
Bones pràctiques en SQL

Codificació de transaccions

PID_00253061

Alexandre Pereiras Magariños



Universitat
Oberta
de Catalunya

Serie de vídeos

1. Codificación SQL
2. Codificación de consultas
3. Codificación de procedimientos/funciones
4. **Codificación de transacciones**

EIMT.UOC.EDU

Benvinguts al quart i últim vídeo de la sèrie *Bones pràctiques en SQL*, sèrie en què veurem un conjunt de bones pràctiques a l'hora de programar SQL en entorns de bases de dades/*data warehouse*.

Aquesta sèrie de vídeos està dividida en 4 parts:

- *Codificació SQL*: centrada en aquelles pràctiques des del punt de vista genèric en SQL, que afecten la llegibilitat i la portabilitat del codi SQL.
- *Codificació de consultes*: en què veurem pràctiques que ens ajudaran a generar consultes SQL més eficients i llegibles.
- *Codificació de procediments i funcions*: en què pararem esment en aquelles pràctiques que ens permetran escriure el codi SQL en un servidor més eficient i gestionar errors de forma controlada.
- *Codificació de transaccions*: en què detallarem pràctiques per a assegurar un control correcte de les transaccions que codifiquem.

És important destacar que, dins de cada categoria, no es proporcionen totes les possibles bones pràctiques del mercat, sinó que es tracta d'un conjunt concret que des de la UOC hem considerat més rellevants.

Aquest vídeo afrontarà l'última part, *Codificació de transaccions*.

Índice

- Codificación de transacciones
- Referencias

Aquest quart i últim vídeo se centrarà a presentar una sèrie de bones pràctiques de codificació de transaccions, que ens permetran generar un codi eficient i evitar problemes de concurrència i bloquejos. Al final del vídeo, us proporcionarem les referències bibliogràfiques utilitzades.

Índice

- Codificación de transacciones
- Referencias

Vegem, doncs, l'última categoria de bones pràctiques: codificació de transaccions.

Codificación de transacciones

- Mejorar la eficiencia del código
- Mejorar el acceso concurrente y evitar bloqueos
- Puntos a considerar:
 - Crear transacciones lo más cortas y concisas posibles
 - Utilización de las cláusulas `START TRANSACTION/BEGIN TRANSACTION` y `COMMIT/ROLLBACK`

EIMT.UOC.EDU

Quan codifiquem transaccions de manera ineficient, això pot tenir un impacte en l'accés concurrent a les dades i causar així un rendiment global pobre de l'aplicació. Amb la finalitat de millorar l'eficiència del codi i evitar problemes en l'accés concurrent, a continuació veurem una sèrie de punts a tenir en consideració a l'hora de codificar transaccions en SQL:

- *Mantenir les transaccions el més curtes i concises possible:* la creació de transaccions curtes en temps ens evita problemes de bloquejos i dependències amb altres transaccions. Es recomana no incloure un codi SQL dins d'una transacció que no sigui essencial per a dur a terme aquesta transacció, com per exemple consultes SQL de recuperació de dades que no s'utilitzen, ja que això perllongarà la transacció de forma innecessària.
- *Utilització de les clàusules `START TRANSACTION/BEGIN TRANSACTION` i `COMMIT/ROLLBACK`:* és molt important que les transaccions estiguin ben delimitades amb les ordres `BEGIN` i `COMMIT`, o en el seu defecte, `ROLLBACK`. D'aquesta forma, evitem deixar transaccions obertes i sofrir possibles pèrdues de dades.

Cláusulas Transacciones - Ejemplo

```
START TRANSACTION;
```

```
-- Recuperar datos  
[...]
```

```
-- Insertar datos  
[...]
```

```
-- Actualizar datos de nuevo  
[...]
```

```
COMMIT;
```

← Comienzo de la transacción

← Lecturas y modificaciones de datos

← Confirmación de los cambios hechos por la transacción

Vegem un exemple de com utilitzar aquestes clàusules de manera senzilla. Com podem veure a la imatge, observem que s'inicia una transacció amb `START TRANSACTION`, en què s'executa un codi específic: recuperació de dades, inserció de dades i actualització de dades. Si aquest codi s'executa de forma correcta, els canvis realitzats per la transacció es confirmaran mitjançant la clàusula `COMMIT`, acabant així la transacció.

Codificación de transacciones

- Mejorar la eficiencia del código
- Mejorar el acceso concurrente y evitar bloqueos
- Puntos a considerar:
 - Crear transacciones lo más cortas y concisas posibles
 - Utilización de las cláusulas START TRANSACTION/BEGIN TRANSACTION y COMMIT/ROLLBACK
 - Nunca permitir el uso de *prompts* al usuario durante una transacción
 - Establecer el nivel de aislamiento correcto en cada transacción

EIMT.UOC.EDU

Com a últim punt a destacar:

- *No permetre mai l'ús de prompts a l'usuari en transaccions*: el fet de requerir a l'usuari que proporcioni algun valor enmig d'una transacció fa que perdem el control de la durada de la transacció, per la qual cosa mai no ho hem de permetre. Aquest tipus d'accions s'ha de realitzar fora d'una transacció.

Imagineu que llanceu una transacció que, a la meitat d'aquesta, llança un *prompt* que demana un codi a l'usuari. Com que no sabeu quan la transacció us demanarà el codi, us en aneu a prendre un cafè amb els vostres companys mentre la transacció s'executa, amb tanta mala sort que just quan heu deixat el vostre PC, la transacció us demana l'entrada del codi. Aquesta transacció estarà oberta i esperant que torneu, de manera que aquelles estructures que heu modificat estaran bloquejades fins que la vostra transacció finalitzi, per la qual cosa altres transaccions estaran a l'espera dels recursos que la vostra transacció ha bloquejat, causant problemes de concurrència en la base de dades.

- *Establir el nivell d'aïllament correcte a cada transacció*: una bona pràctica a l'hora d'implementar transaccions és la d'establir el nivell d'aïllament correcte. En aquells casos en què no sabem si les nostres transaccions s'executaran concurrentment amb unes altres i sigui necessari l'aïllament, llavors es recomana usar el nivell màxim. En canvi, si l'aïllament no ens

preocupa o sabem per endavant quines altres transaccions s'executaran amb les nostres, llavors sempre podem definir el nivell d'aïllament més adequat. Si ho podem relaxar a un nivell inferior, podrem millorar el rendiment de la nostra base de dades o aplicacions.

Índice

- Codificació de transaccions
- Referencias

I fins aquí hem arribat amb aquest últim vídeo de la sèrie *Bones pràctiques en SQL*. Esperem que us hagi agradat la presentació i, en general, la sèrie de vídeos, i que us hagi servit de molta ajuda.

Ara us presentarem un conjunt d'enllaços i referències d'interès sobre aquest tema.

Referencias

Rankins, R.; Bertucci, P.; Gallelli, C.; Silverstein, A.. (2013).
Microsoft® SQL Server 2012 Unleashed. Sams.

PostgreSQL:

<http://www.postgresql.org/docs/9.3/>

SQL Server Performance:

<http://www.sql-server-performance.com/2001/sql-best-practices/>

Que tingueu un bon dia!