

## Trabajo 2

- Este es el segundo trabajo a realizar, que trata sobre modelado E/R y consultas SQL avanzadas.
- En la guía docente se indica que este trabajo serán individuales o por parejas, pero se admitirán entregas de hasta 3 alumnos, para los que lo prefieran. En principio, los grupos serán los mismos que para el trabajo 1, pues hay apartados del trabajo 2 que son continuación del trabajo 1. Sin embargo, en caso de que no podáis o no queráis seguir en el mismo grupo, podéis separaros del compañero, pero no irs a otro grupo. Tenedme informado por mail lo antes posible en ese caso.
  - Si no has hecho el trabajo 1, puedes intentar la parte de E/R, pero lo más lógico es que empieces ya con la entrega en segunda convocatoria del trabajo 1. El enunciado es el mismo.

## Parte E/R

Crear el diagrama E/R correspondiente al siguiente enunciado aplicando el mayor número posible de conceptos vistos en el material de la asignatura

Se pretende hacer una web con información de partidos de fútbol. Cada partido enfrenta a 2 equipos, uno juega de local, y otro de visitante, y tiene una fecha<sup>1</sup> y un resultado. Los equipos pueden ser clubes o selecciones nacionales. Cada equipo tiene un nombre que le distingue de todos los demás, pero también un nombre corto único que permite identificarlos más fácilmente (p.e. RMAD). Todo equipo es de un determinado país, pero la diferencia entre selecciones y clubes, es que los clubes son de una determinada ciudad, a través de la cual se puede inferir el país. De los países solo se registrará su nombre y un nombre corto, (p.e. ESP).

A efectos de las explicaciones que siguen, dividiremos los partidos en amistosos vs. de competición. Algunas competiciones dividen sus partidos en jornadas (por ejemplo en la liga cada semana se suele corresponder con una jornada) o en fases (liguilla, octavos, cuartos, semis, final..). Salvo en el caso de los partidos amistosos, todos los partidos de competición deberán de tener asignada en su caso, su jornada o su fase.

Por tanto, a un partido de competición le corresponde o una jornada o una fase. Cada jornada/fase se corresponde con una combinación temporada-competición. Las temporadas no tienen más información que el año en el que comienzan. Conviene tratarlas como una entidad para evitar que nadie pueda registrar datos correspondientes a temporadas que no hayan sido dadas de alta previamente en la base de datos.

Las competiciones, sin embargo además del nombre, han de registrar si son competiciones de clubes o de selecciones nacionales, y si son masculinas o femeninas. En el caso de ser de clubes, han de registrar si son nacionales o internacionales. También para el caso de las competiciones de clubes, se debe de registrar si una es una versión moderna de otra anterior, (p.e. Champions League es la versión moderna de la Copa de Europa, o la Europa League antes era la copa de la UEFA, la Copa del Rey también tenía otro nombre antes etc...)

Los amistosos no registran toda esa información, pues en todo caso se puede asimilar el dato de la temporada deduciéndolo de la fecha del partido.

En cuanto a los árbitros, los partidos han de tener asociados obligatoriamente un equipo arbitral. Cada equipo estará formado al menos por un árbitro principal y 2 asistentes, pero puede haber otros miembros según qué competición sea. Deberá de registrarse el papel del

---

<sup>1</sup> En el caso de partidos que se suspendan a la mitad, la fecha del partido es la de inicio del mismo.

árbitro en el partido (si es árbitro principal, asistente, VAR ... ). Como el mundo del fútbol evoluciona, la lista de posibles roles arbitrales ha de poder cambiar, por lo que se recomienda estén registrados en una tabla aparte. Cada árbitro tiene un id que lo identifica, un nombre y una nacionalidad.

Se deben de registrar en qué combinaciones temporada-competición participa o participó cada equipo.

Los equipos inscriben a sus jugadores en cada combinación temporada-competición en la que participan. Aunque un jugador tenga ficha en un equipo, puede que no esté inscrito en todas las competiciones de ese equipo para esa temporada por limitaciones de cada competición (límite de extranjeros, de jugadores, que ya haya jugado la misma temporada la competición con otro equipo etc ... ). Por tanto, tenemos que registrar la fecha de alta y baja de cada jugador en en su equipo para esa combinación competición-temporada.

De los jugadores, se registra un id, un nombre deportivo, el nombre completo, estatura, peso, importe de su ficha actual, y fecha en la que termina su contrato actual.

Para cada partido se registran cada uno de los lances de juego. Cada lance ocurre en un minuto y segundo de un periodo, y tiene un texto que se corresponde con un comentario descriptivo que hace el comentarista deportivo de la web. Normalmente, hay 2 periodos: 1ª parte y 2ª parte, pero también se considera que la prórroga, cuando la hay, puede generar otros 2 periodos. Cada lance está asociado un único partido, y le puede corresponder un solo jugador, o un solo árbitro. Cada lance es de un solo tipo. Hay múltiples tipos de lance: estar en la alineación inicial, ser sustituido, marcar un gol, lesionarse, pitar fuera de juego, pitar fin de partido, sancionar con tarjeta, suspender el partido, reanudarlo etc ... Para poder representar todos estos casos, debe haber una tabla aparte con un id\_lance y la descripción del mismo, de manera que evitemos que cada usuario se refiera al mismo lance de distintas maneras, y a la vez permitamos que esa lista de lances vaya creciendo si lo necesitamos más adelante

1. Se pide que, mientras no sea de manera forzada/inverosímil, utilices todos los mecanismos posibles de entre los explicados en los materiales para representar el problema en un modelo E/R, de manera que recoja todos los requisitos que puedas de entre los presentes en el enunciado. Es decir, si encuentras la forma de hacerlo con dos ISAs, mejor dos que una, si encuentras ternarias o agregaciones, cuantas más mejor, etc ... pero cuida que realmente sean correctas.

