
Memoria diseño modelo relacional

Bases de Datos Avanzadas - UPM

Luis Mata bm0613, Javier Morate bm0620

05-04-2019

Contents

Memoria de trabajo sobre técnicas de aceleración de consultas	2
1. Introducción y forma de trabajo	2
2. Puntos pedidos	2
1. Se realizará la importación del esquema y datos de las cuatro tablas	2
2. Estudio de paneles de consulta e índices.	2
3. Optimización de consultas y estudio de planes de consulta	3
4. Estudio de índices en actualizaciones.	3
5. Desnormalización	4
6. Particionamiento.	4
3. Conclusiones	5

Memoria de trabajo sobre técnicas de aceleración de consultas

1. Introducción y forma de trabajo

2. Puntos pedidos

1. Se realizará la importación del esquema y datos de las cuatro tablas

Mediante la opción “Data Import” del menú Server de MySQLWorkbench con el fichero “dumpB-DAprac2.sql” disponible en el Moodle de la asignatura.

2. Estudio de paneles de consulta e índices.

a. Crear una consulta SQL que permita obtener el número de partidos jugados por cada jugador para aquellos jugadores de nacionalidad estadounidense (USA) o canadiense (CAN) que hayan jugado más partidos que la media del número de partidos jugados por todos los jugadores. La consulta devolverá el nombre y apellido del jugador y su edad actual, así como el número de partidos jugados, pero el resultado estará ordenado descendientemente por edad e a igual edad por apellido seguido de nombre pero ascendientemente.

b. Estudiar el plan de consulta, tomando nota de los costes del mismo y comentarlo.

c. Crear las claves principales y foráneas mediante los ficheros script CrearClavesPrimarias.sql y CrearClavesForeaneas.sql, y nuevamente estudiar el plan de consulta, comparando costes con el punto anterior.

d. Crear los índices que se estimen necesarios para mejorar la consulta.

e. Estudiar el plan de consulta con los nuevos índices y comparar resultados con los obtenidos en los puntos anteriores.

3. Optimización de consultas y estudio de planes de consulta

a. Eliminar los índices creados en el apartado anterior, manteniendo claves primarias y foráneas.

b. Definir en SQL al menos dos sentencias diferentes que permitan obtener los datos de los equipos que hayan jugado más partidos en los últimos seis meses del año 2017, devolviendo los atributos: identificador del equipo, nombre del equipo, número de partidos jugados, total de tiros (tshots) realizados en esos seis meses, media de goles del equipo por partido.

c. Crear los índices que permitan optimizar el coste de las consultas, analizando plan de consulta y coste para cada uno de los casos, justificando que solución es la más óptima.

4. Estudio de índices en actualizaciones.

a. Eliminar los índices creados en el apartado anterior manteniendo claves primarias y foráneas.

b. Crear un atributo en la tabla “player_stats” que se denomine “puntuación” que sea un número entero. El contenido de dicho atributo, para cada estadística de cada partido será un valor que se corresponderá con la suma de las cantidades de los atributos goals, shots y assists (es decir, en cada fila se sumarán los valores de estos atributos y se actualizará en el nuevo atributo “puntuación” con esa suma.) Actualizar la tabla player_stats para que contenga dicha información tomando nota de su tiempo de ejecución.

c. Volver a actualizar a null el atributo puntuación en todas las filas.

d. Por preverse un tipo de consultas con ordenación muy complicada, se desea crear un índice en la tabla “player_stats” por los atributos goals, shots, game_id, assists, player_id y puntuación, en ese orden. Crear el índice.

e. Volver a ejecutar la sentencia que establece los valores del atributo puntuación a la suma comentada anteriormente. Comprobar tiempo de ejecución y compararlo razonadamente con el contenido del punto 4.b.

5. Desnormalización

a. Eliminar los índices creados en el apartado anterior, manteniendo claves primarias y foráneas.

b. Crear una consulta que devuelva, para cada jugador su nombre, apellido, edad, total de asistencias (assists) y media de asistencias por partido, total de goles (goals) y media por partido para aquellos jugadores que tengan una edad entre 25 y 33 años.

c. Aplicar la técnica de desnormalización que se considere más adecuada para acelerar la consulta del apartado 5.b, creando los scripts sql necesarios para modificar el esquema de la base de datos.

d. Crear un script que actualice los datos implicados en la desnormalización.

e. Crear los triggers necesarios para mantener actualizados los datos implicados en la desnormalización.

f. Realizar la consulta 5.b sobre la base de datos desnormalizada. Estudiar coste y plan comparándolo con el obtenido en el apartado 5b.

6. Particionamiento.

a. Eliminar las claves foráneas con el script proporcionado “EliminarClavesForaneas.sql”.

b. Crear una consulta sql que obtenga, para cada jugador, su apellido, nombre, el número de partidos jugados y el número de jugadas realizadas por el jugador durante el año 2017. Estudiar coste y plan.

c. Razonar justificadamente (sin necesidad de implementarla realmente en SQL) una variante de la estructura existente realizando un particionamiento horizontal de los datos con el objeto de mejorar el tipo de consultas (con diferentes años) que se ha realizado en el apartado 6.a

d. Implementar en MySQL un particionamiento horizontal (mediante la sentencia ALTER TABLE PARTITION) que separe los datos de los partidos jugados en el año 2017 del resto. Realizar de nuevo la consulta 6.a y estudiar coste y plan comparándolo con lo obtenido en el apartado 6.a. Si se necesita modificar la clave primaria, hágase mediante la sentencia ALTER TABLE.

3. Conclusiones