3ª PRÁCTICA – Diseño Físico en Oracle® BD

Una de las funciones de un DBA es monitorizar y afinar el sistema. En esta práctica vamos a simular esta situación, aunque simplificaremos significativamente los procesos. Concretamente, nos centraremos en el diseño de la representación física del sistema.

Partiremos de la base de datos que estamos utilizando en el resto de las prácticas, y en concreto, en la solución de la práctica 1 que hemos propuesto como solución (tanto el modelo como los datos). Así pues, el primer paso a realizar en esta práctica es establecer el entorno de evaluación. Para ello, se eliminarán el resto de los objetos de la base de datos que puedan dificultar la evaluación (vistas, índices, *triggers*, …), así como los datos utilizados para pruebas y que no son parte de este escenario. Tras la eliminación de estos objetos, ejecutarán los scripts que crean y pueblan la BD ("createBD_2023.sql" y "insert2023 - errors_fixed.sql") y que fueron proporcionados para la P2. Además, se modificará la definición de la tabla "tracks", añadiendo dos nuevas columnas (*searchk* y *lyrics*). El código PL/SQL que crea estas columnas y que proporciona los valores a una de ellas debe ejecutarse una vez (tras ejecutar los dos scripts mencionados). Este código se ofrece en el primer anexo de este documento, y está también incorporado al script de la P3.

El primer paso que hemos dado como DBA, es analizar las transacciones que se producen en la base de datos, centrándose en las operaciones o procesos más costosos a realizar. Por supuesto, no todos los procesos se realizan con la misma frecuencia. Este análisis ya se ha realizado para nuestra base de datos, y se ha obtenido que las operaciones más importantes son:

- Proceso P_1: modificar una fila (selección por el atributo *searchk*)
- Proceso P_2: consulta 1 de la práctica 2.
- Proceso P_3: consulta 2 de la práctica 2.

Las frecuencias relativas de estos tres procesos son {0.98,0.01,0.01} (es decir, por cada vez que se ejecuta cada consulta, se ejecutan 98 operaciones de actualización). El código PL/SQL que implementa estos procesos se ofrece en el anexo, al final de este documento. A partir de ese código, se ha creado una carga de trabajo (*workload*) que incluye estas operaciones para ser ejecutadas en la proporción y orden adecuados. Esto es especialmente relevante, ya que los procesos de actualización afectan al rendimiento del resto de procesos. Por ello, en este escenario de la práctica, no se permite alterar el orden de ejecución de las operaciones. Esta metodología de análisis y evaluación de diseños físicos (evaluación de un workload sobre un entorno controlado) no es la única forma de trabajar para este fin, pero es adecuada en esta práctica.

Junto a este enunciado, se adjunta un script (script_statistics_2023.sql) que al ejecutarse crea un paquete (PKG_COSTES) en la base de datos. Dentro de este paquete, hay tres procedimientos que implementan la carga de trabajo mencionada, y posibilitan su ejecución: el procedimiento PR_PREPARE inicializa la variable que decide qué operaciones de actualización se van a realizar; el procedimiento PR_WORKLOAD que contiene la carga de trabajo propiamente dicha; y el procedimiento PR_RESET que se encarga de devolver la base de datos a la situación inicial (ya que la ejecución de la carga de trabajo altera el estado de la base de datos). También contiene un procedimiento auxiliar para el control del tiempo, y un procedimiento (RUN_TEST) que se encarga de repetir la carga de trabajo una o más veces (indicándolo en el parámetro *ite* del procedimiento).

Este procedimiento (RUN_TEST) es el único que los alumnos deben ejecutar para realizar las mediciones (calcula y muestra el tiempo medio y el número de accesos de las n repeticiones de la carga de trabajo). Opcionalmente, los alumnos pueden modificar el script con tres intenciones:

- Modificar el procedimiento PR_WORKLOAD para añadir directrices de ejecución (*hints*)
- Modificar PR_RESET para inicializar el estado de alguna estructura de su diseño físico que se pudiera ver afectada por las operaciones de actualización del WL (es decir, que afecte a la tabla *tracks*)
- Modificar el experimento, eliminando la inicialización de las estructuras para estudiar otro escenario.

