

Creación de una práctica de bases de datos relacionales con SQLite.

David González Márquez
david.gonzalez.marquez@gmail.com

10 de abril de 2019



Índice

I Estudio del problema.	3
1. Introducción	4
2. Estimación de los resultados del aprendizaje que adquirirán los alumnos.	4
3. Contexto del problema elegido.	5
4. Estudio del esfuerzo requerido para la asimilación de dichos resultados de aprendizaje.	5
II Diseño de la práctica.	6
5. Herramientas.	7
6. Requisitos de la solución.	7
7. Criterios de corrección.	7
Anexo: Práctica de bases de datos relacionales con SQLite	9
A. Introducción breve a SQLite.	10
B. Repaso básico a SQL y SQLite	10
B.1. Tipos de datos	10
B.2. Comandos	11
B.3. Creación de tablas	11
B.4. Consultas en la base de datos	11
B.5. Otros comandos	12
C. Explicación sobre la base de datos.	12
D. Ejercicios de la práctica	12
D.1. Ejercicio 1	13
D.2. Ejercicio 2	14
D.3. Ejercicio 3	15
D.4. Ejercicio 4	15
E. Resultados que se deben entregar.	16
Anexo: Solución documentada de la práctica	17
E.1. Ejercicio 1	18
E.2. Ejercicio 2	21
E.3. Ejercicio 3	24

E.4. Ejercicio 4	27
Bibliografía	29

Parte I
Estudio del problema.

1. Introducción

En este trabajo procederemos a diseñar y crear una práctica sobre bases de datos relaciones con SQLite. La práctica se entrega a alumnos que ya han tenido unas 50 horas de clases teóricas sobre bases de datos. La duración que se estima para la práctica es de 20 horas.

Se ha intentado con la práctica motivar a los alumnos por lo que se ha elegido un tema bastante atractivo para la mayoría como es el fútbol.

En la práctica se describe una estructura básica que los alumnos tendrán que crear y sobre la cuál se realizaran consultas y se trabajará en el resto de la práctica.

2. Estimación de los resultados del aprendizaje que adquirirán los alumnos.

Mediante esta práctica los alumnos adquirirán los siguientes resultados.

- Síntesis de la información dada en el enunciado en un conjunto de tablas que serán nuestra base de datos.
- Creación y diseño de tablas para almacenar la información.
- Poblar una tabla ya creada con datos.
- Trabajar con todo tipo de consultas SQL, desde las más simples hasta las más complejas.
- Crear vistas y ver la utilidad que tienen.
- Añadir nuevas tablas a un modelo de una base de datos ya creado y relacionar la nueva información con la que ya existía.

La práctica está dividida en 4 ejercicios. El primer ejercicio consiste en la creación de las tablas a partir de la información dada en el enunciado, el alumno debe ser capaz de sintetizar la información dada y materializar la base de datos en forma de tablas conexas. El resto de ejercicios dependen en gran parte de este primer ejercicio.

El ejercicio 2 contiene las consultas que se le van a pedir al alumno que realice, para la elaboración del ejercicio 1 se espera que mire también en este apartado las consultas, de forma que pueda valorar si el modelo que esta elaborando podría responder a estas consultas. Las consultas del ejercicio 2 están pensadas para que vayan aumentando de dificultad de forma que la media de los alumnos pueda llegar a hacer la mayoría aunque quizás no termine de hacerlas todas.

Cada consulta se puede hacer siempre de varias formas pero están preparadas para que el alumno vaya utilizando características distintas del lenguaje SQL, consultas por fecha, con condiciones numéricas, etc.

El ejercicio 3 sirve para varios propósitos, en primer lugar el alumno aprende como crear tablas respetando la coherencia con las que ya se han creado. También se le pide que inserte datos específicos. En el apartado 2 de este ejercicio se alcanza un punto clave de la práctica, el ejercicio es sencillo pero para que

sea correcto el alumno debe entender como puede añadir nueva información a la base de datos y relacionarla con la ya existente, al tiempo que es coherente con la realidad detrás de los datos.

Esto se confirma en el apartado 3, 4 y 5. Si ha comprendido bien la práctica y ha realizado bien las modificaciones no habrá ningún problema en que un jugador participe en dos ediciones de la copa del mundo con selecciones distintas pero si no lo ha hecho no realizará bien este ejercicio.

Finalmente el ejercicio 4 le sirve al alumno para ver lo que son las vistas y lo útiles que le hubieran sido si hubiera hecho alguna al principio.

3. Contexto del problema elegido.

El problema elegido es atrayente para la mayor parte de los alumnos. Quizás hubiera sido más atrayente con los equipos de la liga española pero creemos que para el propósito de esta práctica nos servía mejor una base de datos como la de los mundiales. En primer lugar no son tantos datos y en segundo lugar presentan ciertas características exclusivas al no ser propiamente clubes sino países.

El mayor problema que se planteó fue como dejar que los alumnos poblaran la base de datos, se podrían dar scripts sql que realizaran esta labor, además podrían tener datos de internet reales y llenarían la base de datos rápidamente con muchos datos, lo cuál sería positivo a la hora de realizar consultas. Sin embargo finalmente se optó en contra de esta aproximación, ya que al dar los datos estas dando a los alumnos la estructura de sus tablas, por lo que se optó con dejar que cada uno poblara la base de datos libremente, a pesar de sus inconvenientes.

4. Estudio del esfuerzo requerido para la asimilación de dichos resultados de aprendizaje.

La práctica se estima en unas 20 horas de trabajo por parte del alumno.

De estas 20 horas se espera que dedique 2 horas a familiarizarse con SQLite, con las herramientas y con el enunciado que se le presenta.

Al ejercicio 1 se estima que el alumno dedique unas 6 horas, para la creación de las tablas e insertar datos.

