

SQL

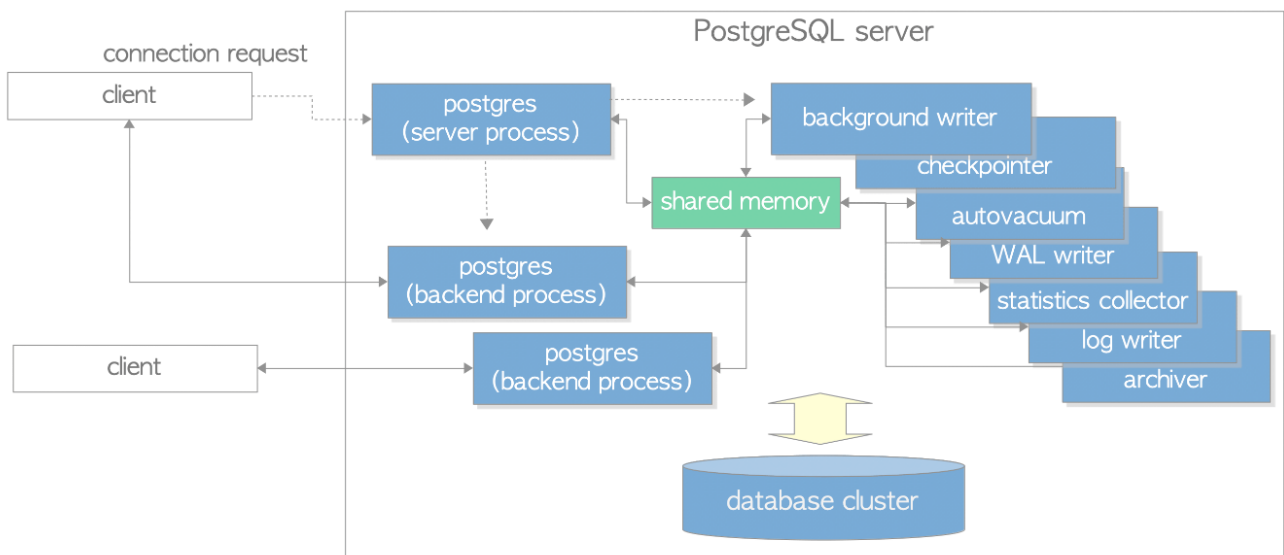
#sql #pratica #comandi-sql

Linguaggio che contiene sia DML(che lavora solo su istanza) sia DDL.

Per `psql`, parlando del comando, esistono diversi *worker* all'interno della nostra macchina linux che vengono avviati all'istanza. Il seguente comando li mostra in command line.

```
ps aux | grep mysql
```

- Il nostro server si prende la libertà di memorizzare le pagine all'interno della RAM, ma si occupa anche di possibili fallimenti, il *logging collector* è uno dei processi;
- *stats collector* ogni tanto lancia (possiamo farlo manualmente) per acquisire info sul database;
- *autovacuum launcher*, nelle mie tabelle ogni tanto faccio eliminazioni che sono logiche (alcune *n*-uple diventano invalide), aiuta a recuperare spazio liberato compattando le tabelle;
- *logical replication launcher* per replicare (come altro server) i contenuti del server corrente (master & slave);
- il *background writer* scrive pagine nella shared memory lentamente verso la memoria persistente;
- *WAL writer* che trasferisce dati WAL su memoria persistente;
- *archiver* che archivia il log eseguito.



Interfaccia grafica

Using `pgAdmin`.

Nell'interfaccia grafica possiamo fare JOIN tra *schemi* (collezione di tabelle), possiamo vedere gli utenti con accesso al DB, possiamo vedere le relazioni. Le operazioni sono molteplici ma equivalgono alle stesse operazioni possibili tramite linea di comando (useremo soltanto da linea di comando).

Note

Personalmente non uso GUI perche' faccio prima da Visual Studio

CREATE TABLE

Per creare porzioni di schema usiamo l'istruzione `CREATE TABLE`:

- definisce uno schema di relazione e ne crea un'istanza vuota;
- specifica attributi, domini e vincoli;

Possiamo usare il risultato di una istruzione come argomento del `CREATE`.

```
CREATE TABLE impiegato (  
    matricola CHAR(6) PRIMARY KEY,  
    nome CHAR(20) NOT NULL,  
    cognome CHAR(20) NOT NULL,  
    dipart CHAR(15),  
    -- il valore default sarà 0  
    stipendio NUMERIC(9) DEFAULT 0,  
    FOREIGN KEY(dipart) REFERENCES  
        dipartimento(NomeDip),  
    -- in questa tabella i valori (cognome,nome) insieme formano  
    -- una chiave (non primaria)  
    UNIQUE(cognome, nome)  
);
```

DOMINI ELEMENTARI

- *carattere*, singoli caratteri o stringhe;
- *numerici* esatti e approssimati;
- *data, ora, intervalli di tempo*;
- *boolean*, scritto per esteso
- *BLOB*(binary long object), *CLOB*(character long object)

Definiamo tipo di dato semplice con `CREATE DOMAIN`.

Ci permette di portarci dietro i vincoli ogni qual'ora ci serve scrivere lo stesso dato in più tabelle.

```
CREATE DOMAIN voto  
    AS SMALLINT DEFAULT NULL  
    CHECK(value >= 18 AND value <= 30)
```

VINCOLI INTRARELAZIONALI

- *NOT NULL*;
- *UNIQUE* per definire chiavi;

- *PRIMARY KEY* la chiave primaria (una soltanto, implica NOT NULL);
- *CHECK* per vincolo di n -upla.

VINCOLI INTERRELAZIONALI

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Auto

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Creiamo chiave esterna sui due attributi (Prov, Numero):

```
CREATE TABLE infrazioni (
  codice CHAR(6) NOT NULL PRIMARY KEY,
  data DATE NOT NULL,
  -- foreign key su singolo attributo
  vigile INTEGER NOT NULL
    REFERENCES vigili(matricola),
  provincia CHAR(2),
  numero CHAR(6),
  -- foreign key su molteplici attributi
  FOREIGN KEY(provincia, numero)
    REFERENCES auto(provincia, numero)
)
```

- *CHECK*;
- *REFERENCES* e *FOREIGN KEY* per definire vincoli d'integrità referenziale;
 - per singoli attributi
 - su più attributi
- è possibile definire politiche di reazione alla violazione

MODIFICHE DEGLI SCHEMI

- ALTER DOMAIN

- ALTER TABLE
- DROP DOMAIN
- DROP TABLE
- ...

Ci sono casi in cui il DBMS si rifiuta di cancellare le tabelle nel caso in cui siano referenziate da altre tabelle da diverse, usiamo in quel caso la parola chiave **CASCADE**.

es.: non posso cancellare **vigili** siccome nella tabella delle **infrazioni** c'è un vincolo di chiave esterna che ne fa indice. Usiamo in questo caso **CASCADE** e il vincolo di chiave esterna sparisce.

lezione: 2022-10-11