

Laboratorio 7. Bases de datos

Objetivos:

- Experimentar con la creación de tablas en bases de datos relacionales y la realización de consultas SQL.
- Familiarizarse con el uso del sistema de gestión de bases de datos relacionales MySQL.

Realización:

- Las tareas de esta práctica se realizarán **operando sobre el gestor de bases de datos del servidor personal en la nube (ver la actividad 4.2)**.

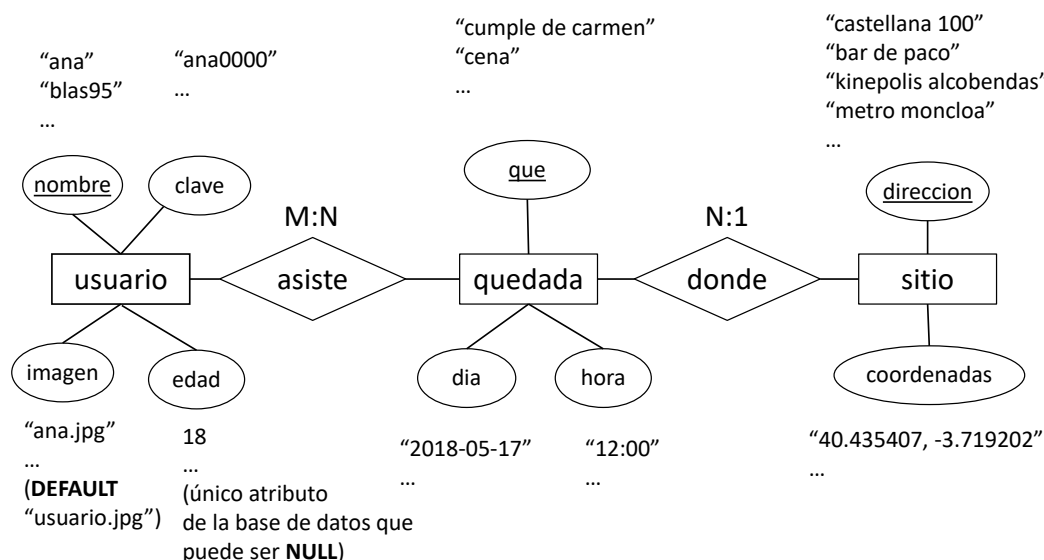
Resultados:

- Entregar el formulario de resultados.
- **En el servidor personal en la nube deberá quedar la base de datos que habrá creado en esta sesión de laboratorio**, ya que se usará más adelante en la práctica final de la asignatura. (Ver nota al final de este enunciado).

Diseño de la base de datos para la práctica final

El servicio telemático de la práctica final de la asignatura maneja una base de datos llamada *misquedadas* con datos de usuarios que pueden asistir a una o varias quedadas. Cada quedada tiene un sitio y el usuario pueden indicar las quedadas a las que piensa asistir. Además de la base de datos que creará en esta práctica de laboratorio, el servicio de la práctica final tiene un interfaz gráfico basado en Web que permite manejar las quedadas y mostrarlas sobre un mapa.

En la figura se muestra el **diagrama Entidad-Relación** de esta base de datos, con ejemplos de valores que pueden tomar los atributos. Cada entrada en la tabla *usuario* se identifica con la clave primaria *nombre* (no se permiten usuarios con el mismo *nombre*). Cada entrada en la tabla *quedada* se identifica con la clave primaria *que* (no se permiten quedadas con el mismo *que*). Cada entrada en la tabla *sitio* se identifica con la clave primaria *direccion* (no se permiten sitios con la misma *direccion*).



Cada entrada en la tabla *asiste* indica que determinado *usuario* asiste a determinada *quedada* mediante dos atributos *nombre* y *que*. Tenga en cuenta que un usuario puede asistir a varias quedadas y que a una quedada pueden asistir varios usuarios.

Una quedada se realiza en un único sitio y en un sitio pueden organizarse varias quedadas.

Los tipos a utilizar son: *int* para *edad*, *date* para *dia*, *time* para *hora* y *varchar(30)* para el resto de atributos.

Ninguno de los atributos de la base de datos puede ser NULL a excepción de la *edad* del usuario.

Tarea 1. Acceso al servidor personal en la nube y al SGBD MySQL

El Sistema de Gestión de Bases de Datos (SGBD) MySQL con el que tiene que crear su base de datos está disponible en su servidor personal en la nube, cuyo manejo se explicó en la actividad 4.2. Una vez arrancado el servidor personal en la nube y conectado al mismo, acceda al SGBD instalado en el servidor personal en la nube escribiendo:

```
mysql -uroot -p
```

Tras teclear la contraseña *ftel*, aparecerá la invitación (prompt) de MySQL

```
mysql>
```

quedando a la espera de que se introduzcan sentencias SQL que deben terminar con el signo de puntuación ";" antes del fin de línea. En caso de no ponerlo, MySQL no dará ninguna respuesta, esperando ese ";" final. Es indiferente usar mayúsculas o minúsculas en las sentencias.