En el ejercicio 2 se estima que el alumno dedique 7 horas. 4 horas para las 6 primeras consultas y 3 horas para las otras 6. Se cuenta con que si bien las primeras consultas son sencillas el alumno tendrá que dedicar más tiempo ya que no tiene experiencia, conforme va avanzando por los apartados tiene más experiencia pero se le introducen elementos nuevos y requisitos en las consultas.

El ejercicio 3 con un análisis apropiado se estima que dedique unas 3 horas, en este caso si el ejercicio no lo hace de forma adecuada podría dedicarle mucho menos pero no estará realizando bien los apartados ya que el análisis se lo ha saltado.

El ejercicio 4 es sencillo ya que una vista es muy similar a una consulta y ya ha realizado varias, por tanto se supone que dedicará 2 horas a este último ejercicio.

Parte II
Diseño de la práctica.

5. Herramientas.

El alumno debe utilizar SQLite como base de datos en su versión 3. Se le aconseja utilizar la línea de comandos pero puede apoyarse en herramientas visuales si lo necesita. Sin embargo se pide que el fichero de entrega sea ejecutable como script lo cuál le obliga al menos a comprobar que puede ejecutarse el fichero que envía por línea de comando.

Durante este tiempo de práctica sería bueno que el alumno tuviera a su disposición tutorías para consultar las posibles dudas. Si bien el trabajo lo debe realizar él, el profesor puede guiar en la dirección adecuada.

6. Requisitos de la solución.

Como se especifica muy claramente en el enunciado el alumno debe entregar dos ficheros por ejercicio, uno con el código SQL y otro con los resultados en texto.

Además de esto el alumno deberá afrontar una defensa de la práctica individual. Esta defensa está preparada para detectar posibles copias. En ella el profesor cuestionará el razonamiento tras alguna de las consultas realizadas y especialmente de las soluciones propuestas en el ejercicio 3. También en esta misma defensa el alumno deberá realizar una pequeña modificación a alguna de sus consultas o realizar una consulta simple nueva.

7. Criterios de corrección.

El total de la práctica esta valorado en 10 puntos que se dividen de la siguiente forma:

1. Ejercicio 1 (4 puntos)
2. Ejercicio 2 (3 puntos) Cada apartado 0.25 puntos
3. Ejercicio 3 (2 puntos) Cada apartado 0.5 puntos
4. Ejercicio 4 (1 punto)
Apartado 1 (0.25 puntos) Apartado 2 (0.5 puntos) Apartado 3 (0.25 puntos)

Con el código en primer lugar se aplicará un programa de detección de plagios para detectar posibles similitudes entre el código de los alumnos. Las consultas son las mismas pero siempre hay varias formas de hacerlas por lo que sería sospechoso que fueran exactamente iguales.

Para que un apartado se considere correctamente resuelto en primer lugar se debe poder ejecutar el código SQL. Además si es una consulta debe satisfacer el enunciado mostrando los datos que se piden. La estructura del código puede ser variada pero una estructura demasiado compleja o ineficiente puede conducir a rebajar el valor del apartado en un %10. En caso de que la solución no sea la pedida pero este suficientemente cerca se valorará con un %50 del valor del apartado.

Uno de los puntos fundamentales de la corrección es el apartado 2 del ejercicio 3. El alumno puede haberlo resuelto de múltiples formas, es raro que dos alumnos lo hayan hecho de forma exactamente igual. No hay una única forma correcta pero hay muchas que no son correctas, el apartado 4 de dicho ejercicio también ayudará en la labor ya que no puede dar un resultado correcto sino se ha realizado el apartado 4 correctamente.

En la defensa de la práctica la nota obtenida por el alumno será de un suspenso o un aprobado, si un alumno no aprueba la defensa de la práctica su nota en la práctica pasa a ser un suspenso independiente de la nota que fuera. Si aprueba la defensa la nota permanece igual.

Anexo: Práctica de bases de datos relacionales con SQLite

A. Introducción breve a SQLite.

En esta práctica sobre SQL utilizaremos el software SQLite.

SQLite es un sistema de gestión de bases relacional, famoso por su pequeño tamaño. Es un proyecto de dominio público creado por Richard Hipp. A diferencia de otros sistemas de gestión cliente-servidor el motor de SQLite no es un proceso independiente lo que hace que la latencia sea menor y más el acceso más eficiente. SQLite también soporta operaciones ACID [9].

La versión 3 permite bases de datos de hasta 2 Terabytes de tamaño y la inclusión de campos tipo BLOB [2]. Esta es la versión que utilizaremos para la práctica.

Debido a su facilidad de uso, su pequeño tamaño y su versatilidad SQLite es utilizado en una gran variedad de aplicaciones. Entre ellas podemos encontrar *Mozilla Firefox*, *Skype* o *XBMC* [8]. Es especialmente adecuado para los sistemas integrados y su uso ha sido muy popular en las aplicaciones para smartphones con SO Android y iOS [1].

En la página web de SQLite podemos encontrar el código fuente y los binarios para los principales sistemas operativos [4]. El programa funciona por línea de comandos y no tiene ninguna interfaz gráfica. Sin embargo es fácil encontrar interfaces gráficas en Internet que podemos utilizar. Entre ellas tenemos *SQLiteStudio* [7], *SQLite Administrator* [6] o *DB Browser for SQLite* [3].

Para el desarrollo de la práctica podéis apoyaros en cualquier herramienta gráfica que queráis utilizar pero se recomienda utilizar, al menos en los primeros pasos, la interfaz de comandos. Además, los resultados de la práctica siempre se evaluarán sobre la interfaz de la línea de comandos por lo que os debéis asegurar que el código se ejecuta. También se valorará la claridad de las consultas, consultas demasiado largas o innecesariamente complejas serán penalizadas.

