
Bones pràctiques en SQL

Codificació SQL

PID_00253042

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 primer 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 la primera part, *Codificació SQL*.

Índice

- Introducción
- Codificación SQL
- Referencias

EIMT.UOC.EDU

L'estructura d'aquest primer vídeo és la següent: primer veurem una petita introducció a les bones pràctiques en SQL i per què és important tenir-les en consideració: entre d'altres, ens faciliten la lectura del codi i milloren el rendiment o gestionen les dades de forma més eficient. Amb la finalitat d'ajudar-vos a escriure i programar l'SQL més llegible i eficient, a continuació veurem una sèrie de bones pràctiques de codificació SQL, que ens permetran generar el codi d'una forma llegible i portable, i us proposarem alguns exemples per a facilitar el seu enteniment. Finalment, us proporcionarem les referències bibliogràfiques utilitzades.

Índice

- Introducció
- Codificació SQL
- Referències

Vegem, doncs, el primer punt, que és la introducció a les bones pràctiques.

Introducción

- Escribir código SQL no es fácil
- Escribir código SQL eficiente es aún más difícil
- El código ineficiente afecta a las aplicaciones

EIMT.UOC.EDU

Escriure codi SQL no és una tasca fàcil. La construcció de consultes i blocs de codi requereix un cert nivell de destresa que s'obté mitjançant l'estudi i l'ús del llenguatge. Codificar codi SQL eficient i llegible encara és més difícil. No solament és important conèixer el llenguatge a fons, sinó que requereix un coneixement pràctic que s'adquireix mitjançant l'ús constant d'aquest llenguatge i d'una experiència en el desenvolupament del codi per a aplicacions. Aquest coneixement pràctic és indispensable quan treballem amb sistemes gestors de bases de dades (SGBD): malgrat que la configuració d'un SGBD i el disseny de la base de dades es puguin considerar òptims, el codi SQL es pot haver escrit de manera ineficient, causant així un problema de rendiment i afectant les aplicacions d'usuari.

Índice

- Introducció
- Codificació SQL
- Referències

Després d'aquesta introducció, vegem la primera categoria de bones pràctiques: codificació de codi SQL.

Codificación SQL

- Mejorar la legibilidad y la portabilidad del código
- Puntos a considerar:
 - Establecer una convención de nombres
 - Sangrado de código

EIMT.UOC.EDU

En aquesta secció veurem una sèrie de punts genèrics que es refereixen, especialment, a com hem de codificar SQL de manera que aquest sigui el més llegible possible i, a més, faciliti la portabilitat del codi en cas de possibles migracions a altres SGBD. Des d'aquest punt de vista, es recomana seguir els punts següents:

- *Establir una convenció de noms*: el fet de nomenar objectes de forma consistent millora l'aparença del codi, facilita la seva llegibilitat i interpretació, i el manteniment d'aquest, a més de proporcionar informació addicional sobre l'ús d'aquests objectes en les aplicacions.
- *Sagnat del codi*: entenem per sagnat el moviment de blocs de text cap a la dreta mitjançant espais o tabulacions per a separar-lo del marge esquerre i, així, distingir-lo del text adjacent. En informàtica, se sol utilitzar més el terme *indentació*, encara que aquest és considerat un anglicisme. L'ús del sagnat en codi SQL millora la llegibilitat per part dels programadors i facilita el manteniment.

Sangrado de código – Ejemplo

```
SELECT E.EMPLOYEE_CODE, E.EMPLOYEE_NAME,  
       E.ADDRESS, E.POST_CODE,  
       E.BIRTH_DATE, E.DEPARTMENT_CODE,  
       D.DEPARTMENT_NAME FROM EMPLOYEE E  
INNER JOIN DEPARTMENT D  
ON  
E.DEPARTMENT_CODE =  
D.DEPARTMENT_CODE
```



```
SELECT  
    E.EMPLOYEE_CODE,  
    E.EMPLOYEE_NAME,  
    E.ADDRESS,  
    E.POST_CODE,  
    E.BIRTH_DATE,  
    E.DEPARTMENT_CODE,  
    D.DEPARTMENT_NAME  
FROM  
    EMPLOYEE E  
INNER JOIN DEPARTMENT D ON  
    E.DEPARTMENT_CODE = D.DEPARTMENT_CODE
```



EIMT.UOC.EDU

Vegem l'exemple següent de sagnat. La consulta SQL següent obté informació d'empleats i del departament a què pertanyen. Com podem veure a la imatge, malgrat que és una consulta relativament simple, aquesta és complexa d'entendre: és difícil distingir quines columnes s'obtenen i quines taules s'utilitzen. Imagineu-vos si es tractés d'una consulta amb 20 taules i 80 columnes en el `SELECT`! La forma en què aquesta consulta està codificada és un exemple clar de com no hem de codificar SQL.

