

ADVERTENCIAS:

- Para el desarrollo de esta práctica se necesitan los contenidos teóricos aportados en los temas 1, 2, 6, 8, 9 y 10.

PRACTICA N° 9 PARTICIONADO DE ÍNDICES

A fin de poder comprender adecuadamente los conceptos necesarios para el desarrollo de esta práctica, estudie detenidamente la parte correspondiente a particionado de índices de los siguientes manuales:

- Data Warehousing Guide
- Concepts
- SQL Reference: Sintaxis y semántica de las sentencias:
 - CREATE INDEX y
 - ALTER INDEX

Y además, específicamente para la pregunta 2 de esta práctica, se usará la sección 8-4 (*constraint*) del manual SQL Reference.

Comprobar que existe el usuario 'ALUMNO' con las características especificadas en la práctica anterior.

Comprobar que los tablespaces PRUEBA y PRUEBA_TEMP están creados de manera que el primero está soportado por dos ficheros, uno en el disco C: (o E:) y otro en el disco F:, debiendo ser uno de ellos autoextensible (el del disco F:); y el segundo está soportado por un único fichero en el disco F:, también autoextensible.

A lo largo de toda esta práctica se usa la misma estructura relacional para la tabla B del usuario 'ALUMNO' que en la práctica anterior.

Pregunta 1: Desarrolle los scripts para realizar las tareas indicadas y para cada uno indique si el índice es prefijado o no:

S11 = 1,4,5	S21 = 2,4,5	S31 = 3,4,5
S12 = 1,4,6	S22 = 2,4,6	S32 = 3,4,6
S13 = 1,4,7	S23 = 2,4,7	S33 = 3,4,7
S14 = 1,4,8	S24 = 2,4,8	S34 = 3,4,8
S15 = 1,4,9	S25 = 2,4,9	S35 = 3,4,9
S16 = 1,4,10	S26 = 2,4,10	S36 = 3,4,10

Tareas a realizar:

1. Crear la tabla 'B' con particionado por rango por el campo fecha, según se indica:
 - Partición 'añoencurso': tuplas con fecha del año actual ó posteriores, en el tablespace prueba_temp.
 - Partición 'añoyyyy' (siendo yyyy el año pasado): tuplas con fecha del año pasado, en el tablespace prueba.
 - Partición 'añoyyyy' (siendo yyyy el año ante-pasado): tuplas con fecha del año ante-pasado, en el tablespace prueba.

- Partición ‘restoaños’: tuplas con el resto de fechas, en el tablespace prueba_temp.
2. Crear la tabla ‘B’ con particionado por hash con 4 particiones, atendiendo al campo fecha.
 3. Crear la tabla ‘B’ con particionado por listas según el campo b, de la siguiente forma:
 - ‘Vocales’ con aquellas tuplas con valor del campo b (‘A’, ‘E’, ‘I’, ‘O’, ‘U’)
 - ‘Consonantes’ con aquellas tuplas con el resto de valores.
 4. Cargar 400.000 de valores, mediante el procedimiento inserta_en_b creado en la práctica anterior.
 5. Crear un índice local por ‘fecha’.
 6. Crear un índice local por ‘c’.
 7. Crear un índice global por ‘fecha’, particionado en rango, por el campo ‘fecha’.
 8. Crear un índice global por ‘fecha’, particionado por hash, por el campo ‘fecha’.
 9. Crear un índice global por ‘fecha’, particionado en rango, por el campo ‘c’.
 10. Crear un índice global por ‘fecha’, particionado por hash, por el campo ‘c’.