B. Repaso básico a SQL y SQLite

En SQLite lo primero que debemos hacer es iniciar nuestra base de datos, el mismo comando que nos servirá para abrirla lo utilizamos para crearla ya que detecta que la base de datos no existe y la crea. Simplemente debemos escribir el nombre del ejecutable en la consola seguido del nombre de nuestra base de datos “sqlite3 nombreBaseDatos.db”.

B.1. Tipos de datos

Cada gestor de bases de datos tiene unos tipos predefinidos. En SQLite tenemos tan solo los siguientes:

- NULL
- INTEGER
- REAL
- TEXT
- BLOB

Los nombres son autoexplicativos en su mayor parte, pero para mayor información podemos consultar el manual [5]. Allí explica como por ejemplo las fechas no tienen un tipo especial sino que se almacenan o como texto o como real o como entero.

B.2. Comandos

En SQLite los comandos van precedidos por punto y son similares a los comandos de otros gestores de bases de datos. A continuación repasamos los más importantes, para un listado completo tan solo tenemos que utilizar el comando ayuda.

- `.help` Muestra los comandos disponibles
- `.quit` Nos permite salir del programa
- `.tables` Muestra una lista de las tablas creadas en la base de datos
- `.output ?FILENAME?` Envía la salida estándar a un fichero.
- `.schema` Muestra el código SQL necesario para crear los elementos presentes

B.3. Creación de tablas

La creación de tablas sigue la misma sintaxis que cualquier motor de SQL, utilizando el comando `CREATE table`.

```
CREATE TABLE ciudades (  
    id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
    nombre VARCHAR (255) NOT NULL,  
    nombre_Largo VARCHAR (255),  
    id_pais INTEGER NOT NULL  
);
```

Posteriormente debemos insertar datos en las tablas que hemos creado, para ello utilizamos la orden `INSERT INTO table`.

```
insert into ciudades values(1,"Sevilla", "Luz_de_las_Gentes", 3);  
insert into ciudades values(2,"Madrid", NULL, 3);
```

B.4. Consultas en la base de datos

Una vez tenemos la información en la base de datos es fundamental que podamos acceder a ella, para ello utilizamos las consultas. Las consultas se realizan utilizando la orden `SELECT ? FROM table`.

```
select * from ciudades;  
select nombre from ciudades;  
select nombre from ciudades where id_pais="3";
```

Las consultas se realizan con toda la potencia de la sintaxis de SQL, pudiéndose realizar consultas a varias tablas, con varias condiciones, agrupadas, por orden, etc.

El formato completo sería el siguiente:

```
SELECT [ALL | DISTINCT ]
           <nombre_campo> [{, <nombre_campo>}]
FROM <nombre_tabla>|<nombre_vista>
           [{, <nombre_tabla>|<nombre_vista>}]
[WHERE <condicion> [{ AND|OR <condicion> }]]
[GROUP BY <nombre_campo> [{, <nombre_campo> }]]
[HAVING <condicion> [{ AND|OR <condicion> }]]
[ORDER BY <nombre_campo>|<indice_campo> [ASC | DESC]
           [{, <nombre_campo>|<indice_campo>
           [ASC | DESC }]]]
```

Dónde los corchetes indican todas las opciones que pueden o no aparecer.

B.5. Otros comandos

Otras operaciones SQL que se pueden realizar son las modificaciones de datos o el borrado.

Para modificar un dato ya guardado utilizaremos el comando UPDATE table.

```
update ciudades set nombre_Largo="Madrid_La_llana" where
nombre="Madrid" ;
```

Para borrar un registro de la base de datos utilizaremos el comando DELETE FROM table

```
delete from ciudades where nombre="Madrid" ;
```

C. Explicación sobre la base de datos.

En esta práctica vamos a crear una base de datos para almacenar información de los resultados de los mundiales de fútbol. Necesitaremos almacenar las selecciones que participan, a que país pertenecen, en que evento participan y los detalles de cada uno de los partidos.

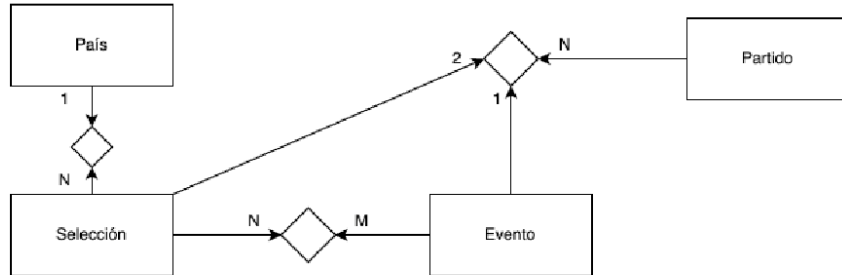
También es necesario que sepamos el desarrollo de la competición, por lo tanto tenemos que saber a que ronda pertenece cada uno de los partidos, si es la final o si son los preliminares.

En la figura 1 adjunta se muestra un diagrama de entidad relación de la práctica pedida.

D. Ejercicios de la práctica

En esta práctica se te pedirá que realicéis ciertas operaciones utilizando SQLite.

Figura 1: Diagrama de Entidad Relación



Cualquier comando que se haya ejecutado debe quedar reflejado en los ficheros que se entreguen.

Por cada ejercicio se deben entregar dos ficheros diferentes:

- NombreApellidosEjercicioXCodigo.sql

Este fichero deberá poder ser interpretado por SQLite y contendrá todos los comandos utilizados para el Ejercicio correspondiente. Para diferenciar entre los apartados del ejercicio utiliza un comentario en el fichero.

```

/* Apartado 1*/
Select * from cities;
/* Apartado 2 */
CREATE TABLE cities (
    id          INTEGER          PRIMARY KEY AUTOINCREMENT
                                NOT NULL,
    name        VARCHAR (255)   NOT NULL,
    code        VARCHAR (255),
    country_id  INTEGER          NOT NULL
);
/* Apartado 3*/
Select * from places;
    
```

- NombreApellidosEjercicioResultado.txt

Este fichero contendrá el resultado de los comandos ejecutados. En el caso de consultas el resultado de la misma y en el caso de una vista el resultado de una consulta de todos los elementos de la vista.

