

PostgreSQL
the world's most advanced open source database

The world's most advanced class on PostgreSQL... o quasi!

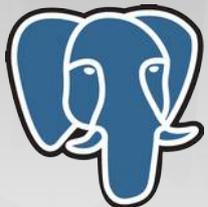
4 Marzo - 8 Aprile 2015

Il Primo - 18 Marzo



GrappaLUG

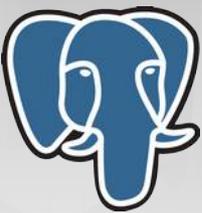
Gruppo utenti GNU/Linux di Bassano del Grappa



Riepilogo della puntata precedente

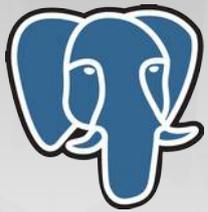
- Gestione della sicurezza:
 - Accesso al database: pg_hba.conf
 - Permessi sugli oggetti: CREATE USER/GRANT
- Creazione di tabelle
- Inserimento di dati mediante COPY
- Istruzioni SQL INSERT, UPDATE, DELETE

- Multi Version Concurrency Control
- Ogni query vede le sole transazioni completate prima della propria transazione
- Cos'è una transazione
 - Un blocco di istruzioni SQL comprese tra BEGIN e COMMIT
- All'interno di una transazione vede inoltre anche quanto fatto da query eseguite all'interno della stessa



Come funziona?

- Ogni tupla (riga) ha associato un identificativo della transazione che l'ha creata/modificata/cancellata
- Ogni tupla è visibile all'interno di una transazione se:
 - L'id che l'ha creata (AND):
 - È "committed"
 - E È inferiore all'id transazione associato alla transazione corrente
 - L'id che l'ha cancellata:
 - E' vuoto o appartiene ad una transazione annullata
 - E' superiore all'id transazione associato alla transazione corrente



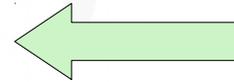
Esempio

Creazione	100
Eliminazione	



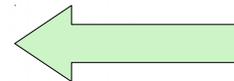
INSERT

Creazione	100
Eliminazione	120



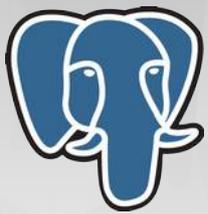
DELETE

Creazione	200
Eliminazione	210



UPDATE

Creazione	210
Eliminazione	

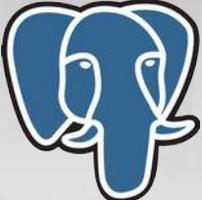


Esempio

Creazione	900
Eliminazione	
Creazione	1102
Eliminazione	
Creazione	900
Eliminazione	950
Creazione	1102
Eliminazione	1110
Creazione	900
Eliminazione	1102

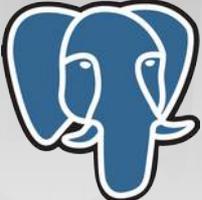
Il nostro ID è 1101:

- In verde le transazioni i cui risultati **sono** visibili
- In grigio le transazioni i cui risultati **non** sono visibili



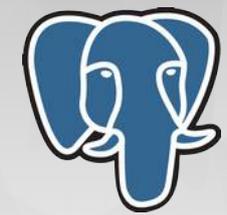
Esercizio 1

- Aprire due psql ed avviamo due transazioni
- Inserire un record in anagrafica in entrambe le transazioni
- Eseguire SELECT in entrambe le transazioni



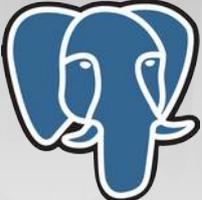
Esercizio 2

- Aprire due psql ed avviamo due transazioni
- In una eliminare un record dall'anagrafica
- Nell'altra eseguire un SELECT



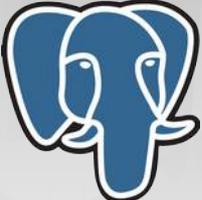
Gestione della concorrenza

- PostgreSQL è ACID compliant:
 - Atomicità
 - Consistenza
 - Isolamento
 - Durabilità
- Meccanismi di Locking:
 - SELECT ... FOR UPDATE
 - LOCK



Esercizio 3

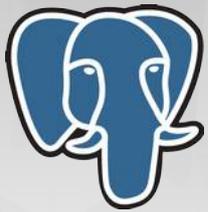
- Aprire due psql ed avviamo due transazioni
- Eseguire in una un LOCK esclusivo sulla tabella anagrafica
- Nell'altra eseguire SELECT



Esercizio 4

- Aprire due psql ed avviamo due transazioni
- In una eseguire un `SELECT ... FOR UPDATE` su un record
- Nell'altra provare ad eseguire un `UPDATE` dello stesso record

- Se si volesse salvare una transazione ad un certo punto?
- Creare un savepoint è la soluzione!
- Comandi:
 - SAVEPOINT nome
 - ROLLBACK TO SAVEPOINT nome
 - RELEASE TO SAVEPOINT nome



Esempi di SAVEPOINT - Esercizio 5

```
corso=> CREATE TABLE test_savepoint(  
    id INTEGER NOT NULL  
);  
  
BEGIN;  
    INSERT INTO table1 VALUES (1);  
    SAVEPOINT my_savepoint;  
    INSERT INTO table1 VALUES (2);  
    ROLLBACK TO SAVEPOINT my_savepoint;  
    INSERT INTO table1 VALUES (3);  
COMMIT;  
  
BEGIN;  
    INSERT INTO table1 VALUES (3);  
    SAVEPOINT my_savepoint;  
    INSERT INTO table1 VALUES (4);  
    RELEASE SAVEPOINT my_savepoint;  
COMMIT;
```

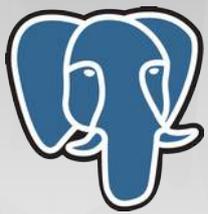
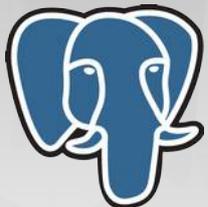


Table temporanee - Esercizio 6

- CREATE TEMP TABLE nome (...) [ON COMMIT DROP]
- SELECT * INTO TEMP nome FROM TABELLA
- Esercizio:
 - Aprire due psql e creare in una di esse una tabella temporanea
 - Verificare la visibilità della tabella nell'altra connessione
 - In una connessione:
 - (In transazione) creare una tabella temporanea con ON COMMIT DROP
 - Verificare la visibilità della tabella fuori e dentro la transazione



Evviva PostgreSQL!

Grazie!

Smart  Solutions

denis@gasparin.net

<http://www.gasparin.net>

Attribuzione – Non commerciale – Condividi allo stesso modo 3.0 Unported (CC BY-NC-SA 3.0)
<http://creativecommons.org/licenses/by-nc-sa/3.0/deed.it>