

Tarea Chica 2

Bases de Datos

Ayudantes Valentina Álvarez y Constanza Olate

vjalvarez@uc.cl clolate@uc.cl

Profesor Denis Parra

Enunciado: 27 de septiembre de 2021

Indicaciones

- Fecha de Entrega: 19:59 hrs el 08 de octubre de 2021.
 - Debes entregar tu tarea en tu repositorio GitHub privado asignado para esta evaluación.
 - La tarea es *individual*. La copia será evaluada con nota 1 en el la tarea, además de las sanciones disciplinarias correspondientes.
 - La resolución de las consultas mediante [consultas anidadas](#) conllevará un descuento de 0.5 en el puntaje de esta.
 - Para realizar las consultas en SQL deberán usar [SQLiteOnline](#), mientras que para realizar el diagrama deberán usar [draw.io](#).
-

Objetivo

El objetivo de esta tarea es que aprendas a:

- Modelar la base de datos de un problema.
- Hacer consultas en una base de datos utilizando SQLite.

Generar repositorio personal

Haciendo click en el siguiente [link](#) con tu cuenta de github, podrás crear tu repositorio personal para realizar la tarea.

Parte I: Diagrama Entidad Relación (3 puntos)

En este vasto multiverso existe la fabulosa DCCLeague de Fútbol. Desde la preciada organización UDCCFA (Unión de Federaciones Asociadas de Fútbol del DCC) te encargan la importantísima misión de realizar un modelo consistente para preservar estadísticas e información relevante de cada partido, jugador y club de un torneo en la DCCLeague. Tu tarea consiste en modelar la información entregada a continuación, para que otra persona pueda implementar una base de datos consistente. La UDCCFA te comunica las siguientes instrucciones:

” Estimado:

- Lo que nos interesa guardar de un Jugador es su nombre, altura, fecha de nacimiento, la posición en que juega y el número que elige en los partidos. Sabemos que naturalmente cada jugador tiene una nacionalidad, pertenecen a un club, juegan partidos y a veces meten goles.
- Los jugadores pertenecen a un Club o Equipo, de los cuales nos interesa poder recordar su nombre, el nombre del estadio cuando juegan de local, el nombre del entrenador y un ranking que es asignado por su respectiva federación.
- Para las escasas ocasiones en que los jugadores hacen Goles en partidos, nos interesa conservar el minuto en que se hizo y la posición en la cancha en que se realizó (por ejemplo, un glorioso gol de media cancha al minuto 89)
- En el caso de los Partidos que juegan los jugadores nos interesa conservar la información de fecha y hora que ocurren, además de guardar la estadística de espectadores por partido.
- Finalmente, nos interesa guardar los nombres de las Ciudades a las que están asociados cada Club, ciudades que a su vez pertenecen a un País.

Saludos, UDCCFA ”

Deberás realizar un diagrama E/R que modele lo descrito anteriormente. Como ayuda, en tu repositorio personal se encuentra el archivo **DiagramaBase.drawio** (debes utilizar draw.io para abrirlo y editarlo), el cual contiene parte del diagrama ya avanzado, de forma que tengas una base para trabajar.

Tu diagrama será evaluado por contener los siguientes elementos:

- Entidades entregadas en DiagramaBase.drawio
- Atributos, subrayando la llave primaria de cada entidad. **(1 punto)**

- Relaciones (1 punto)
- Cardinalidades (1 punto)

La notación a seguir para los elementos anteriores es **estrictamente la vista en clases para un diagrama E/R**, y no se aceptará ningún diagrama que no sea de este tipo (como un diagrama de clases, por ejemplo). Para la notación de cardinalidad pueden ser notación de flecha o con 1:1 N:N 1:N. Finalmente, podrás incluir todos los supuestos adicionales que estimes conveniente para la modelación (como atributos no mencionados explícitamente), **siempre y cuando estos no contradigan al enunciado**. Deberás subir tu diagrama a tu repositorio en un archivo de formato **.pdf** (cualquier archivo con otra extensión no será evaluado). Cualquier supuesto que hayan usado deberá escribirse en el archivo `README.md` presente también en el repositorio o en el **PDF** del diagrama.