Por ello, el siguiente paso es medir el rendimiento de la carga de trabajo estándar para ver la situación inicial de partida. Para ello, ejecuta el paquete mencionado, para tener disponible el procedimiento que se encarga de esta acción. Se debe tener presente que el número de iteraciones a repetir el workload no debería ser menor a 5 si se quiere tener una estimación mínimamente fiable (aunque se recomiendan entre 10 y 20 iteraciones, no son necesarias más de 10 iter.).

Una vez que conocemos la situación inicial, lo siguiente es analizarla en profundidad. Esta es la parte más importante de la práctica. Para ello, analizaremos (a) el diseño físico de partida; (b) la naturaleza de las operaciones que se van a ejecutar; (c) el plan de ejecución de cada operación (algoritmos utilizados en su resolución), proporcionando detalles de su ejecución; y (d) los costes medios calculados sobre varias ejecuciones.

A partir de este estudio, desarrollaremos nuestra propuesta de diseño físico orientada a reducir el número de accesos a memoria secundaria, y el tiempo de ejecución global. Es decir, los cambios que el DBA propondría para mejorar (afinar) el sistema. Las propuestas de diseño físico pueden conllevar cambios que requieran reconstruir parte del sistema. Por esto, en un sistema ya existente los cambios estructurales son difíciles de implementar, porque implica alterar estructuras en producción que ya contienen datos (en un sistema de nueva creación, los cambios suelen ser fácilmente aplicables). En esta práctica se permite casi todas las estructuras estudiadas en la asignatura que están dentro de las posibilidades del SGBD: cambiar parámetros físicos de los objetos (pctfree, tablespace, etc), crear estructuras (clusters monotabla o multitabla, índices, etc), añadir directrices de ejecución, ... No se permite utilizar vistas materializadas. De modo obligatorio, se deberá considerar (al menos) la inclusión de un índice y de un clúster (se incluirán su diseño, implementación y evaluación en la memoria; si bien al analizar su rendimiento se puede concluir que no es adecuado y eliminarlo del diseño final).

El diseño físico será implementado sobre las estructuras originales (es decir, sobre el script createBD_2023.sql). Tras ejecutarlo, deberá poblarse de nuevo la base (“insert2023 - errors_fixed.sql”) y será necesario también crear las dos nuevas columnas de la tabla “tracks” (ese código puede añadirse al final del script createBD para mayor comodidad). En caso de haber alterado el WL, será también necesario volver a crear PKG_COSTES ejecutando el script.

El último paso sería comprobar que las mejoras han surgido el efecto deseado. Se deben comparar los resultados de la ejecución del workload sobre el diseño físico inicial, con los resultados obtenidos sobre el diseño físico completo propuesto e implementado. Opcionalmente, se puede ampliar el análisis realizando un nuevo diseño físico y evaluándolo (habitualmente, el proceso de monitorizar y afinar el sistema es un proceso de refinamiento continuo, si bien en esta práctica esto no es preciso, y sólo se propone mejorar la situación de partida). También se podría (opcionalmente) modificar el experimento para evaluar el rendimiento de sucesivas ejecuciones sin que se realice la operación de reinicio (del estado de la BD).

Titulación: GRADO INGENIERIA INFORMATICA Año Académico: 2022/2023 -- 2º Cuatrimestre Asignatura: Ficheros y bases de datos Título: Enunciado Práctica 3	uc3m | Universidad Carlos III de Madrid

Resumen de pasos a realizar

- **Situación de partida**: restablece el estado original de la BD.
- **Mide** el rendimiento de la carga de trabajo estándar (tanto en tiempo como en número de accesos) a través de varias repeticiones.
- **Analiza** el plan de ejecución para cada operación, extrayendo los procesos (la tipología) de los que se compone. Clasifica y ordena todos los procesos en la estructura física, su frecuencia y una aproximación del coste.
- A partir de ese estudio, desarrolla una **propuesta** para reducir el número de accesos a soporte secundario.
- Describe el **diseño físico completo**, comenzando por las organizaciones básicas involucradas en el mismo, así como las estructuras auxiliares utilizadas para mejorar el rendimiento.
- **Implementa** el diseño físico completo propuesto, y mide su rendimiento sobre la carga de trabajo estándar propuesta.
- **Compara** el rendimiento del diseño físico original frente al propuesto e interpreta los resultados. Opcionalmente, puedes refinar la propuesta, e implementar mejoras al diseño físico (cuyo rendimiento después analizarás y compararás con los anteriores).
- **Documenta** todo el trabajo realizado en esta práctica, poniendo especial atención al diseño físico, a las medidas de rendimiento, a los planes de ejecución, a las propuestas de mejora, y a la comparación de resultados.

