

Bases de Datos

Memoria de prácticas - Gestión de la liga española

Grupo: Miércoles B

Hora: 17:00 - 19:00

Turno: Tardes, en todos los integrantes

***Integrantes: Jorge Aznar López, Ángel Cañal
Muniesa y Abel Chils Trabanco***

NIPS, en orden relativo: 721556, 716205, 718997

INDICE:

PARTE 1: CREACIÓN DE LA BASE DE DATOS

- | | |
|--|------------------|
| <i>1. Diseño del esquema E/R</i> | <i>Página 3</i> |
| <i>2. Decisiones de diseño tomadas</i> | <i>Página 4</i> |
| <i>3. Modelo relacional</i> | <i>Página 5</i> |
| <i>4. Tablas SQL</i> | <i>F.Adjunto</i> |

PARTE 2: POBLACIÓN DE LA BASE Y CONSULTAS REALIZADAS

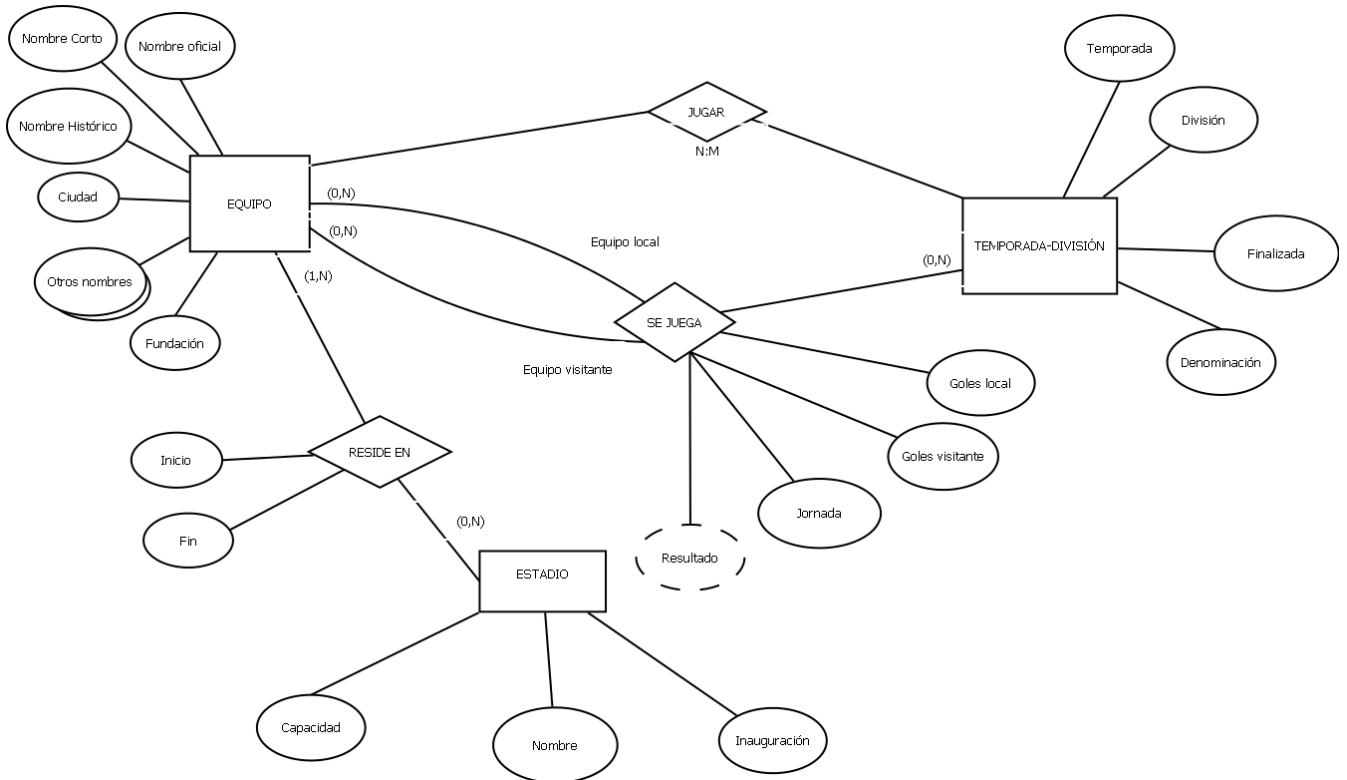
- | | |
|---|------------------|
| <i>1. Población de la base de datos</i> | <i>Página 8</i> |
| <i>2. Diseño y explicación de las consultas</i> | <i>Página 9</i> |
| ● <i>En álgebra relacional</i> | <i>Página 14</i> |
| <i>3. Tuplas cargadas y su tamaño</i> | <i>Página 22</i> |
| <i>4. Consultas SQL</i> | <i>F.Adjunto</i> |

PARTE 3: OPTIMIZACIÓN, DISEÑO FÍSICO Y TRIGGERS

- | | |
|---|------------------|
| <i>1. Diseño físico y mejora de eficiencia</i> | <i>Página 24</i> |
| <i>2. Explicación y justificación de los triggers creados</i> | <i>Página 26</i> |
| <i>4. Triggers SQL</i> | <i>F.Adjunto</i> |

PARTE 1: CREACIÓN DE LA BASE DE DATOS

ESQUEMA ENTIDAD-RELACIÓN



A la hora de diseñar el esquema entidad-relación que hará referencia a nuestra base de datos hemos decidido que para evitar introducir datos calculables a partir de otros el diseño se base en relaciones que permiten, en base a sus atributos, calcular el resto de datos necesarios tanto como para realizar las futuras consultas como para evitar sobrecargar de datos la base.