Parte II: Consultas SQL (3 puntos)

Después de tanto fútbol, te aburres de los deportes y te interesas más en la lectura. Se te ocurre hacer una página web con información de libros y sus autores, donde distintos usuarios pueden evaluarlos. En tu repositorio personal se encuentra la base de datos **books.db** que tendrás que cargar en SQLiteOnline. El esquema que sigue la base de datos es el siguiente:

Books	
bid	integer PK
title	varchar(50)
price	integer
year	integer
aid	integer

Authors	
aid	integer PK
name	varchar(50)
age	integer
cid	integer

Countries	
cid	integer PK
name	varchar(50)

Users	
uid	integer PK
username	varchar(50)

Reviews	
uid	integer
bid	integer
score	integer
comment	varchar(50)
uid,bid	PK

En la tabla **Books** se guarda el identificador, título, precio y año de publicación de los diferentes libros,


además del identificador del autor que lo escribió. En Authors se guarda el identificador de los autores, su nombre, edad e identificador del país en que nació. En Countries se guarda el identificador y el nombre de cada país. En Users se guarda el identificador y nombre de usuario de los usuarios de la aplicación. Finalmente en Reviews tenemos qué usuario ha hecho evaluado qué libro, con su respectivo puntaje y comentario.

Deberás obtener la siguiente información, mediante una consulta SQL:

1. Entrega el título, precio y año de los libros del autor “George R.R. Martin”. **(0.5 puntos)**.
2. Entrega el título y score de los reviews de los libros cuyo título incluya la palabra “Dog” y el año sea anterior al 2019. **(0.5 puntos)**.
3. Para cada libro al que el usuario “TheDarkKnight” le ha hecho un review, entrega su título, junto con el score y el comentario. **(0.5 puntos)**.
4. Para cada autor, entrega su id, su nombre y la cantidad de libros que ha escrito (renombrar esta columna como “Cantidad de libros”). Ordena la información de mayor a menor y usa limit 5 para encontrar los 5 autores con mayor cantidad de libros. **(0.5 puntos)**.
5. Entrega el identificador y nombre de los países donde el promedio de los scores de los libros de escritores nacidos en ese país es mayor a 4. En el resultado también debe mostrarse el promedio calculado bajo la columna de nombre “Promedio”). **(1 punto)**

Debes entregar un **PDF** con las consultas que hagas en SQL (escritas por ti) y para cada una de ellas debes mostrar un **pantallazo** de su ejecución (la consulta + la tabla resultante) en SQLi-
teOnline. Ejemplo de lo que tienes que entregar en esta parte:

1. SELECT * FROM paises;



The screenshot shows a SQL query editor with the following code:

```
1 SELECT * |
2 FROM paises;
```

Below the editor is a table with the following data:

pid	nombre
1	Italia
2	Francia
3	Bélgica
4	Alemania
5	Inglaterra

Para cargar la base de datos, dirígete a la página de [SQLiteOnline](#), luego haz click en *File*, luego *Open DB* y finalmente selecciona el archivo *books.db*. Para ejecutar las consultas que te pedimos, escribe la consulta en el espacio designado para esto y haz click en *Run*.

Ten en consideración que tus consultas deben funcionar para cualquier instancia de tabla. Por ejemplo, si quitamos o agregamos filas a las tablas de la base de datos, tu consultas deben seguir arrojando una respuesta correcta.

Bonus (0,5 puntos)

Escribe en álgebra relacional la primera consulta.

Bonus del Diagrama ER (0,2 puntos)

Explicar brevemente (en no más de 5 líneas) ¿Por qué Gol debe ser modelado como una entidad y es incorrecto modelarlo como una relación entre jugador y partido?

Bonus Concurso de Memes (0,3 puntos)

1. Como ya saben, para cada tarea tenemos implementado nuestro concurso de memes. Tendrán una bonificación de tres décimas aquellos alumnos que elaboren y envíen los 3 mejores memes dentro del curso. Para hacer envío del meme, junto a su tarea, deben entregar el archivo en formato de imagen JPEG, PNG o GIF; para luego ser evaluado por nuestro comité especializado en memes. El archivo debe tener el siguiente formato: **meme_TC2_apellidos.extensión** y debe estar en su repositorio de la tarea.
2. Nuestro comité de memes tendrá en mente la siguiente rúbrica para la evaluación:
 - Nivel de creatividad
 - Relación con los contenidos de la tarea
 - Sentido de humor
 - Respeto con los compañeros/ayudantes/profesores al usar imágenes de otras personas, siempre verificar la autorización para evitar ofensas.
3. Las decisiones del comité no son apelables.

Entregables

Deberás subir a tu repositorio los siguientes archivos:

- Archivo **PDF** con el Diagrama ER realizado en la parte I. En este archivo puedes incluir la respuesta del bonus del diagrama.
- Otro archivo **PDF** con las consultas de la parte II junto con sus respectivos pantallazos. En este archivo puedes incluir la respuesta del bonus.
- Archivo `README.md` con cualquier supuesto que hayas realizado para la parte I.

Entregas atrasadas

No se aceptan entregas fuera del horario publicado, se revisará la hora del último commit en GitHub. No entregar dentro del horario estipulado es equivalente a no entregar y se califica con la nota mínima 1.0