Donde X es el número de ejercicio que corresponda.

D.1. Ejercicio 1

Este ejercicio consiste en la creación de las tablas que representen la información que se pide almacenar, además de poblarlas con datos. No se pide que es complete toda la información de todas las copas del mundo, pero en cada tabla deben aparecer un número de registros aceptable, al menos unos diez y con la

mayor variedad posible. Tanto la creación de datos como la inserción de datos deben quedar perfectamente reflejadas en el archivo correspondiente, de forma que se pueda reproducir a la exactitud la base de datos.

Las tablas que se piden elaborar son como mínimo las siguientes;

- Selecciones: que represente las distintas selecciones que hay.
- Países: los distintos países que existen y que están representados por su selección.
- Eventos: las competiciones que se han disputado
- Partidos: que equipos jugaron, cuando y el resultado.

Cualquier otra tabla que se considere necesaria se puede y debe añadir.

D.2. Ejercicio 2

Este ejercicio consiste en realizar varias consultas. Se pide que haya datos suficientes en la base de datos para poder observar el correcto funcionamiento de las consultas, si hacen falta más datos se pueden añadir, reflejándolo en el archivo de comandos.

Realiza las siguientes consultas:

1. Enumera el nombre de todos los equipos que han participado alguna vez en el mundial.
2. Muestra la fecha de inicio de todas las ediciones del mundial.
3. Para un mundial muestra en dos columnas para cada partido el nombre de los dos países que se enfrentaron entre sí.
4. A la consulta anterior añade además la puntuación final de cada uno de los dos equipos.
5. Muestra el nombre de los equipos que participaron en la primera edición de la que tenemos datos.
6. Lista los partidos que se hayan realizado entre el 1900 y el 2010 en los que haya habido al menos un gol.
7. Ya sabemos que los Españoles somos muy buenos al fútbol pero ahora muestra en que ediciones del mundial ha estado la selección española.
8. Lista los partidos en los que ha jugado la selección española y no ha empatado.
9. Ahora vamos a ver cual ha sido la mayor cantidad de goles que ha conseguido marcar nuestra selección
10. Muestra el número de veces que cada equipo ha participado en el mundial, ordénalos desde el que más veces haya participado hasta el que menos.
11. Los nombres de los equipos que participan en la copa del mundo no siempre coinciden con el nombre del equipo, en una consulta muestra para cada país los equipos que tiene asociados.

12. Muestra en una consulta los resultados de las finales de la copa del mundo. Debes mostrar en la misma consulta la competición que fue, el nombre de los dos equipos que participaban y el resultado final de cada uno de ellos.

D.3. Ejercicio 3

Sobre la base de datos que se te ha entregado se te pide que crees algunas tablas. Ten en cuenta las posibles limitaciones y la estructura de la base de datos existente. Razona las decisiones que tomes, si tienes algún comentario o aclaración la puedes añadir en el fichero .sql como un comentario.

En los casos que se pida que se inserten datos en las tablas también debemos añadir el código que lo hace en el fichero .sql en el apartado correspondiente.

1. Hay jugadores que han marcado la historia de los mundiales, añade una tabla para almacenar los nombres de estos jugadores. En el caso que puedas también debes dejar espacio para añadir la fecha de nacimiento en caso de que este disponible.
Inserta algún valor en la tabla de tus jugadores favoritos.
2. Pero los jugadores sin sus equipos no serían nada. Ahora queremos saber de nuestras leyendas con que equipos jugaron, crea las tablas necesarias e inserta los datos apropiados en ellas para que podamos consultar esta información.
3. Ahora vamos a introducir nuestro nombre en la historia, en este caso vamos a sustituir el nombre de uno de los grandes “Luis Monti” por el nuestro. Introduce por lo tanto tu nombre en la tabla con las leyendas del fútbol pero con los datos de este jugador. Añade también en los equipos en los que participó este jugador.
4. Vamos a mirar nuestro registro de partidos, haz una consulta que muestre todos los partidos que nos corresponden con nuestra nueva identidad futbolística en los datos de las copas del mundo. No es necesario diferenciar entre los partidos en los que jugamos o no, lo importante es participar, mientras que formáramos parte del equipo cuenta para el listado.

D.4. Ejercicio 4

En este último ejercicio se te pedirá que crees ciertas vistas sobre la base de datos.

1. Crea una vista dónde se muestren todos los partidos jugados, con el nombre del evento y de los dos equipos.
2. Crea una vista que muestre el número de copas del mundo en las que ha participado cada equipo, una fila por cada equipo en la que una columna ponga el número de veces que ha participado. Ordena el resultado de forma que en la primera línea salga el equipo que más veces ha participado.
3. Repite la vista anterior pero en este caso en vez de por equipos por países.

E. Resultados que se deben entregar.

Se deben comprimir todos los ficheros en un archivo sin meterlos en ninguna carpeta, nombra al fichero:

NombreApellidosBD.zip

Adjunta también la base de datos que has creado:

BDCompleteNombreApellidos.db

Antes de entregarlo revisa que la nomenclatura de los ficheros es correcta y que los ficheros .sql se pueden ejecutar.