La creación de la relación jugar entre Equipo y Temporadas-División(N:M) se debe a que surgió la necesidad de conocer los equipos que participan en una temporada, no finalizada, que acaba de comenzar, pues de otra manera no podríamos disponer de esa información.

En la relación Se Juega existen dos ramas desde la entidad Equipo debido a que se debe diferenciar el equipo local del visitante, asimismo el

atributo Resultado de la relación es calculable debido a que con los goles del equipo local y los goles del equipo visitante podemos saber quién ha ganado el partido.

Relación Reside En, contemplar que en un estadio pueden entrenar varios equipos (filiales, etcétera) y además puede variar el equipo que reside allí, por ello existen los atributos Inicio y Fin, que indican este periodo.

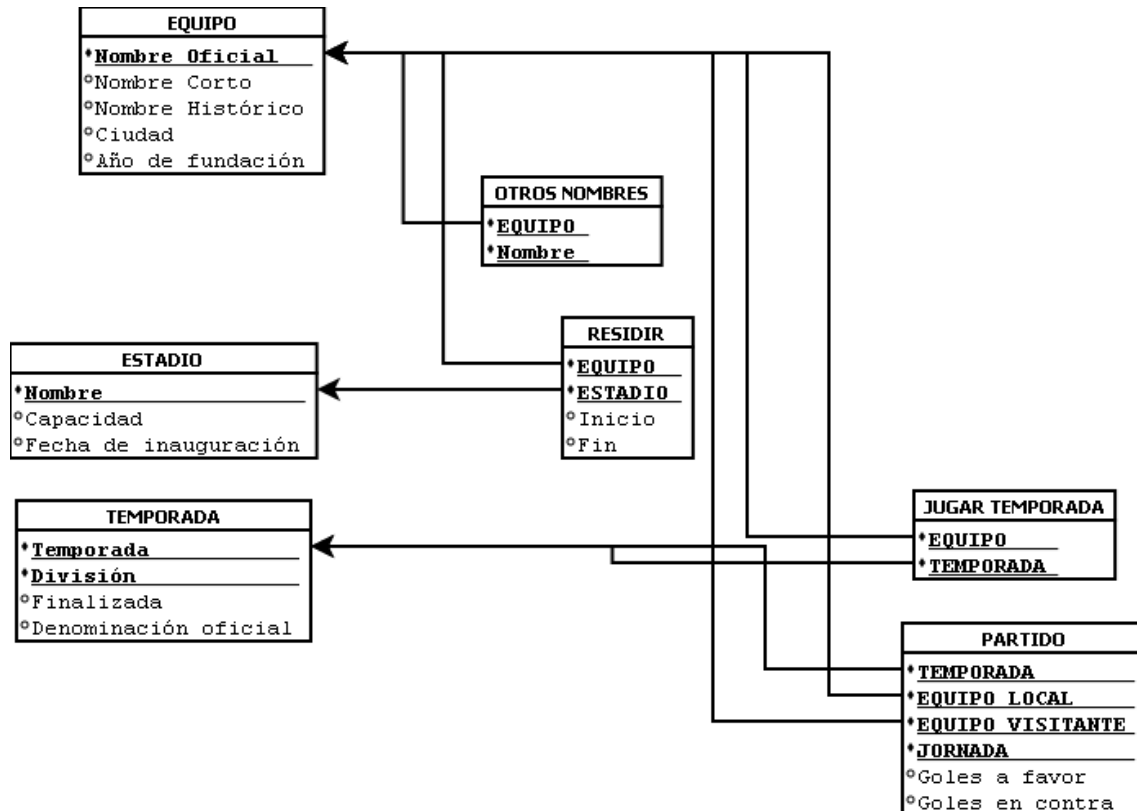
DECISIONES DE DISEÑO

Las principales dificultades en el proceso de diseño con las que nos hemos han sido las siguientes:

1. Hemos decidido no incluir el estadio en el que se disputa cada partido debido a que no disponemos de información fiable acerca de este dato, pues podría darse el caso de que un partido se tuviese que jugar en un estadio diferente al del equipo local debido a cualquier imprevisto.
2. Tampoco hemos incluido la fecha pues a pesar de que las jornadas son semanales un partido podría atrasarse o adelantarse debido a diversas causas (p.ej. meteorológicas).
3. Hemos decidido añadir la entidad "Temporada de División" debido a que es el único método que tenemos de asegurar que en caso de que una temporada acabe de comenzar se pueda acceder a los datos de los equipos que participan en ella, pues si no se dependería totalmente de los partidos disputados, que en caso de haber comenzado la temporada no habría ninguno.