Tarea 2. Creación de la base de datos *misquedadas*

Para crear una base de datos se usa una sentencia *CREATE DATABASE*. Para que sea compatible con el resto de componentes de la práctica final **es necesario** que el nombre que demos a la base de datos sea exactamente *misquedadas*. La sentencia que deberá escribir será, por tanto:

```
mysql> create database misquedadas;  
Query OK, 1 row affected
```

En **color más suave** se muestra el resultado que da MySQL.

Para comprobar que la base de datos se ha creado, escriba la sentencia *SHOW DATABASES*:

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| misquedadas |  
| ... |  
| ... |  
+-----+  
... rows in set
```

Como pueden existir otras bases de datos que ya estaban creadas antes de definir la nuestra, es necesario indicar al SGBD MySQL qué base de datos se va a usar mediante la sentencia *USE*:

```
mysql> use misquedadas;  
Database changed
```

Tarea 3. Definición y creación de las tablas

Siguiendo el diagrama E-R anterior y las indicaciones del apartado “Diseño de la base de datos para la práctica final”, cree las tablas que forman la base de datos usando sentencias *CREATE TABLE*.

Como se ha dicho, la base de datos forma parte de la práctica final y debe haber una compatibilidad total con el resto de componentes de la práctica final para que el servicio funcione. Por ello, debe **respetar estrictamente los nombres de los atributos y los tipos de datos indicados previamente**.

Como ejemplo se muestra la sentencia que crea la tabla *usuario*:

```
mysql> CREATE TABLE usuario (nombre varchar(30), clave varchar(30) NOT NULL,
edad int, imagen varchar(30) NOT NULL DEFAULT "usuario.jpg", PRIMARY KEY
(nombre));
Query OK, 0 rows affected (0.17 sec)
```

No olvide indicar las claves foráneas (FOREIGN KEY... REFERENCES...) en las tablas que lo necesiten.

3.1. Copie al formulario de resultados las sentencias *CREATE TABLE* que necesite para crear las tablas de la base de datos

Puede comprobar la estructura de las tablas que acaba de crear mediante la sentencia *DESCRIBE*, por ejemplo:

```
mysql> describe usuario;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nombre | varchar(30)   | NO   | PRI | NULL    |       |
| clave  | varchar(30)   | NO   |     | NULL    |       |
| edad   | int(11)       | YES  |     | NULL    |       |
| imagen | varchar(30)   | NO   |     | usuario.jpg |     |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

3.2. Copie al formulario el resultado de la sentencia *DESCRIBE* para cada una de las tablas creadas.

3.3. En total debe haber creado **cuatro tablas**. Justifíquelo en el formulario de resultados, explicando brevemente cuáles son los elementos del diagrama E-R de la práctica que se deben transformar en tablas, cuáles no y por qué.

3.4. Copie al formulario de resultados lo que muestra MySQL con la sentencia *SHOW TABLES*.

Tarea 4. Introducción de valores

Utilice sentencias *INSERT INTO* para introducir valores concretos en las cuatro tablas, de forma que haya **3 entradas válidas en cada tabla como mínimo**.

Además de las consideraciones hechas en el apartado “Diseño de la base de datos para la práctica final” debe tener en cuenta lo siguiente:

No se puede hacer referencia a valores indefinidos. Por ejemplo, antes de introducir en la tabla *quedada* una quedada que tendrá lugar en el sitio con *dirección*=“moncloa”, deberá haberse introducido en la tabla *sitio* los datos del sitio con *dirección*=“moncloa”. Si no se hace así, MySQL dará un mensaje de error.

El atributo *clave* indica la contraseña que deberá introducir el usuario para usar el servicio durante la práctica final.

El atributo *imagen* indica el nombre de un fichero que contendrá una foto o imagen del usuario. No hace falta que los ficheros existan todavía. Se crearán más adelante en la práctica final. Los nombres de fichero elegidos ahora para completar este apartado de la práctica pueden modificarse más adelante cuando use la base de datos para la práctica final.

Cada valor del atributo *coordenadas* es un texto que contiene una pareja de coordenadas, latitud y longitud, en grados decimales separadas por una coma, por ejemplo “40.453096, -3.687849”.

Debe respetar el formato de los valores de fecha y hora para los atributos de tipo *date* y *time*.

Los valores de atributos introducidos para completar este apartado de la práctica pueden modificarse más adelante cuando vaya a usar la base de datos para la práctica final.