Pregunta 2: Desarrolle los scripts para realizar las tareas indicadas y cuando no sea posible particionar el índice de la clave primaria tal y como se pide, deduzca la razón:

R11 = 1,4,5	R21 = 2,4,5	R31 = 3,4,5
R12 = 1,4,6	R22 = 2,4,6	R32 = 3,4,6
R13 = 1,4,7	R23 = 2,4,7	R33 = 3,4,7
R14 = 1,4,8	R24 = 2,4,8	R34 = 3,4,8
R15 = 1,4,9	R25 = 2,4,9	R35 = 3,4,9

Tareas a realizar:

1. Crear la tabla ‘B’ con particionado por rango por el campo ‘fecha’, según se indica:
 - Partición ‘añoencurso’: tuplas con fecha del año actual ó posteriores, en el tablespace prueba_temp.
 - Partición ‘añoyyyy’ (siendo yyyy el año pasado): tuplas con fecha del año pasado, en el tablespace prueba.
 - Partición ‘añoyyyy’ (siendo yyyy el año ante-pasado): tuplas con fecha del año ante-pasado, en el tablespace prueba.
 - Partición ‘restoaños’: tuplas con el resto de fechas, en el tablespace prueba_temp.
2. Crear la tabla ‘B’ con particionado por hash con 4 particiones, atendiendo al campo ‘a’.
3. Crear la tabla ‘B’ con particionado por listas según el campo ‘b’, de la siguiente forma:
 - ‘Vocales’ con aquellas tuplas con valor del campo b (‘A’, ‘E’, ‘I’, ‘O’, ‘U’)
 - ‘Consonantes’ con aquellas tuplas con el resto de valores.
4. Cargar, mediante el procedimiento inserta_en_b 400.000 de valores.
5. Crear un índice para la clave primaria con particionado local.

6. Crear un índice para la clave primaria con un particionado global en rango por el campo 'fecha'.
7. Crear un índice para la clave primaria con un particionado global en rango por el campo 'a'.
8. Crear un índice para la clave primaria con un particionado global en hash por el campo 'fecha'.
9. Crear un índice para la clave primaria con un particionado global en hash por el campo 'a'.

Pregunta 3: ¿Es posible realizar un particionado de índice global coincidente con el particionado de la tabla? En caso afirmativo, ¿qué diferencias existirían con el particionado local? Genere un script para ejemplarizar las diferencias.

Notas de interés:

- En un índice global particionado en rango es obligatorio el uso de *maxvalue*.
- Oracle solo admite índices globales particionados en rango o en hash.
- En un índice particionado global cualquier partición del índice puede señalar a cualquier partición de la tabla, mientras que en un índice particionado local cada partición del índice señala a una única partición de la tabla.
- Un índice particionado local presenta siempre un equiparticionado con su tabla subyacente de manera automática.
- Un índice no particionado es un índice global.
- Un índice prefijado posibilita al optimizador podar particiones del índice para cualquier búsqueda indexada.
- Un índice no prefijado posibilita al optimizador podar particiones del índice para las búsquedas indexadas que incluyan algún campo de la clave de partición.
- Oracle no permite índices globales no prefijados.
- En Oracle, un índice puede ser único si dilucida la unicidad en un único árbol. Por lo tanto, un índice particionado podrá ser único si los campos que forman la clave de particionado constituyen un subconjunto de los campos que forman la clave de indización (no importa el orden ni la contigüidad).
- Un índice local no prefijado posibilita buscar de forma indexada por un campo que no esté prefijado por su clave de particionado. Las ventanas de tiempo presentadas en la práctica anterior constituyen un paradigma de este caso: es evidente que el índice de la tabla particionada ha de ser local para evitar su reconstrucción completa en cada operación de *exchange* y que está particionado por fecha; necesariamente será no prefijado cuando se desee buscar de forma indexada por cualquier campo de la tabla que no sea la fecha.
- Los índices particionados globales son imprescindibles para las claves únicas de las tablas particionadas por un campo diferente a esa clave. Por ejemplo, una tabla con clave primaria 'a' y particionada por 'b' no puede tener un índice único local para 'a' (tiene que estar indexado y particionado por 'a').
- Un índice particionado global no puede reconstruirse (*rebuild*) como un todo. Ha de reconstruirse partición a partición. (*alter index ... rebuild partition ...;*). O bien, se borra y se vuelve a crear (completo, evidentemente).

- En un índice particionado en rango, cuando se borra una partición del índice (el índice ha de ser global, evidentemente, porque los índices locales no permiten alterar su particionado directamente, sino a través de su tabla), la partición que le sigue queda inválida, dado que ha de acoger a los valores de clave del subárbol de la partición borrada. Las otras subsecuentes, siguen válidas.