Y estos datos no se podría disponer de manera fiable a partir de los datos contenidos en la temporada anterior, ya que los equipos pueden disolverse o ser descalificados. Por otro lado, tampoco podríamos saber los equipos que jugarán en segunda división debido a que no se dispone de datos de segunda división B.

MODELO RELACIONAL Y PROCESO DE NORMALIZACIÓN



1ª Forma normal: Debido a la existencia del atributo multi-evaluado "Otros nombres" en la entidad equipo se creó la tabla Otros Nombres con clave primaria compuesta por Equipo, que referencia a la tabla Equipo, y Nombre que es propia de la relación.

2ª Forma normal: El modelo relacional se hallaba en segunda forma normal cuando fue analizado, dado que para todo atributo clave A no existía dependencia funcional con cualquier otro atributo también clave B, es decir, ningún atributo A clave depende funcionalmente de parte de la clave. No fue necesario analizar las relaciones con clave única, sólo aquellas con clave compuesta (varias claves candidata).

3ª Forma normal: El modelo relacional se hallaba en tercera forma normal cuando fue analizado, dado que para todo atributo clave A no existía dependencia funcional con cualquier otro atributo, o conjunto de atributos, no clave B, es decir, ningún atributo A clave depende funcionalmente de otro atributo de la relación o conjunto de atributos no clave. No fue necesario analizar las relaciones con clave única, sólo aquellas con uno o varios atributos no clave.

4ª Forma normal: El modelo relacional se hallaba en cuarta forma normal cuando fue analizado, dado que no existen dependencias multi-valoradas entre cualquier conjunto de atributos A y otro conjunto de atributos B, es decir, los valores de B sí que dependen de los valores que tome A en un caso determinado.

ANEXO I: GESTIÓN DE PROYECTO

Alumno	Horas invertidas	Concepto
Jorge	2	Desarrollo esquema E/R
	2	Diseño del esquema E/R y del modelo relacional
	3	Desarrollo de la memoria y la presentación del trabajo
Abel	2	Desarrollo esquema E/R
	2,5	Diseño del esquema E/R y del modelo relacional
	0,5	Desarrollo de la memoria y la presentación del trabajo
Ángel	2	Desarrollo esquema E/R
	2	Diseño del esquema E/R y del modelo relacional

PARTE 2: POBLACIÓN DE LA BASE Y CONSULTAS

REALIZADAS

POBLACIÓN DE LA BASE DE DATOS

Para realizar la tarea de poblar la base de datos tras la creación de tablas (fichero adjunto) se tuvo que afrontar los siguientes problemas previamente a la población de la base:

A partir de los enlaces que se nos proporcionaron, se obtuvieron los datos ahí contenidos. Luego se transformaron a formato csv separado por comas mediante el uso de expresiones regulares en Sublime Text (editor de texto) y se normalizaron los datos para evitar conflictos entre conjuntos de datos, modificando los elementos con nombres diferentes en los distintos conjuntos.

Posteriormente mediante Excel se trabajó sobre estos conjuntos de datos para poder crear conjuntos que se pudiesen insertar en la base de datos.

En la tabla Otros Nombres no se pudo insertar ningún dato debido a que no se encontró ninguna fuente fiable de donde obtener estos datos.

Una vez obtenidos los ficheros con los conjuntos de datos, estos fueron introducidos a la Base de Datos Oracle mediante el IDE Datagrip (JetBrains), orientado al uso de bases de datos. Este IDE permite insertar datos contenidos en un fichero csv en la base de datos.

DECISIONES DE DISEÑO DE LAS CONSULTAS

Para resolver las consultas se decidió valerse de la creación de vistas, en las cuales cada una resuelve una parte del problema y apoyándose en las anteriores se logra resolver la consulta pedida. Esto permite tanto reutilizar vistas como separar de una manera más clara los elementos que resuelven la consulta.

EXPLICACIÓN DE LAS CONSULTAS

En este apartado se explica el funcionamiento de cada una de las vistas utilizadas para resolver las consultas. Las cuales están contenidas en consultas.sql.

Primera consulta obligatoria:

Descripción: Lista el/los equipo(s) que ha tenido el mayor número de empates cada temporada y el número de empates

VIEW TEMP_1_2_DIV: Obtiene las temporadas finalizadas que representan 1ª o 2ª división.

VIEW empates: Obtiene por cada equipo todos los empates que ha tenido en 1ª o 2ª división.

VIEW TOTAL_EMPATES: Obtiene los empates por equipo y temporada.