Anexo: Solución documentada de la práctica

E.1. Ejercicio 1

Este ejercicio consiste en la creación de las tablas que representen la información que se pide almacenar, además de poblarlas con datos. No se pide que es complete toda la información de todas las copas del mundo, pero en cada tabla deben aparecer un número de registros aceptable, al menos unos diez y con la mayor variedad posible. Tanto la creación de datos como la inserción de datos deben quedar perfectamente reflejadas en el archivo correspondiente, de forma que se pueda reproducir a la exactitud la base de datos.

Las tablas que se piden elaborar son como mínimo las siguientes;

- Selecciones: que represente las distintas selecciones que hay.
- Países: los distintos países que existen y que están representados por su selección.
- Eventos: las competiciones que se han disputado
- Partidos: que equipos jugaron, cuando y el resultado.

Cualquier otra tabla que se considere necesaria se puede y debe añadir.

```
CREATE TABLE "countries"  
("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
"name" varchar(255) NOT NULL);  
  
CREATE TABLE "events"  
("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
"key" varchar(255) NOT NULL, "start_at" date NOT NULL, "end_at" date);  
CREATE TABLE "events_teams"  
("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
"event_id" integer NOT NULL, "team_id" integer NOT NULL);  
  
CREATE TABLE "games"  
("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
"event_id" integer NOT NULL, "round" varchar(255) NOT NULL,  
"team1_id" integer NOT NULL, "team2_id" integer NOT NULL,  
"play_at" datetime NOT NULL, "score1" integer, "score2" integer,  
"winner" integer);  
  
CREATE TABLE "teams"  
("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
"key" varchar(255) NOT NULL, "title" varchar(255) NOT NULL,  
"country_id" integer NOT NULL);
```

```
INSERT INTO "countries" VALUES(63, 'China');  
INSERT INTO "countries" VALUES(109, 'Germany');  
INSERT INTO "countries" VALUES(110, 'Estonia');  
INSERT INTO "countries" VALUES(111, 'Spain');  
INSERT INTO "countries" VALUES(112, 'Finland');  
INSERT INTO "countries" VALUES(113, 'France');
```

```

INSERT INTO "countries" VALUES(114, 'Greece ');
INSERT INTO "countries" VALUES(115, 'Ireland ');
INSERT INTO "countries" VALUES(116, 'Italy ');
INSERT INTO "countries" VALUES(120, 'Portugal ');
INSERT INTO "countries" VALUES(127, 'Poland ');
INSERT INTO "countries" VALUES(186, 'Argentina ');
INSERT INTO "countries" VALUES(194, 'Uruguay ');

```

```

INSERT INTO "events" VALUES(1, 'world.1930 ', '1930-07-13 ', NULL);
INSERT INTO "events" VALUES(2, 'world.1934 ', '1934-05-27 ', NULL);
INSERT INTO "events" VALUES(3, 'world.1938 ', '1938-06-04 ', NULL);
INSERT INTO "events" VALUES(4, 'world.1950 ', '1950-06-24 ', NULL);
INSERT INTO "events" VALUES(5, 'world.1954 ', '1954-06-16 ', NULL);
INSERT INTO "events" VALUES(6, 'world.1958 ', '1958-06-08 ', NULL);
INSERT INTO "events" VALUES(7, 'world.1962 ', '1962-05-30 ', NULL);
INSERT INTO "events" VALUES(8, 'world.1966 ', '1966-07-11 ', NULL);
INSERT INTO "events" VALUES(9, 'world.1970 ', '1970-05-31 ', NULL);
INSERT INTO "events" VALUES(10, 'world.1974 ', '1974-06-13 ', NULL);
INSERT INTO "events" VALUES(11, 'world.1978 ', '1978-05-01 ', NULL);
INSERT INTO "events" VALUES(12, 'world.1982 ', '1982-06-13 ', NULL);
INSERT INTO "events" VALUES(13, 'world.1986 ', '1986-05-31 ', NULL);
INSERT INTO "events" VALUES(14, 'world.1990 ', '1990-06-08 ', NULL);
INSERT INTO "events" VALUES(15, 'world.1994 ', '1994-06-17 ', NULL);
INSERT INTO "events" VALUES(16, 'world.1998 ', '1998-06-10 ', NULL);
INSERT INTO "events" VALUES(17, 'world.2002 ', '2002-05-31 ', NULL);
INSERT INTO "events" VALUES(18, 'world.2006 ', '2006-06-09 ', NULL);
INSERT INTO "events" VALUES(19, 'world.2010 ', '2010-06-11 ', NULL);
INSERT INTO "events" VALUES(20, 'world.2014 ', '2014-06-12 ', NULL);

```

```

INSERT INTO "teams" VALUES(70, 'chn ', 'China ', 63);
INSERT INTO "teams" VALUES(224, 'frg ', 'West_Germany_(-1989) ', 109);
INSERT INTO "teams" VALUES(225, 'gdr ', 'East_Germany_(-1989) ', 109);
INSERT INTO "teams" VALUES(127, 'ger ', 'Germany ', 109);
INSERT INTO "teams" VALUES(128, 'est ', 'Estonia ', 110);
INSERT INTO "teams" VALUES(129, 'esp ', 'Spain ', 111);
INSERT INTO "teams" VALUES(130, 'fin ', 'Finland ', 112);
INSERT INTO "teams" VALUES(131, 'fra ', 'France ', 113);
INSERT INTO "teams" VALUES(132, 'gre ', 'Greece ', 114);
INSERT INTO "teams" VALUES(133, 'irl ', 'Ireland ', 115);
INSERT INTO "teams" VALUES(134, 'ita ', 'Italy ', 116);
INSERT INTO "teams" VALUES(138, 'por ', 'Portugal ', 120);
INSERT INTO "teams" VALUES(145, 'pol ', 'Poland ', 127);
INSERT INTO "teams" VALUES(210, 'arg ', 'Argentina ', 186);
INSERT INTO "teams" VALUES(214, 'uru ', 'Uruguay ', 194);
INSERT INTO "teams" VALUES(137, 'ned ', 'Netherlands ', 119);