Documentación a entregar

Se debe entregar una memoria que contenga todos los pasos anteriormente descritos. Además, contendrá el código de la implementación del diseño físico completo que has realizado. Si has modificado el script de creación de la BD, o el script con el paquete que contiene el workload, debes también incluirlo en la memoria. Se permite también, de forma adicional, añadir estos scripts como elementos a entregar.

Material de Apoyo:

- Diapositivas correspondientes a la sesión de laboratorio.
- SQL Scripts (sintaxis Oracle PL/SQL): carga de trabajo estándar y procedimientos relacionados con las medidas de rendimiento basadas en consultar las estadísticas que recoge y almacena Oracle.
- Cuenta de Oracle DBMS 12c.

ANEXO: Código de apoyo

Modificaciones estructurales: (creación de dos columnas en 'tracks' y valuación de una de ellas)

```
ALTER TABLE tracks add (searchk varchar2(20), lyrics VARCHAR2(4000));
UPDATE tracks set searchk=pair||'//'||sequ;
COMMIT;
```

Código para preparar la WL:

```
select searchk bulk collect into changes
   from (select searchk from tracks order by dbms_random.value)
   where rownum<=98;
```

Carga de trabajo - Actualización de una fila:

```
UPDATE tracks
   set lyrics = dbms_random.string('a',dbms_random.value(900,1200))
   where searchk=changes(i);
```

Carga de trabajo – Consulta 1:

```
FOR fila in (
WITH authors as (select title,writer, writer musician from songs
                 UNION select title,writer,cowriter musician from songs),
     authorship as (select distinct band performer, title, writer, 1 flag
                         FROM involvement join authors using(musician) ),
     recordings as (select performer,tracks.title,writer
                         from albums join tracks using(pair)),
     recs_match as (select performer,
                             round(sum(flag)*100/count('c'),2) pct_recs
                         from recordings left join authorship
                              using(performer,title,writer)
                         group by performer),
     pers_match as (select performer,
                              round(sum(flag)*100/count('c'),2) pct_pers
                         from (select performer, songtitle title,
                                      songwriter writer
                                 from performances) P
                              left join authorship
                              using(performer,title,writer)
                         group by performer)
SELECT performer, pct_recs, pct_pers
   from recs_match full join pers_match using(performer)
) LOOP null; END LOOP;
```

Carga de trabajo – Consulta 2:

```
FOR fila in (
WITH recordings as (select performer,tracks.title, writer,
                           min(rec_date) rec, 1 token
                    from albums join tracks using(pair)
                    group by performer,tracks.title,writer),
     playbacks as (select P.performer,
                       sum(token)*100/count('x') percentage,
                       avg(nvl2(rec,when-rec,rec)) as age
                    FROM performances P left join recordings R
                         on(P.performer=R.performer
                            AND R.title=P.songtitle
                            AND R.writer=P.songwriter
                            AND P.when>R.rec)
                    GROUP BY P.performer
                    ORDER BY percentage desc)
SELECT performer, percentage, floor(age/365.2422) years,
       floor(mod(age,365.2422)/30.43685) months,
       floor(mod(age,365.2422)-
(floor(mod(age,365.2422)/30.43685)*30.43685)) days
   FROM playbacks WHERE rownum<=10
) LOOP null; END LOOP;
```

Código para restablecer el estado inicial de la BD:

```
-- Operaciones (DML) para restablecer el contenido inicial
-- eliminamos físicamente el contenido de la tabla, y lo volvemos a crear
-- execute immediate 'TRUNCATE TABLE tracks';

INSERT INTO TRACKS (PAIR, sequ, title, writer, duration,
                    rec_date, studio, engineer, searchk)
  (SELECT DISTINCT album_pair, tracknum, min(song), min(writer),
                   min(duration), min(rec_date), min(studio),
                   min(engineer), album_pair||'//'||tracknum
     FROM fsdb.recordings
     WHERE album_pair is not null and tracknum is not null
           and song is not null and writer is not null
           and duration is not null and rec_date is not null
           and engineer is not null
           and not (song='Something jazzes' and writer='GB>>0785936179')
     GROUP BY album_pair,tracknum having count('s')=1
    );
COMMIT;
```