VIEW MAX_EMPATES_TEMPORADA: Obtiene el mayor número de empates logrado por algún equipo cada temporada.

VIEW EQUIPO_MAX_EMPATES_TEMPORADA: Al resultado de la vista MAX_EMPATES_TEMPORADA se le asocia el equipo o los equipos a los que corresponde ese número de empates.

RESULTADO PRIMERA CONSULTA: Se transforma el identificador de temporada en el año y la división asociada.

Segunda consulta obligatoria:

Descripción: Obtiene los equipos que en alguna temporada han tenido una media de goles por partido mayor o igual a 3 y las temporadas en las que eso ha ocurrido

VIEW equipos_goles_partido: Obtiene los goles que ha marcado cada equipo por partido en 1ª o 2ª división.

VIEW mas_de_3_goles_temporada: Se agrupa el resultado anterior por equipo y temporada y se obtiene su media de puntos, descartando aquellos en los que esta es inferior a 3

RESULTADO SEGUNDA CONSULTA: Se transforma el identificador de temporada en el año y la división asociada.

Tercera consulta obligatoria:

Descripción: Lista los 3 equipos con mejor palmarés (más veces han quedado primeros) en primera división e indica cuántas veces han quedado primeros cada uno de ellos.

VIEW TEMP_1_DIV: Obtiene las temporadas que representan la 1ª división.

VIEW PARTIDOS_GANADOS_1_DIV: Obtiene por cada equipo y temporada los partidos que este ganó.

VIEW PARTIDOS_EMPATADOS_1_DIV: Obtiene por cada equipo y temporada los partidos que este empató.

VIEW PUNTOS_EMPATES_VICT_1_DIV: Se obtienen por separado los puntos obtenidos en victorias y en empates

VIEW PUNTOS_TEMPORADA: Se suman los puntos obtenidos en victorias y empates, por lo que se obtiene la puntuación final de cada equipo en cada temporada.

VIEW MAX_PUNTOS_TEMPORADA: Obtiene la mayor puntuación lograda cada temporada.

VIEW GANADORES_TEMPORADAS: Asocia al equipo correspondiente que obtuvo cada puntuación en MAX_PUNTOS_TEMPORADA.

VIEW PALMARES: Obtiene las temporadas ganadas por cada equipo y las ordena en orden decreciente.

RESULTADO TERCERA CONSULTA: Selecciona los 3 equipos con mejor palmarés.

Primera consulta personal:

Descripción: Obtiene el/los equipo(s) que obtuvieron la mayor diferencia de goles (goles anotados - goles recibidos) de cada temporada así como esta diferencia.

VIEW diferencia_de_goles_parcial: Obtiene la diferencia de goles en casa y fuera por separado cada temporada.

VIEW diferencia_de_goles: Obtiene la diferencia de goles de cada equipo cada temporada.

VIEW max_diferencia_de_goles: Obtiene la mayor diferencia de goles lograda por algún equipo cada temporada

VIEW equi_max_dif_de_goles_temp: Obtiene al equipo asociado a cada diferencia de goles obtenida en max_diferencia_de_goles.

RESULTADO PRIMERA CONSULTA PERSONAL: Se transforma el identificador de temporada en el año y la división asociada.

Segunda consulta personal:

Descripción: Obtiene los años en los que se han marcado más goles en primera división que en segunda.

VIEW GOLES_POR_PARTIDO: Obtiene los goles marcados en cada partido.

VIEW CORRESPONDENCIA_1_2: Obtiene la correspondencia de los identificadores de temporada que representan la 1ª división y la 2ª división del mismo año.

VIEW GOLES_POR_TEMPORADA: Obtiene los goles por temporada.

RESULTADO SEGUNDA CONSULTA PERSONAL: Devuelve los años asociados a las temporadas en las cuales se marcó más goles en 1ª división que en 2ª división.

Tercera consulta personal:

Descripción: Obtiene la(s) temporada(s) de primera división en la cual más equipos empataron a puntos, así como estos equipos y los puntos que consiguieron.

VIEW MISMA_PUNT_FINAL: Basándose en la puntuación final de cada temporada, calculado en la tercera consulta, se obtiene por cada puntuación el número de equipos que la obtuvieron.

VIEW NUM_MAX_EQUIPOS_EMPATADOS: Se obtiene por cada temporada el máximo número de empates.

VIEW MAX_EQUIPOS_TEMP_PUNT_EMP: Obtiene los puntos asociados al máximo empate.

VIEW EQUIPOSEMP_IDTEMP_PUNTOS: Obtiene los equipos que asociados a los resultados de VIEW MAX_EQUIPOS_TEMP_PUNT_EMP.

RESULTADO TERCERA CONSULTA PERSONAL: Se transforma el identificador de temporada en el año y la división asociada.