Evita simplificar el modelo reduciéndolo a relaciones binarias que al final, si que es verdad que siendo más simples se entienden mejor, pero no pueden representar tantos requisitos como otros elementos avanzados del diagrama E/R.

## 2. ENTREGABLES

1. **Deberás de entregar un word o PDF con el diagrama.** Puede contener comentarios explicativos de las decisiones de diseño tomadas, o incluyendo el histórico de versiones en el que ha ido evolucionando hasta llegar a la versión definitiva. En ese histórico, justifica por qué descartas la versión anterior, y qué ventajas aporta la siguiente. Las versiones iniciales pueden ser fotos de una hoja con esbozos a bolígrafo, no os compliquéis.
2. **Entregar un script con los create tables correspondientes para representar ese modelo E/R.** En el caso de haber relaciones ISA, proponer las posibles soluciones, dejar entre comentarios las soluciones que descartas, y explicar por qué ese descarte o por qué esa preferencia por una determinada solución.
3. Esta vez no hay vídeos porque en el examen oral, en la parte de E/R te preguntaré preferentemente cosas de este trabajo. Si no entregas el trabajo, o lo entregas muy mal,

tendré que buscar otras preguntas.

## Parte Consultas SQL

Para esta parte utiliza las tablas del trabajo 1 (aunque aún no se haya corregido dicho trabajo)

- Si necesitas en algún caso añadir alguna tabla más o algún campo, utiliza los CREATE/ALTER tables oportunos, y justifícalos con comentarios en el script.
- En particular necesitarás alguna relación reflexiva. Intenta no usar las de siempre (jefes, partes, parientes ...). ¡Qué tengan que ver algo con tu trabajo 1! ¡sé un poco original! Te podemos ayudar a refinar una idea, si nos la comentas en el foro

Para que te sea más fácil explicar cada SELECT, idealmente cada una llevará sus propias filas de ejemplo:

- Utiliza ROLLBACK después de cada SELECT para limpiar las tablas antes de insertar las filas de ejemplo de la consulta siguiente.
- Si una consulta solo necesita filas de algunas tablas, sus filas de ejemplo solo harán INSERT en las tablas estrictamente necesarias.

Se pide, por cada miembro del equipo:

1. Una consulta que utilice la nueva relación reflexiva en el cálculo de una función de agregación. Naturalmente, la consulta tendrá agrupamiento.
2. Una consulta que represente un anidamiento de funciones de agregación. Haz 2 versiones, una con subconsulta en el FROM y otra con subconsulta en el WITH
3. Modifica la versión con WITH haciendo que el resultado de la consulta anterior sea otra subconsulta que puedas utilizar en una nueva consulta principal.
4. Dos cocientes relacionales con subconsulta en HAVING. En lugar de pedir elementos del dividendo que se relacionan con todos los elementos del divisor, puedes relajarlo a elementos del dividendo que se relacionan al menos con un *porcentaje* de elementos del divisor, seguramente así te sea más fácil reutilizar algunas filas de ejemplo
  1. Uno de los 2 cocientes ha de necesitar algún COUNT(DISTINCT) en la subconsulta, y el otro lo ha de necesitar en la consulta principal. Tendrás que justificar en el script esa necesidad a través de la información de los create tables (i.e., las claves). Por ejemplo, como la clave es X, y agrupamos por Y, se pueden repetir las Zs dentro de cada grupo.
  2. Al menos uno de los dos cocientes deberá de tener un WHERE en la subconsulta
5. Una consulta con subconsulta en el WHERE, que se pueda hacer tanto con MAX/MIN en la subconsulta, o como con ALL en la subconsulta. Presentar una versión con MAX/MIN, otra con ALL y otra con NOT EXISTS
6. Una consulta con subconsulta correlacionada en la SELECT. La subconsulta calculará una función de agregación.
  1. Explica en el script el resultado de la misma deshaciendo la correlación para el caso de una fila que salga en el resultado, y para el caso de otra que no salga.
  2. Haz una consulta equivalente que no necesite subconsulta. Pista: utiliza GROUP BY.
7. Haz una consulta que tenga una intersección y una resta. Rehazla sustituyendo ambas operaciones de conjunto con subconsultas EXISTS/NOT EXISTS.
8. Una consulta con subconsulta correlacionada en el HAVING. Explica el resultado de la misma deshaciendo la correlación para una fila que salga en el resultado, y para otra que no

salga.

Notas:

1. Todas las consultas tendrán al menos un JOIN que sea necesario (Cuidado: Si una tabla es innecesaria para resolver una consulta, resta puntos)
2. Al menos la mitad de las consultas necesitarán que algún JOIN sea externo (Cuidado: Si un JOIN externo no es necesario, también quita puntos )

## **ENTREGABLES**

Entregar **un único script SQL** con:

1. DROPs y CREATEs iniciales
2. Por cada consulta
  1. Los INSERTs con los que vas a ilustrarlo, explicando en un comentario qué efecto estás buscando en el resultado de la consulta al hacer esas inserciones.
  2. La consulta (o consultas, si se piden varias versiones)
  3. Las explicaciones en un comentario, cuando se pidan explicaciones
  4. ROLLBACK para descartar los INSERTs de esa consulta
  5. Esta vez no hay vídeos porque en el examen oral, en la parte de SQL te preguntaré preferentemente cosas de este trabajo. Si no entregas el trabajo, o lo entregas muy mal, tendré que buscar otras preguntas.