo i dati, solo lo schema, effettuare un salvataggio che non memorizzi i proprietari degli oggetti, realizzare uno script che clona esattamente il database, o uno script che rimuove fisicamente il database.

Restore

Il ripristino del database avviene con una procedura simile a quella di backup. Dal menù contestuale che appare selezionando con il tasto destro il database o l'oggetto da ripristinare è necessario selezionare la voce "Restore". Apparirà la seguente finestra:

💽 Restor	re Database Studenti 📃 🗖 🔀
Options (Contents Messages
Filename	I
	Only data
	Only schema
	Single object
	No owner
	Disable Trigger
	Verbose messages
<u>H</u> elp	<u>V</u> iew <u>O</u> K <u>C</u> ancel

Per il ripristino è necessario indicare il file sul quale è stato effettuato il dump, indicare se si intende

ripristinare solo i dati o anche lo schema, e cliccare su OK. Per ripristinare dump realizzati con l'opzione "Plain" è invece necessario eseguire lo script dal Query editor.