Consultas en álgebra formal

PRIMERA CONSULTA OBLIGATORIA

TEMP_1_2_DIV

$(\pi_{id}) - (\sigma_{(división = 1 \text{ or } división = 2) \text{ and } finalizada=1}) - (TEMPORADA)$

EMPATES

$(\pi_{idtemp, equipo_local \text{ as } equi}) - (\sigma_{(goles_local = goles_visitante) \text{ and exists } ((\pi_{idtemp}) - (\sigma_{idtemp = id}) - TEMP_1_2_DIV)) , (PARTIDO)$

U

$(\pi_{idtemp, equipo_visitante \text{ as } equi}) - (\sigma_{(goles_local = goles_visitante) \text{ and exists } ((\pi_{idtemp}) - (\sigma_{idtemp = id}) - TEMP_1_2_DIV)) , (PARTIDO)$

TOTAL_EMPATES

$((temp, equi) \text{ f count(*) as emp}) - (EMPATES)$

MAX_EMPATES_TEMPORADA

$((idtemp) \text{ f max(emp) as empates_totales}) - (TOTAL_EMPATES)$

EQUIPO_MAX_EMPATES_TEMPORADA

$(\pi_{equi, total_EMPATES.idtemp \text{ as } temp, MAX_EMPATES_TEMPORADA.empates_totales \text{ as } num_empates}) - (\sigma_{MAX_EMPATES_TEMPORADA.idtemp = TOTAL_EMPATES.idtemp \text{ AND } MAX_EMPATES_TEMPORADA.empates_totales = TOTAL_EMPATES.emp}) - (TOTAL_EMPATES) , (MAX_EMPATES_TEMPORADA)$

RESULTADO PRIMERA CONSULTA OBLIGATORIA

$(\pi_{TEMPORADA.anyo, TEMPORADA.division, equi, num_empates}) - (\sigma_{EQUIPO_MAX_EMPATES_TEMPORADA.temp = TEMPORADA.id}) - (EQUIPO_MAX_EMPATES_TEMPORADA), (TEMPORADA)$

SEGUNDA CONSULTA OBLIGATORIA

EQUIPOS_GOLES_PARTIDO

$(\rho_{goles_local/goles, idTemp/temporada_equipo_local/equipo_}(\pi_{goles_local, idTemp, equipo_local}(\sigma_{EXISTS((1))(partido)))) \cup (\rho_{goles_visitante/goles, idTemp/temporada_equipo_visitante/equipo_}(\pi_{goles_visitante, idTemp, equipo_visitante}(\sigma_{EXISTS((1))(partido))))$
 $(1) \Rightarrow \sigma_{idTemp = id(TEMP_1_2_DIV)}$

MAS_DE_3_GOLE_TEMPORADA

$\pi_{equipo_temporada_}(\sigma_{AVR(goles) \geq 3}((temporada_equipo_)\zeta_{AVR(goles)}(equipos_goles_partido))))$

RESULTADO SEGUNDA CONSULTA OBLIGATORIA

$\pi_{temporada.anyo, temporada.division, equipo_}((mas_de_3_goles_temporada) \bowtie_{temporada=temporada.id} (temporada))$

TERCERA CONSULTA OBLIGATORIA

TEMP_1_DIV

$(\pi_{id}) - (\sigma(\text{división} = 1) \text{ and } \text{finalizada} = 1) - (\text{TEMPORADA})$

PARTIDOS_GANADOS_1_DIV

$(\pi_{idtemp, equipo_local \text{ as } equi}) - (\sigma(\text{goles_local} > \text{goles_visitante}) \text{ and exists } ((\pi^*) - (\sigma_{idtemp = id}) - \text{TEMP_1_DIV})) ,$

(PARTIDO)

U

$(\pi_{idtemp, equipo_visitante \text{ as } equi}) - (\sigma(\text{goles_local} < \text{goles_visitante}) \text{ and exists } ((\pi^*) - (\sigma_{idtemp = id}) - \text{TEMP_1_DIV})) ,$

(PARTIDO)

PARTIDOS_EMPATADOS_1_DIV

$(\pi_{idtemp, equipo_local \text{ as } equi}) - (\sigma(\text{goles_local} = \text{goles_visitante}) \text{ and exists } ((\pi^*) - (\sigma_{idtemp = id}) - \text{TEMP_1_DIV})) ,$

(PARTIDO)

U

$(\pi_{idtemp, equipo_visitante \text{ as } equi}) - (\sigma(\text{goles_local} = \text{goles_visitante}) \text{ and exists } ((\pi^*) - (\sigma_{idtemp = id}) - \text{TEMP_1_DIV})) ,$

(PARTIDO)

PUNTOS_EMPATES_VICT_1_DIV

$((\text{idtemp}, \text{equi}) \text{f}_{\text{count}(*)} \text{as puntos}) - (\text{PARTIDOS_EMPATADOS_1_DIV})$

U

$((\text{idtemp}, \text{equi}) \text{f}_{\text{count}(*)*3} \text{as puntos}) - (\text{PARTIDOS_GANADOS_1_DIV})$