```

```

INSERT INTO "events_teams" VALUES(303, 17, 70);
INSERT INTO "events_teams" VALUES(64, 5, 224);
INSERT INTO "events_teams" VALUES(78, 6, 224);

```

```

INSERT INTO "events_teams" VALUES(93,7,224);
INSERT INTO "events_teams" VALUES(110,8,224);
INSERT INTO "events_teams" VALUES(127,9,224);
INSERT INTO "events_teams" VALUES(141,10,224);
INSERT INTO "events_teams" VALUES(158,11,224);
INSERT INTO "events_teams" VALUES(178,12,224);
INSERT INTO "events_teams" VALUES(203,13,224);
INSERT INTO "events_teams" VALUES(226,14,224);
INSERT INTO "events_teams" VALUES(142,10,225);
INSERT INTO "events_teams" VALUES(34,3,127);
INSERT INTO "events_teams" VALUES(21,2,127);
INSERT INTO "events_teams" VALUES(249,15,127);
INSERT INTO "events_teams" VALUES(282,16,127);
INSERT INTO "events_teams" VALUES(312,17,127);
INSERT INTO "events_teams" VALUES(334,18,127);
INSERT INTO "events_teams" VALUES(363,19,127);
INSERT INTO "events_teams" VALUES(414,20,127);
INSERT INTO "events_teams" VALUES(22,2,129);
INSERT INTO "events_teams" VALUES(47,4,129);
INSERT INTO "events_teams" VALUES(97,7,129);
INSERT INTO "events_teams" VALUES(115,8,129);
INSERT INTO "events_teams" VALUES(164,11,129);
INSERT INTO "events_teams" VALUES(185,12,129);
INSERT INTO "events_teams" VALUES(211,13,129);
INSERT INTO "events_teams" VALUES(233,14,129);
INSERT INTO "events_teams" VALUES(257,15,129);
INSERT INTO "events_teams" VALUES(288,16,129);
INSERT INTO "events_teams" VALUES(319,17,129);
INSERT INTO "events_teams" VALUES(340,18,129);
INSERT INTO "events_teams" VALUES(370,19,129);
INSERT INTO "events_teams" VALUES(420,20,129);

```

```

INSERT INTO "games" VALUES(1,1,'Matchday_1',131,190
,'1930-07-13_12:00:00.000000',4,1,1);
INSERT INTO "games" VALUES(150,7,'Matchday_2',223,129
,'1962-05-31_12:00:00.000000',1,0,1);
INSERT INTO "games" VALUES(151,7,'Matchday_3',211,223
,'1962-06-02_12:00:00.000000',0,0,0);
INSERT INTO "games" VALUES(152,7,'Matchday_4',129,190
,'1962-06-03_12:00:00.000000',1,0,1);
INSERT INTO "games" VALUES(153,7,'Matchday_5',211,129
,'1962-06-06_12:00:00.000000',2,1,1);
INSERT INTO "games" VALUES(22,2,'Preliminary_round',129,211
,'1934-05-27_12:00:00.000000',3,1,1);
INSERT INTO "games" VALUES(31,2,'Quarter-finals_replays'
,134,129,'1934-06-01_12:00:00.000000',1,0,1);
INSERT INTO "games" VALUES(25,2,'Preliminary_round'
,134,191,'1934-05-27_12:00:00.000000',7,1,1);
INSERT INTO "games" VALUES(32,2,'Semi-finals',134,124

```

```

, '1934-06-03_12:00:00.000000' , 1, 0, 1);
INSERT INTO "games" VALUES(35, 2, 'Final', 134, 223
, '1934-06-10_12:00:00.000000' , 2, 1, 1);
INSERT INTO "games" VALUES(40, 3, 'First_round', 134, 160
, '1938-06-05_12:00:00.000000' , 2, 1, 1);
INSERT INTO "games" VALUES(752, 19, 'Matchday_6', 129, 153
, '2010-06-16_16:00:00.000000' , 0, 1, 2);
INSERT INTO "games" VALUES(754, 19, 'Matchday_11', 129, 117
, '2010-06-21_20:30:00.000000' , 2, 0, 1);
INSERT INTO "games" VALUES(755, 19, 'Matchday_15', 212, 129
, '2010-06-25_20:30:00.000000' , 1, 2, 2);
INSERT INTO "games" VALUES(768, 19, 'Quarterfinals', 213, 129
, '2010-07-03_20:30:00.000000' , 0, 1, 2);
INSERT INTO "games" VALUES(770, 19, 'Semifinals', 127, 129
, '2010-07-07_20:30:00.000000' , 0, 1, 2);
INSERT INTO "games" VALUES(772, 19, 'Final', 137, 129
, '2010-07-11_20:30:00.000000' , 0, 1, 2);

```

E.2. Ejercicio 2

Realiza las siguientes consultas:

1. Enumera el nombre de todos los equipos que han participado alguna vez en el mundial.

```

SELECT t.title
FROM teams t
INNER JOIN
events_teams et ON et.team_id = t.id
INNER JOIN
events e ON e.id = et.event_id;

```

2. Muestra la fecha de inicio de todas las ediciones del mundial.

```

SELECT events.[key],
events.start_at
FROM events;

```

3. Para un mundial muestra en dos columnas para cada partido el nombre de los dos países que se enfrentaron entre sí.

```

SELECT t1.title ,
t2.title
FROM games g
INNER JOIN
teams t1 ON t1.id = g.team1_id
INNER JOIN
teams t2 ON t2.id = g.team2_id
INNER JOIN
events e ON e.id = g.event_id
WHERE e.[key] = 'world.2010';