En la segona imatge, observem la mateixa consulta a què s'ha aplicat un sagnat de codi. En aquest cas, veiem que és molt més senzill identificar les diferents parts de la consulta (`SELECT`, `FROM`, `INNER JOIN`). Aquest és un exemple de com sí que hem de codificar SQL.

Codificación SQL

- Mejorar la legibilidad y la portabilidad del código
- Puntos a considerar:
 - Establecer una convención de nombres
 - Sangrado de código
 - Uso de SQL estándar

Continuem amb el punt següent a considerar en la codificació SQL:

- *Ús d'SQL estàndard:* molts dels SGBD actuals implementen clàusules que no formen part de l'ANSI estàndard. L'ús d'aquestes notacions pròpies no solament dificulta la lectura del codi, sinó que també complica la migració del codi d'un SGBD a un altre. A més del que hem esmentat, l'ús de clàusules estàndards com ara `JOIN`, en lloc de la forma proposada en SQL:1989, facilita que en la clàusula `WHERE` solament s'afegeixin condicions i no el codi per a la realització d'operacions de combinació.

Uso de SQL estándar – Ejemplo

```
SELECT
  E.EMPLOYEE_CODE,
  E.EMPLOYEE_NAME,
  E.ADDRESS,
  E.POST_CODE,
  E.BIRTH_DATE,
  E.DEPARTMENT_CODE,
  D.DEPARTMENT_NAME,
  SH.SALARY,
  SH.SALARY_FROM_DATE
FROM
  EMPLOYEE E,
  DEPARTMENT D,
  SALARY_HISTORY SH
WHERE
  E.DEPARTMENT_CODE = 'SALES'
AND E.DEPARTMENT_CODE = D.DEPARTMENT_CODE (+)
AND E.EMPLOYEE_CODE = SH.EMPLOYEE_CODE (+)
```



```
SELECT
  E.EMPLOYEE_CODE,
  E.EMPLOYEE_NAME,
  E.ADDRESS,
  E.POST_CODE,
  E.BIRTH_DATE,
  E.DEPARTMENT_CODE,
  D.DEPARTMENT_NAME,
  SH.SALARY,
  SH.SALARY_FROM_DATE
FROM
  EMPLOYEE E
LEFT OUTER JOIN DEPARTMENT D ON
  E.DEPARTMENT_CODE = D.DEPARTMENT_CODE
LEFT OUTER JOIN SALARY_HISTORY SH ON
  E.EMPLOYEE_CODE = SH.EMPLOYEE_CODE
WHERE
  E.DEPARTMENT_CODE = 'SALES'
```



EIMT.UOC.EDU

Vegem l'exemple següent d'ús de l'SQL estàndard. En el cas d'Oracle, la clàusula `OUTER JOIN` es pot implementar mitjançant la notació `(+)`. Aquesta notació és pròpia d'Oracle i no és vàlida per a altres SGBD com ara PostgreSQL, SQL Server o DB2. En aquests casos, es recomana utilitzar la clàusula SQL estàndard `OUTER JOIN` (com es pot veure en la segona imatge), ja que no solament facilita la lectura de la consulta, sinó que a més ens evitarà problemes en cas d'una possible migració a un SGBD diferent. En la mesura que es pugui, sempre es recomana l'ús de l'SQL estàndard.

Codificación SQL

- Mejorar la legibilidad y la portabilidad del código
- Puntos a considerar:
 - Establecer una convención de nombres
 - Sangrado de código
 - Uso de SQL estándar
 - Utilización de entornos similares al de producción para tareas de testeo

Com a últim punt de la llista a considerar en codificació SQL:

- *Utilització d'entorns similars al de producció per al testeig*: quan es creen aplicacions que utilitzen bases de dades, sol haver-hi una separació d'entorns per a facilitar el desenvolupament de la fase de testeig i aquest de l'entorn de producció. Aquests entorns de testeig haurien de ser, en la mesura que es pugui i des del punt de vista del maquinari i programari, similars a l'entorn de producció. Una de les raons principals és la de garantir que l'aplicació s'executi de forma correcta, però també és molt important que aquelles consultes implementades s'executin de forma correcta. La disponibilitat d'un entorn similar al de producció ens facilita realitzar proves de rendiment en l'SGBD i verificar que el codi SQL generat sigui eficient segons les necessitats de l'aplicació.

Índice

- Introducción
- Codificación SQL
- Referencias

EIMT.UOC.EDU

I fins aquí hem arribat amb aquest primer vídeo de la sèrie *Bones pràctiques en SQL*. Esperem que us hagi agradat la presentació i que aquesta 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!