PUNTOS_TEMPORADA

$((\text{idtemp}, \text{equi}) \text{f}_{\text{sum}(\text{puntos})} \text{as puntos}) - (\text{PUNTOS_EMPATES_VICT_1_DIV})$

MAX_PUNTOS_TEMPORADA

$((\text{idtemp}) \text{f}_{\text{max}(\text{puntos})} \text{as puntos}) - (\text{PUNTOS_TEMPORADA})$

GANADORES_TEMPORADAS

$(\pi_{\text{equi}, \text{MAX_PUNTOS_TEMPORADA.IDTEMP AS temp}, \text{MAX_PUNTOS_TEMPORADA.PUNTOS AS puntos}) - (\sigma_{\text{PUNTOS_TEMPORADA.IDTEMP = temp AND PUNTOS_TEMPORADA.PUNTOS = puntos}) -$

$(\text{PUNTOS_TEMPORADA}) , (\text{MAX_PUNTOS_TEMPORADA})$

PALMARES

$((\text{equi}) \text{f}_{\text{count}(\text{equi})} \text{as TITULOS}) - (\text{GANADORES_TEMPORADA})$

RESULTADO TERCERA CONSULTA OBLIGATORIA

$(\pi_{*}) - (\sigma_{\text{ROWNUM} \leq 3}) - (\text{PALMARES})$

PRIMERA CONSULTA PERSONAL

DIFERENCIA_DE_GOLES_PARCIAL

$(\pi_{(idtemp, equipo_local \text{ as } equi)} \text{ f sum(goles_visitante - goles_local) as goles }) - (\sigma_{\text{exists(} (\pi^*) - (\sigma_{idtemp = id}) - (TEMP_1_2_DIV))) - (PARTIDO)$

U

$(\pi_{(idtemp, equipo_visitante \text{ as } equi)} \text{ f sum(goles_local - goles_visitante) as goles }) - (\sigma_{\text{exists(} (\pi^*) - (\sigma_{idtemp = id}) - (TEMP_1_2_DIV))) - (PARTIDO)$

DIFERENCIA_DE_GOLES

$(\pi_{(idtemp, equi)} \text{ f sum(goles) as goles }) - (DIFERENCIA_DE_GOLES_PARCIAL)$

MAX_DIFERENCIA_DE_GOLES

$(\pi_{(idtemp)} \text{ f max(goles) as max_diferencia }) - (DIFERENCIA_DE_GOLES)$

EQUI_MAX_DIF_DE_GOLES_TEMP

$(\pi_{equi, MAX_DIFERENCIA_DE_GOLES.max_diferencia \text{ as } diferencia, MAX_DIFERENCIA_DE_GOLES.idtemp \text{ as } temp}) - (\sigma_{MAX_DIFERENCIA_DE_GOLES.idtemp = DIFERENCIA_DE_GOLES.idtemp \text{ and } MAX_DIFERENCIA_DE_GOLES.max_diferencia = DIFERENCIA_DE_GOLES.goles }) - (MAX_DIFERENCIA_DE_GOLES, DIFERENCIA_DE_GOLES)$

RESULTADO PRIMERA CONSULTA PERSONAL

$(\pi_{TEMPORADA.anyo, TEMPORADA.division, equi, diferencia}) - (\sigma_{EQUI_MAX_DIF_DE_GOLES_TEMP.temp = TEMPORADA.id}) - (EQUI_MAX_DIF_DE_GOLES_TEMP, TEMPORADA)$

SEGUNDA CONSULTA PERSONAL

GOLES_POR_PARTIDO

$$\rho_{goles_visitante + partido_goles_local/goles}(\pi_{goles_visitante + partido_goles_local}(\sigma_{EXISTS((1))}(pa)))$$
$$(1) \Rightarrow \sigma_{idTemp = id}(TEMP_1_2_DIV)$$

GOLES POR TEMPORADA

$$\rho_{SUM(goies)/goies}(\pi_{idTemp, SUM(goies)}((idTemp) \zeta_{SUM(goies)}(goies_por_partido)))$$

CORRESPONDENCIA_1_2

$$\rho_{t1.id/id_1, t2.id/id_2}(\pi_{t1.id, t2.id}((t1) \bowtie_C (t2)))$$
$$t1 \Rightarrow \rho_{temporada/t1}(temporada)$$
$$t2 \Rightarrow \rho_{temporada/t2}(temporada)$$
$$C \Rightarrow t1.anyo = t2.anyo \wedge t1.division = '1' \wedge t2.division = '2'$$