```

4. A la consulta anterior añada además la puntuación final de cada uno de los dos equipos.

```
SELECT t1.title ,
        g.score1 ,
        g.score2 ,
        t2.title
FROM games g
INNER JOIN
teams t1 ON t1.id = g.team1_id
INNER JOIN
teams t2 ON t2.id = g.team2_id
INNER JOIN
events e ON e.id = g.event_id
WHERE e.[key] = 'world.2010';
```

5. Muestra el nombre de los equipos que participaron en la primera edición de la que tenemos datos.

```
SELECT t.title
FROM teams t
INNER JOIN
events_teams et ON et.team_id = t.id
INNER JOIN
events e ON e.id = et.event_id
WHERE e.id = 2;
```

6. Lista los partidos que se hayan realizado entre el 1900 y el 2010 en los que haya habido al menos un gol.

```
SELECT e.[key] AS EdicionCopa ,
        e.start_at AS FechaPartido ,
        t1.title ,
        g.score1 ,
        g.score2 ,
        t2.title
FROM games g
INNER JOIN
teams t1 ON t1.id = g.team1_id
INNER JOIN
teams t2 ON t2.id = g.team2_id
INNER JOIN
events e ON e.id = g.event_id
WHERE (e.start_at BETWEEN '1900-01-01
_' AND '2010-01-01') AND
( (g.score2 > 0) OR
  (g.score1 > 0) );
```

7. Ya sabemos que los Españoles somos muy buenos al fútbol pero ahora muestra en que ediciones del mundial ha estado la selección española.

```

SELECT DISTINCT e.[key]
  FROM events_teams et,
       events e
  INNER JOIN
    teams t1 ON t1.title = 'Spain'
 WHERE t1.id = et.team_id;

```

8. Lista los partidos en los ha jugado la selección española y no ha empatado.

```

SELECT e.[key] AS EdicionCopa,
       t1.title,
       g.score1,
       g.score2,
       t2.title
  FROM games g
  INNER JOIN
    teams t1 ON t1.id = g.team1_id
  INNER JOIN
    teams t2 ON t2.id = g.team2_id
  INNER JOIN
    events e ON e.id = .event_id
 WHERE ( (t1.title = 'Spain' OR
         t2.title = 'Spain') AND
        (g.score1 != g.score2) );

```

9. Ahora vamos a ver cual ha sido la mayor cantidad de goles que ha conseguido marcar nuestra selección

```

SELECT MAX(MAX(g.score1),
           MAX(g.score2) ) AS maximosGoles
  FROM games g

  INNER JOIN
    teams t1 ON t1.title = 'Spain'
 WHERE t1.id = g.team1_id;

```

10. Muestra el número de veces que cada equipo ha participado en el mundial, ordénalos desde el que más veces haya participado hasta el que menos.

```

SELECT teams.title,
       events_teams.team_id,
       count() AS veces
  FROM teams,
       events_teams
 WHERE teams.id = events_teams.team_id
 GROUP BY teams.title
 ORDER BY veces DESC;

```


11. Los nombres de los equipos que participan en la copa del mundo no siempre coinciden con el nombre del equipo, en una consulta muestra para cada país los equipos que tiene asociados.

```
SELECT c.name AS pais ,
       t.title AS nombre_equipo
FROM teams t
INNER JOIN
     countries c ON c.id
              = t.country_id;
```

12. Muestra en una consulta los resultados de las finales de la copa del mundo. Debes mostrar en la misma consulta la competición que fue, el nombre de los dos equipos que participaban y el resultado final de cada uno de ellos.

```
SELECT e.[key] AS event_key ,
       t1.title AS equipoA ,
       g.score1 AS A_goles_totales ,
       g.score2 AS B_goles_totales ,
       t2.title AS equipoB
FROM games g
INNER JOIN
     teams t1 ON t1.id = g.team1_id
INNER JOIN
     teams t2 ON t2.id = g.team2_id
INNER JOIN
     events e ON e.id = g.event_id
WHERE g.round = "Final";
```

E.3. Ejercicio 3

Sobre la base de datos que se te ha entregado se te pide que crees algunas tablas. Ten en cuenta las posibles limitaciones y la estructura de la base de datos existente. Razona las decisiones que tomes, si tienes algún comentario o aclaración la puedes añadir en el fichero .sql como un comentario.

En los casos que se pida que se inserten datos en las tablas también debemos añadir el código que lo hace en el fichero .sql en el apartado correspondiente.

1. Hay jugadores que han marcado la historia de los mundiales, añade una tabla para almacenar los nombres de estos jugadores. En el caso que puedas también debes dejar espacio para añadir la fecha de nacimiento en caso de que este disponible.

Inserta algún valor en la tabla de tus jugadores favoritos.

```
CREATE TABLE legends (
  id          INTEGER          PRIMARY KEY AUTOINCREMENT
                                NOT NULL,
  name       VARCHAR (255) NOT NULL,
  birth_date DATE
);
```

```
INSERT INTO legends(name, birth_date)
VALUES ( 'Cristiano_Ronaldo', '5/2/1985' );
```

```
INSERT INTO legends(name, birth_date)
VALUES ( 'Antonio_Carbajal', '7/06/1929' );
```

```
INSERT INTO legends(name, birth_date)
VALUES ( 'Dino_Zoff', '28/02/1942' );
```

```
INSERT INTO legends(name, birth_date)
VALUES ( 'Diego_Maradona', '30/08/1960' );
```

2. Pero los jugadores sin sus equipos no serían nada. Ahora queremos saber de nuestras leyendas con que equipos jugaron, crea las tablas necesarias e inserta los datos apropiados en ellas para que podamos consultar esta información.

