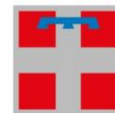




Cofinanziato
dall'Unione europea



REGIONE
PIEMONTE

FinTech Software Developer- 2023/2025

Basi di dati SQL

Docente: Roi Davide Simone

Titolo argomento: Architettura Postgresql – Federazione di Database

Roi Davide
Dispense

Federazione di Database

Una federazione di database consente a un database di accedere e utilizzare i dati presenti su altri database (installati su server diversi dal primo).

In pratica è una soluzione che permette la connessione tra più database e consente di leggere o scrivere dati come se fossero tutti presenti in un unico database.

Vantaggi e punti di forza dell'utilizzo di una Federazione di Database

- Riduzione dei costi di archiviazione dei dati: con una federazione di database, è possibile mantenere i dati in modo distribuito su più server, riducendo la necessità di avere un singolo database centralizzato di grandi dimensioni. Ciò può ridurre i costi di archiviazione dei dati.
- Miglioramento delle prestazioni: la distribuzione dei dati su più server consente di eseguire le query in modo parallelo, migliorando le prestazioni dell'applicazione.
- Flessibilità e scalabilità: con una federazione di database, è possibile aggiungere nuovi server al sistema in modo facile e rapido, consentendo di aumentare la capacità di archiviazione e di elaborazione dei dati in modo flessibile e scalabile.
- Accesso ai dati distribuiti: una federazione di database consente di accedere ai dati distribuiti su più server come se fossero presenti in un singolo database. Ciò consente di utilizzare i dati in modo efficiente e di semplificare le operazioni di gestione dei dati.
- Sicurezza dei dati: una federazione di database consente di mantenere i dati in modo distribuito su più server, riducendo il rischio di perdita di dati in caso di guasti hardware o software.

Federazione di Database

--Creare un utente per la federazione:

```
CREATE USER utente_federazione WITH PASSWORD 'password';
```

--Creare una estensione per la federazione:

```
CREATE EXTENSION postgres_fdw;
```

--Creare una connessione remota verso il database esterno, e la chiamo: "db_esterno_server":

--in questo passaggio inserisco gli estremi della connessione al database esterno (host, porta, user)

```
CREATE SERVER db_esterno_server  
FOREIGN DATA WRAPPER postgres_fdw  
OPTIONS (host 'nome_host_esterno', port 'numero_porta', dbname 'nome_utente_esterno');
```

--Creare un mapping tra l'utente locale e l'utente remoto:

--in questo passaggio inserisco gli estremi dell'utente esterno (user, password)

```
CREATE USER MAPPING FOR nome_utente_locale  
SERVER db_esterno_server  
OPTIONS (user 'nome_utente_esterno', password 'password_utente_esterno');
```

--Creare un foreign table nel database locale che rappresenti la tabella "clienti" del database esterno:

-- (devo dichiarare i nomi colonne e la tipologia di valori uguale alla tabella esterna che vado a leggere)

```
CREATE FOREIGN TABLE clienti  
(id int4, nome varchar(50), cognome varchar(50), email varchar(50), telefono varchar(20))  
SERVER db_esterno_server  
OPTIONS (schema_name 'public', table_name 'clienti');
```

--Lancio una select per verificare di leggere correttamente la tabella esterna:

```
SELECT * FROM clienti;
```

Fonti:

- Documentazione ufficiale di Postgresql
<https://www.postgresql.org/docs/>

Fine della Presentazione