RESULTADO SEGUNDA CONSULTA PERSONAL

$$\rho_{temporada.anyo/anyo, t1.goies - t2.goies/diff}(\pi_{temporada.anyo, t1.goies - t2.goies}((UNIONS)))$$
$$UNIONS = (c1) \bowtie_{C1} ((t2) \bowtie_{C2} ((t1) \bowtie_{C3} (temporada)))$$
$$C1 \Rightarrow (t1.idTemp = c1.id_1 \wedge t2.idTemp = c1.id_2)$$
$$C2 \Rightarrow (t1.goies > t2.goies)$$
$$C3 \Rightarrow (t1.idTemp = temporada.id)$$
$$t1 \Rightarrow (\rho_{goies_por_temporada/t1}(puntos_por_temporada))$$
$$t2 \Rightarrow (\rho_{goies_por_temporada/t2}(puntos_por_temporada))$$

$$c1 \Rightarrow (\rho_{correspondencia_1_2/c1}(correspondencia_1_2))$$

TERCERA CONSULTA PERSONAL

MISMA_PUNT_FINAL

$(\pi_{idtemp, puntos} f_{count(*)} \text{ as EQUIPOS_EMPATADOS }) - (PUNTOS_TEMPORADA)$

NUM_MAX_EQUIPOS_EMPATADOS

$(f_{max(EQUIPOS_EMPATADOS) \text{ as NUM_EQUI }}) - (MISMA_PUNT_FINAL)$

MAX_EQUIPOS_TEMP_PUNT_EMP

$(\pi_{idtemp, puntos}) - (\sigma_{NUM_MAX_EQUIPOS_EMPATADOS.NUM_EQUI = EQUIPOS_EMPATADOS}) - (MISMA_PUNT_FINAL), (NUM_MAX_EQUIPOS_EMPATADOS)$

EQUIPOSEMP_IDTEMP_PUNTOS

$(\pi_{PUNTOS_TEMPORADA.EQUI \text{ AS EQUI}, PUNTOS_TEMPORADA.IDTEMP \text{ AS IDTEMP}, PUNTOS_TEMPORADA.PUNTOS \text{ AS PUNTOS}}) -$

$(\sigma_{PUNTOS_TEMPORADA.IDTEMP = MAX_EQUIPOS_TEMP_PUNT_EMPT.IDTEMP \text{ AND } PUNTOS_TEMPORADA.PUNTOS = MAX_EQUIPOS_TEMP_PUNT_EMP.PUNTOS}) - (PUNTOS_TEMPORADA), (MAX_EQUIPOS_TEMP_PUNT_EMP)$

RESULTADO TERCERA CONSULTA OBLIGATORIA

$(\pi_{\text{TEMPORADA.anyo,}}$
 $\text{equi, puntos}) - (\sigma_{\text{EQUIPOSEMP_IDTEMP_PUNTOS.IDTEMP}}$
 $= \text{TEMPORADA.id}) -$
 $(\text{EQUIPOSEMP_IDTEMP_PUNTOS, TEMPORADA})$

Tamaño de tablas

Tabla	Tamaño (bytes)	Tuplas
Empates	24234	(1, Granada, 11) (1, Zaragoza, 14) (1, Celta, 9) (1, Espanyol, 11) (1, Málaga (C.D.), 11)
Equipo	1272	(Alavés, NULL, NULL, NULL, 1921) (Albacete, NULL, NULL, NULL, 1940) (Alcorcón, NULL, NULL, NULL, 1971) (Alcoyano, NULL, NULL, NULL, 1928) (Algeciras, NULL, NULL, NULL, 1912)
Estadio	1783	(Santo Domingo, 5000, 1978) (Carlos Belmonte, 17400, 1960) (Nuevo Mirador, 7500, 1999) (San Mamés, 53289, 2013) (Lasesarre, 7960, 2001)
Media_goles	28479	(Granada, 1, 25, 34) (Zaragoza, 1, 36, 34) (Barcelona, 1, 41, 34) (Dptivo. Coruña, 1, 22, 34) (At. Madrid, 1, 49, 34)
Partido	887962	(1, At. Madrid, Valencia, 1, 1, 3) (1, Barcelona, Dptivo. Coruña, 1, 3, 1) (1, Burgos, Real Sociedad, 1, 1, 0) (1, Castellón, Real Madrid, 1, 2, 3) (1, Celta, Espanyol, 1, 1, 1)
Residir	2181	(Alcorcón,Santo Domingo) (Albacete,Carlos Belmonte) (Algeciras,Nuevo Mirador) (Ath. Bilbao,San Mamés) (Barakaldo,Lasesarre)
Temporada	1571	(1, 1972-1973, 1, 1) (2, 1972-1973, 2, 1) (3, 1973-1974, 1, 1) (4, 1973-1974, 2, 1) (5, 1974-1975, 1, 1)
Total	947482	

ANEXO II: GESTIÓN DE PROYECTO

Alumno	Horas invertidas	Concepto
Jorge	2	Desarrollo consultas obligatorias/personales
	2	Obtención de datos
	2	Traducción a notación algebraica de las consultas 3ª en SQL
	3	Desarrollo de la memoria y la presentación del trabajo
Abel	5	Desarrollo consultas obligatorias/personales
	3	Obtención de datos
	1	Traducción a notación algebraica de las consultas 1ª en SQL
	3	Desarrollo de la memoria y la presentación del trabajo
Ángel	2	Desarrollo consultas obligatorias/personales
	2,5	Obtención de datos
	1,5	Traducción a notación algebraica de las consultas 2ª en SQL