En este caso hay varias soluciones, esta es la mas sencilla simplemente relacionando la nueva tabla creada con los eventos y con los clubs por separado, se podría relacionar también con la tabla events_teams por id pero esta forma parece mucho más elegante.

```
CREATE TABLE legends_events_teams (
  id          INTEGER PRIMARY KEY AUTOINCREMENT
             NOT NULL,
  event_id   INTEGER NOT NULL,
  team_id    INTEGER NOT NULL,
  legend_id  INTEGER NOT NULL,
  created_at DATETIME DEFAULT (CURRENT_TIMESTAMP),
  updated_at DATETIME DEFAULT (CURRENT_TIMESTAMP)
);
```

```
INSERT INTO legends_events_teams
(legend_id, team_id, event_id)
VALUES (1,138,18);
```

```
INSERT INTO legends_events_teams
(legend_id, team_id, event_id)
VALUES (1,138,19);
```

```
INSERT INTO legends_events_teams
(legend_id, team_id, event_id)
VALUES (1,138,20);
```

```
INSERT INTO legends_events_teams
(legend_id, team_id, event_id)
VALUES (2,190,4);
```

```
INSERT INTO legends_events_teams
(legend_id, team_id, event_id)
VALUES (2,190,5);
```

```
INSERT INTO legends_events_teams
(legend_id, team_id, event_id)
```

```
VALUES (2,190,6);
INSERT INTO legends_events_teams
(legend_id , team_id , event_id)
VALUES (2,190,7);
INSERT INTO legends_events_teams
(legend_id , team_id , event_id)
VALUES (2,190,8);
```

3. Ahora vamos a introducir nuestro nombre en la historia, en este caso vamos a sustituir el nombre de uno de los grandes “Luis Monti” por el nuestro. Introduce por lo tanto tu nombre en la tabla con las leyendas del fútbol pero con los datos de este jugador. Añade también en los equipos en los que participó este jugador.

```
INSERT INTO legends(name, birth_date)
VALUES ( 'David_Gonzalez ' , '15/05/1901 ' );

INSERT INTO legends_events_teams(legend_id , team_id , event_id)
VALUES (5,210,1);

INSERT INTO legends_events_teams(legend_id , team_id , event_id)
VALUES (5,134,2);
```

4. Vamos a mirar nuestro registro de partidos, haz una consulta que muestre todos los partidos que nos corresponden con nuestra nueva identidad futbolística en los datos de las copas del mundo. No es necesario diferenciar entre los partidos en los que jugamos o no, lo importante es participar, mientras que formáramos parte del equipo cuenta para el listado.

```
SELECT l.name AS Nombre,
       e.[key] AS EdicionCopa ,
       g.round ,
       t1.title ,
       g.score1 ,
       g.score2 ,
       t2.title
FROM games g
INNER JOIN
teams t1 ON t1.id = g.team1.id
INNER JOIN
teams t2 ON t2.id = g.team2.id
INNER JOIN
events e ON e.id = g.event_id
INNER JOIN
legends_events_teams let ON
let.event_id = e.id AND
(let.team_id = t1.id OR
let.team_id = t2.id)
INNER JOIN
legends l ON l.id = let.legend_id
```

```
WHERE l.name = "David_Gonzalez";
```

E.4. Ejercicio 4

En este último ejercicio se te pedirá que crees ciertas vistas sobre la base de datos.

1. Crea una vista dónde se muestren todos los partidos jugados, con el nombre del evento y de los dos equipos.

```
CREATE VIEW games_played AS
SELECT e.[key] AS Evento,
       t1.title AS equipoA,
       t2.title AS equipoB
FROM games g
INNER JOIN
teams t1 ON t1.id = g.team1_id
INNER JOIN
teams t2 ON t2.id = g.team2_id
INNER JOIN
events e ON e.id = g.event_id;
```

2. Crea una vista que muestre el número de copas del mundo en las que ha participado cada equipo, una fila por cada equipo en la que una columna ponga el número de veces que ha participado. Ordena el resultado de forma que en la primera línea salga el equipo que más veces ha participado.

```
CREATE VIEW teams_numerWorldscup AS
SELECT count(e.[key]) AS numero_veces,
       t.title AS nombre_equipo,
       t.id AS id_Equipo
FROM teams t
INNER JOIN
events_teams et ON et.team_id = t.id
INNER JOIN
events e ON e.id = et.event_id
GROUP BY t.title
ORDER BY numero_veces DESC;
```

3. Repite la vista anterior pero en este caso en vez de por equipos por países.

```
CREATE VIEW countries_numerWorldscup AS
SELECT count(e.[key]) AS numero_veces,
       c.name AS nombre_pais,
       c.id AS id_pais
FROM countries c
INNER JOIN
teams t ON c.id = t.country_id
INNER JOIN
events_teams et ON et.team_id = t.id
```

```
INNER JOIN
  events e ON e.id = et.event_id
GROUP BY c.name
ORDER BY numero_veces DESC;
```

Bibliografía

Referencias

- [1] Android sqlite package. URL: <http://developer.android.com/reference/android/database/sqlite/package-summary.html>.
- [2] Blob types in mysql manual. URL: <http://dev.mysql.com/doc/refman/5.0/en/blob.html>.
- [3] Db browser for sqlite webpage. URL: <http://sqlitebrowser.org/>.
- [4] Download sqlite. URL: <http://www.sqlite.org/download.html>.
- [5] Manual de sqlite 3. URL: <https://www.sqlite.org/cli.html>.
- [6] Sqlite administrator webpage. URL: <http://sqliteadmin.orbmu2k.de/>.
- [7] Sqlite studio webpage. URL: <http://sqlitestudio.pl/>.
- [8] Users of sqlite. URL: <http://www.sqlite.org/famous.html>.
- [9] Theo Haerder and Andreas Reuter. Principles of transaction-oriented database recovery. *ACM Comput. Surv.*, 15(4):287–317, December 1983. URL: <http://doi.acm.org/10.1145/289.291>, doi:10.1145/289.291.