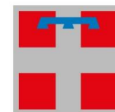




Cofinanziato
dall'Unione europea



REGIONE
PIEMONTE

FinTech Software Developer

Basi di dati SQL

Docente: Roi Davide Simone

Titolo argomento: Sviluppo su PostgreSQL – Creazione di Indici e tecniche di indicizzazione

Roi Davide
Dispense

Cosa sono gli INDICI

Liste ordinate per velocizzare ricerca DB
Nel caso di aggiunta/modifica = problemi

Gli indici sono liste ordinate utili per velocizzare la ricerca dei dati all'interno di una tabella.

Vantaggi:

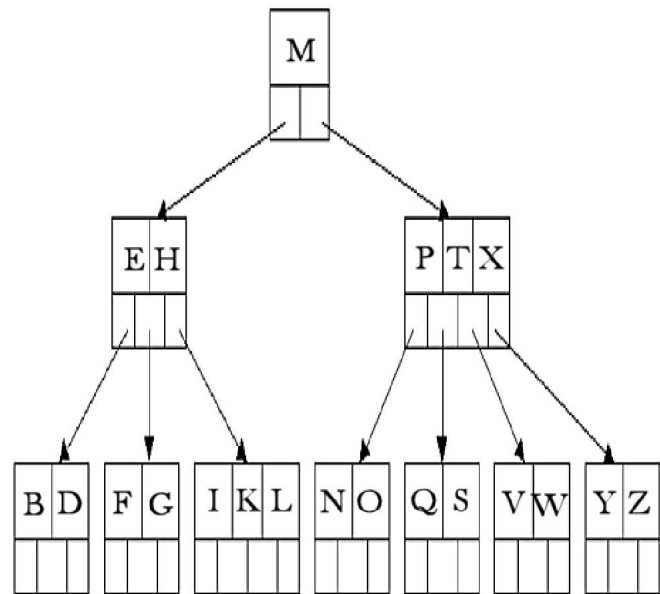
Velocizzano la lettura dei dati (SELECT) se utilizzati dall'ottimizzatore della QUERY

Svantaggi:

Rallentano le operazioni di modifica dei dati (INSERT, UPDATE, DELETE)

Esistono varie **tipologie di Indici** ma uno dei più efficaci e più utilizzati è il B-TREE

- B-TREE (Albero Bilanciato)



Di default gli indici creati in postgresql sono di tipo B-TREE se non specificato diversamente

Per creare un indice in postgresql il comando è il seguente:

CREATE INDEX nome_indice ON nome_tabella (nome_colonna);

Nell'esempio sopra, stiamo creando un indice chiamato "nome_indice" sulla tabella "nome_tabella" e sulla colonna "nome_colonna". L'indice verrà generato in base ai valori contenuti nella colonna specificata.

Per cancellare un indice in postgresql il comando è il seguente :

DROP INDEX nome_indice;

Per individuare le principali colonne da utilizzare nella creazione di un l'indice bisogna analizzare la query e identificare quali sono le colonne della tabella utilizzate all'interno delle condizioni di JOIN (quelle scritte dopo il comando **ON**) e all'interno del costrutto **WHERE**

Esempio di select da ottimizzare con gli indici:

```
select
  *
from
  ft_cliente as t1
inner join
  tabella_2 as t2 on
  t1.codice_fiscale = t2.colonna1
  and t1.cognome = t2.colonna2
where
  t1.data_nascita > '2000-01-01'
```

Esempio di creazione indici:

**CREATE INDEX idx_01_ft_cliente ON ft_cliente
(codice_fiscale, cognome, data_nascita);**

**CREATE INDEX idx_01_tabella_2 ON tabella_2
(colonna1, colonna2);**

Tenere presente che l'efficacia nell'utilizzo di un indice ha un legame diretto con:

- 1) L'utilizzo che decide di farne l'ottimizzatore del codice (non è detto che l'ottimizzatore del codice decida di utilizzare l'indice)
- 2) La dimensione dei dati contenuti (su tabelle con poche centinaia di record non ha senso la loro creazione)
- 3) La varietà dei contenuti per le colonne indicizzate (più i contenuti sono diversi tra loro e più sarà efficace l'utilizzo di un indice in quanto li riordina accorciando di molto i tempi di ricerca nel momento del suo utilizzo).

Per quanto esistano vari strumenti di monitoraggio dell'efficacia di un indice, la prova del 9 che determina la sua efficacia resta sempre quella di eseguire la query prima e dopo la creazione dell'indice e confrontare i tempi di esecuzione.

Opzione INCLUDE

L'opzione **INCLUDE** consente di includere colonne aggiuntive nell'indice che potrebbero essere utilizzate nelle clausole **SELECT** senza dover accedere alla tabella di origine. Questo può ridurre la necessità di leggere i dati dalla tabella principale, migliorando le prestazioni delle query. Ad esempio, se hai un indice sulla colonna "age" e desideri selezionare anche la colonna "name" e «email" nella tua query, puoi includere la colonna "name" e «email" nell'indice utilizzando INCLUDE.

```
CREATE INDEX idx_age ON users (age) INCLUDE (name, email);
```

L'ottimizzazione degli indici è un processo iterativo. Monitorare le prestazioni delle query, eseguire analisi delle query e valutare l'efficacia degli indici creati, permette di migliorare le prestazioni complessive del sistema.

È consigliabile consultare la documentazione ufficiale di PostgreSQL e utilizzare strumenti di analisi delle prestazioni per aiutarsi nella scelta delle colonne per l'indicizzazione in base al caso d'uso specifico.

Analisi degli indici:

EXPLAIN

Puoi utilizzare il comando EXPLAIN per ottenere il piano di esecuzione di una query. Questo ti darà informazioni dettagliate su come PostgreSQL eseguirà la query, compresi gli indici che utilizzerà. Ad esempio

```
EXPLAIN SELECT * FROM nome_tabella WHERE nome_colonna = 'valore_ricerca';
```

EXPLAIN ANALYZE

Il comando EXPLAIN ANALYZE offre una visualizzazione ancora più dettagliata del piano di esecuzione della query, inclusi i tempi di esecuzione stimati. Ad esempio:

```
EXPLAIN ANALYZE SELECT * FROM nome_tabella WHERE nome_colonna = 'valore_ricerca';
```

pg_stat_user_indexes:

La vista di sistema pg_stat_user_indexes fornisce informazioni sulle statistiche degli indici per l'utente corrente. Puoi utilizzare questa vista per controllare se un indice specifico viene effettivamente utilizzato dalle query. Ad esempio:

```
SELECT * FROM pg_stat_user_indexes WHERE indexname = 'nome_indice';
```

Questo comando restituirà una riga di output che corrisponde all'indice specificato. Puoi quindi controllare le colonne idx_scan e idx_tup_read per verificare se l'indice è stato scansionato o se sono state lette tuple tramite l'indice.

pg_stat_activity:

La vista di sistema pg_stat_activity fornisce informazioni sulle **attività correnti** nel database, inclusi i dettagli delle query in esecuzione. Puoi utilizzare questa vista per controllare quali query stanno utilizzando un indice specifico. Ad esempio:

```
SELECT * FROM pg_stat_activity WHERE query LIKE '%nome_indice%';
```

Questo comando restituirà le righe corrispondenti alle query che contengono il nome dell'indice specificato.

Utilizzando questi comandi e viste di sistema, puoi verificare se una query utilizza un certo indice che hai creato e monitorare le prestazioni degli indici nel tuo database PostgreSQL.

Fonti:

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

Fine della Presentazione