PARTE 3: OPTIMIZACIÓN, DISEÑO FÍSICO Y TRIGGERS

Diseño físico

Consulta obligatoria nº1

En el plan de ejecución de esta consulta se ve que realiza hasta 3 joins y 5 ordenaciones. Además, se ve que se usan los índices de las claves primarias, por lo que añadir índices aquí no cambiará la consulta. Se puede mejorar el acceso a la tabla partido, ya que se accede 4 veces a toda la tabla, y esto puede mejorarlo un trigger.

Consulta obligatoria nº2

En el plan de ejecución se observa que se realizan 5 accesos a tablas, 2 accesos a toda la tabla “*partido*” y 3 a la tabla “*temporada*”, por lo que se mejorará mediante un trigger que evite la mayoría de esos accesos.

Consulta obligatoria nº3

En el plan de ejecución se ve que se accede dos veces a la vista “*puntos_temporada*”, lo que desencadena por duplicado un proceso bastante pesado, y que podría ser causante de la mayoría del tiempo consumido en ejecución.

Tiempo de ejecución

Consultas obligatorias	T(s) sin optimizar	T(s) con índices	T(s) con triggers
Consulta 1	1.140 s	1.150 s	0.740 s
Consulta 2	1.240 s	1.220 s	0.620 s
Consulta 3	1.290 s	1.290 s	-

Consultas nuestras	T(s) sin optimizar	T(s) con índices	T(s) con triggers
Consulta 1	1.800 s	1.780 s	-
Consulta 2	1.050 s	1.030 s	-
Consulta 3	1.640 s	1.630 s	-

De la tabla anterior podemos deducir que los índices que hemos añadido no han tenido apenas efecto, ya que las consultas no hacen uso de esos índices.

Como caso especial se encuentra la 2ª consulta nuestra, que sí que usa el índice *"div_index"* de la columna *"division"* en la tabla *"temporada"*, y por lo tanto se ha visto reducido su tiempo en 0.02 segundos.

Una vez hemos introducido dos triggers para reducir el tiempo de las dos primeras consultas vemos que efectivamente disminuye, en 0.41 segundos y 0.6 segundos respectivamente.

Tomando como ejemplo el trigger usado para la segunda consulta, que permite reducir a la mitad el tiempo de ejecución de la misma, se ve que de hacer 50 llamadas recursivas se pasa a apenas 7. Además, el número de bloques leídos de memoria cae de 895 a sólo 21 (más de un 95%).

TRIGGERS

La primera optimización que se ha creado corresponde al create table empates y el trigger empates (insert, update y delete) el cual se encuentra en el fichero triggers.sql. Esta optimización almacena en la tabla empates el número de empates por temporada y equipo. Esto se consigue, al insertar un nuevo partido, aumentando en una unidad el número de empates de cada uno de los equipos insertados si el número de goles de ambos equipos es el mismo.

Con esto se reduce el tiempo de obtención de la primera consulta obligatoria (obtener equipos con más empates cada temporada), así como la tercera consulta obligatoria (a la hora de obtener puntos por empates) y la tercera consulta opcional (a la hora de obtener puntos por empates).

La segunda optimización que se ha creado corresponde al create table media_goles y el trigger media_goles (insert, update y delete) el cual se encuentra en el fichero triggers.sql. Esta optimización almacena en la tabla media_goles el número de goles obtenido por cada equipo en cada temporada, así como el número de partidos jugados por cada uno de ellos. De esta forma, obtener la media de puntos consiste en dividir el número de goles entre el número de partidos jugados.

Esto se consigue, al insertar un nuevo partido, sumando el número de goles marcado por cada equipo al número de goles totales de ese equipo en esa temporada y aumentando en una unidad el número de partidos jugados por ese equipo en esa temporada.

Esto es útil en la segunda consulta obligatoria.

Por último, se ha creado un trigger llamado overlap (insert, update) el cual se encuentra en el fichero triggers.sql y ayuda a mantener la consistencia en la base de datos impidiendo que en la tabla residir se establezca que un equipo dejó de residir en un estadio antes de empezar a residir en él. Esto se consigue comprobando la fecha de comienzo de la residencia y fin de esta y lanzando una excepción en el caso de que la segunda sea inferior a la primera.

ANEXO III: GESTIÓN DE PROYECTO

Alumno	Horas invertidas	Concepto
Abel	2,5	Desarrollo de los triggers
	2	Desarrollo de la memoria y la presentación del trabajo
Ángel	5	Desarrollo de los triggers
	3	Análisis de datos (diseño físico)
	1	Desarrollo de la memoria y la presentación del trabajo
Jorge	1	Desarrollo de la memoria y la presentación del trabajo