

1. Funciones en PL/SQL

1. Crea una **función** `golesLocal` que recibe una cadena de texto **VARCHAR2** representando el resultado de un partido con el formato `GolesLocal–GolesVisitante` y devuelve el número de goles del equipo local como un valor de tipo **NUMBER**. P. ej. `golesLocal('3–0')` debe devolver 3.
2. Crea una **función** `golesVisitante` similar a `golesLocal` pero que devuelve el número de goles del equipo visitante. P. ej. `golesVisitante('3–0')` debe devolver 0.

2. Procedimientos en PL/SQL

1. La tabla `user_objects` almacena todos los objetos creados por el usuario local, como pueden ser las tablas, vistas, funciones y procedimientos. Concretamente, tiene una columna llamada `object_type` que contiene el tipo de cada objeto: **'VIEW'**, **'FUNCTION'**, **'TABLE'** y **'PROCEDURE'**.
Crea un **procedimiento** `elimina_tipo` que recibe una cadena de texto **VARCHAR2** con uno de los 4 tipos de objeto anteriores, y elimina de la base de datos todos los objetos que haya de ese tipo. Para realizar eliminaciones desde un procedimiento PL/SQL utiliza la sentencia **«EXECUTE IMMEDIATE stmt»**, donde `stmt` es una cadena de texto conteniendo la instrucción a ejecutar. Por ejemplo, podríamos eliminar la vista de nombre `DatosPepe` con **«EXECUTE IMMEDIATE 'DROP VIEW DatosPepe'»**. Hay que tener cuidado con dos situaciones:

- Al borrar una tabla hay que añadir **CASCADE CONSTRAINTS** para que las restricciones de clave foránea se eliminen en cascada (si existían) y así garantizar que la eliminación de la tabla no falla.
- A la hora de borrar procedimientos, hay que evitar borrar el procedimiento actual. Se puede acceder al nombre del procedimiento actual con `$$plsql_unit`

Por cada eliminación realizada se debe mostrar un mensaje por pantalla indicando la instrucción ejecutada, y al final mostrar el número de objetos eliminados. Por ejemplo, la salida generada para `elimina_tipo('TABLE')` sería:

```
* Ejecutando: DROP TABLE PERSONA CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE PATROCINADOR CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE PATROCINA CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE JUGADOR CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE FINANCIA CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE ENFRENTA CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE CLUB CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE ASISTE CASCADE CONSTRAINTS
* Ejecutando: DROP TABLE ARBITRO CASCADE CONSTRAINTS
Borrados 9 objetos de tipo TABLE
```

2. Crea un **procedimiento** `actualiza_socios` que acepta el CIF de un Club (tipo **CHAR**) y actualiza el número de sus socios en base a los asistentes que ha tenido en sus partidos. Las reglas de actualización de socios son las siguientes:
- Por cada partido con entre 1 y 3 asistentes, gana 10 socios.
 - Por cada partido con más de 3 asistentes, gana 100 socios.
 - Por cada partido con 0 asistentes, el número de socios no varía.

Al inicio del procedimiento se debe mostrar un mensaje en mayúsculas «ACTUALIZANDO NÚMERO DE SOCIOS DEL CLUB nombre_del_club» seguido de una línea de almohadillas. Por cada partido se debe mostrar el **nombre** de los clubes enfrentados, el número de asistentes y el número de socios que gana (que puede ser 0). Además, cada partido debe estar numerado desde el 1 en adelante, aunque no importa el orden en el que se recorren.

Al final, se debe mostrar el número total de socios ganados y se debe **actualizar el número de socios** del club con el incremento de socios. Por ejemplo, la salida generada por `actualiza_socios ('1111111X')` sería:

```
ACTUALIZANDO NUMERO DE SOCIOS DEL CLUB: REAL MADRID CF
#####
1) Futbol Club Barcelona vs. Real Madrid CF
2 asistentes -> +10 socios

2) Real Madrid CF vs. Futbol Club Barcelona
5 asistentes -> +100 socios

3) Real Madrid CF vs. Paris Saint-Germain Football Club
4 asistentes -> +100 socios

4) Paris Saint-Germain Football Club vs. Real Madrid CF
2 asistentes -> +10 socios

TOTAL: incremento de 220 socios nuevos
```

Si no existe ningún club con el CIF pasado como parámetro, no se debe realizar ningún cambio en la base de datos pero se debe mostrar el mensaje «**No** existe ningún club con CIF cif_del_club». P.ej. la salida de `actualiza_socios ('11111114X')` sería:

```
No existe ningún club con CIF 11111114X
```

3. Crea un **procedimiento** `actualiza_patrocinios` que acepta el CIF de un Club (tipo **CHAR**) y actualiza las cantidades que recibe por cada patrocinador (información de la tabla **Financia**) en base a los resultados de sus partidos. Las reglas son las siguientes:

- Cada victoria como local le aumenta 100 EUR por cada gol de diferencia. P.ej. una victoria local por 3-1 le aumentaría 200 EUR.
- Cada victoria como visitante le aumenta 200 EUR por cada gol de diferencia. P.ej. una victoria local por 0-3 le aumentaría 600 EUR.
- Cada empate como visitante le aumenta 50 EUR.
- El resto de partidos no modifica sus patrocinios.

Al inicio del procedimiento se debe mostrar un mensaje en mayúsculas «ACTUALIZANDO PATROCINIOS DEL CLUB nombre_del_club» seguido de una línea de almohadillas. Por cada partido se debe mostrar el **CIF** de los clubes enfrentados, el resultado entre paréntesis y el incremento en euros que genera (que puede ser 0 EUR).

Al final, se debe mostrar la cantidad total que hay que incrementar cada financiación recibida por el club, además de **actualizar convenientemente las entradas de la tabla Financia** relacionadas con el club. Por ejemplo, la salida generada por `actualiza_patrocinios ('11111112X')` sería:

```
ACTUALIZANDO PATROCINIOS DEL CLUB: FUTBOL CLUB BARCELONA
#####
11111111X - 11111112X (3-2) -> +0 EUR
11111112X - 11111111X (2-2) -> +0 EUR
11111112X - 11111113X (3-2) -> +100 EUR
11111113X - 11111112X (0-1) -> +200 EUR
* Actualizando +300 EUR cada patrocinio al club
```

Si no existe ningún club con el CIF pasado como parámetro, no se debe realizar ningún cambio en la base de datos pero se debe mostrar el mensaje «No existe ningún club con CIF cif_del_club». P.ej. la salida de `actualiza_socios('11111114X')` sería:

No existe ningún club con CIF 11111114X

3. Ejercicios adicionales: comprobación de restricciones de la BD

Como hemos ido viendo a lo largo de la asignatura, hay varias restricciones de la BD sobre la liga de fútbol que no se han podido representar en el modelo relacional resultante. Por cada una de estas restricciones crea un **procedimiento** que comprueba si se cumplen y en caso contrario muestra un mensaje informativo que indica qué entidades las incumplen. Las restricciones que habría que manejar son:

- «Los patrocinadores que financian un club deben patrocinar al menos 5 de sus jugadores»
- «Un club debe recibir más financiación que la suma de patrocinios de sus jugadores»
- «Un club tiene como mínimo un jugador»
- «Un club tiene como máximo 25 jugadores»
- «Cada patrocinador debe financiar al menos un club»

4. Ejercicios adicionales: actualización de la BD preservando restricciones

Crea **procedimientos** para insertar, actualizar y borrar en las tablas adecuadas de tal manera que se consiga garantizar las **restricciones de la BD** que no se pudieron representar en el diagrama ER o en su transformación a MR. Para ese apartado puedes asumir que la BD ya cumple las restricciones en el estado actual, y debes modificar la BD únicamente si el resultado sigue cumpliéndolas. Concretamente:

1. «Los patrocinadores que financian un club deben patrocinar al menos 5 de sus jugadores»
Será necesario manejar la inserción de una financiación a un club y su actualización (por si cambia el club o el patrocinador).
2. «Un club debe recibir más financiación que la suma de patrocinios de sus jugadores»
Sería necesario manejar la inserción y actualización de un patrocinio a un jugador, además del borrado de la financiación a un club. Si la BD es consistente y se borra un patrocinio a jugador entonces seguirá siendo consistente, al igual que si se añade una nueva financiación a un club (tened en cuenta que las cantidades son siempre positivas).
3. «Un club tiene como mínimo un jugador»
Hay que manejar el borrado de jugadores para evitar borrar de más, pero también su actualización por si cambian de club.
4. «Un club tiene como máximo 25 jugadores»
Necesitamos manejar la inserción de jugadores y su actualización, por si cambian de club.
5. «Cada patrocinador debe financiar al menos un club»
Sería necesario manejar el borrado y actualización de financiaciones a clubes.