4.1. Copie al formulario de resultados todas las sentencias *INSERT INTO* que haya utilizado para rellenar las tablas.

Tarea 5. Consultas de comprobación

Utilice sentencias *SELECT* para obtener el contenido completo de cada una de las tablas y compruebe si los datos introducidos cumplen todas las reglas indicadas más arriba.

Si observa algún error puede utilizar las sentencias *REPLACE*, *UPDATE*, *DELETE*, *TRUNCATE* para hacer correcciones. Si se ha equivocado al definir el tamaño de algún campo puede actualizarlo usando la sentencia *ALTER*. En el apéndice final se explica el uso de estas sentencias.

5.1. Copie al formulario el resultado de las sentencias *SELECT* para cada una de las tablas creadas.

5.2. Escriba una sentencia *SELECT* para realizar la consulta **que se indique en la tarea correspondiente a su grupo de clase**. Copie al formulario dicha sentencia y el resultado que da MySQL al ejecutarla.

Cierre y copias de seguridad de la base de datos creada

Una vez rellenado el formulario con las cuestiones planteadas en la práctica, debe cerrar la base de datos escribiendo *EXIT* o *QUIT*.

```
mysql> exit
```

Aunque la base de datos *misquedadas* permanece en su servidor personal en la nube, puede crear una copia de seguridad (respaldo) de la base de datos en un fichero. Para ello, debe ejecutar en su servidor personal en la nube la orden:

```
ftel@...:~$ mysqldump -uroot -pftel -B --opt misquedadas > mqdb-backup.sql
```

y a continuación bajar a su cuenta local en el laboratorio el fichero de respaldo que ha creado (en este ejemplo se supone que el fichero se ha llamado *mqdb-backup.sql*).

Si necesitara recuperar la base de datos, debe subir el fichero de respaldo *mqdb-backup.sql* a su servidor personal en la nube y allí ejecutar la orden:

```
ftel@...:~$ mysql -uroot -pftel misquedadas < mqdb-backup.sql
```

Apéndice. Resumen de sentencias SQL y tipos de datos utilizables en la práctica

- Aunque en las sentencias se puede usar indistintamente mayúsculas o minúsculas, se ponen en mayúsculas los términos que deben escribirse tal como aparecen en la tabla.
- Los términos en minúsculas deben sustituirse por lo que corresponda en cada caso, por ejemplo `tab` debe sustituirse por el nombre concreto de la tabla, `cx` por el nombre de un campo, etc.
- Los corchetes [] indican partes opcionales.
- Ver más detalles en las transparencias de la asignatura.

Sentencia	Utilidad
ALTER TABLE <code>tab</code> DROP COLUMN <code>cx</code> ;	Elimina en la tabla <code>tab</code> el campo <code>cx</code>
ALTER TABLE <code>tab</code> CHANGE <code>cviejo</code> <code>cnuevo</code> <code>tiponuevo</code> ;	Cambia un campo <code>cviejo</code> de la tabla <code>tab</code> y le da otro nombre <code>cnuevo</code> y otro tipo <code>tiponuevo</code>
CREATE DATABASE <code>db</code> ;	Crea una nueva base de datos de nombre <code>db</code>
CREATE TABLE <code>tab</code> (<code>c1</code> <code>tipo1</code> , <code>c2</code> <code>tipo2</code> , ..., PRIMARY KEY (<code>cp</code> , ...) [<code>,</code> FOREIGN KEY (<code>c</code>) REFERENCES <code>tab2</code> (<code>c</code>), ...]);	Crea una tabla <code>tab</code> con campos <code>c1</code> , <code>c2</code> , ... indicando los que forman la clave primaria y, en caso de tenerlas, las claves foráneas hacia otras tablas
DELETE FROM <code>tab</code> WHERE <code>condicion</code> ;	Borra de la tabla <code>tab</code> las filas que cumplan la <code>condicion</code> indicada en <code>WHERE</code>
DESCRIBE <code>tab</code> ;	Muestra la estructura de la tabla de nombre <code>tab</code>
DROP TABLE <code>tab1</code> , <code>tab2</code> , ...;	Borra la tabla o tablas indicadas
INSERT INTO <code>tab</code> VALUES (<code>val1</code> , <code>val2</code> , <code>val3</code> , ...);	Inserta una nueva fila en la tabla <code>tab</code> poniendo en los respectivos campos los valores <code>val1</code> , <code>val2</code> , ...
REPLACE INTO <code>tab</code> VALUES (<code>val1</code> , <code>val2</code> , <code>val3</code> , ...);	Si ya existía esa fila la sustituye por la nueva, si no hace lo mismo que <code>INSERT</code>
SELECT * FROM <code>tab</code> ;	Muestra el contenido completo de la tabla <code>tab</code>
SELECT <code>c1</code> , <code>c2</code> , ... FROM <code>tab</code> ;	Muestra los valores de <code>c1</code> , <code>c2</code> , ... de la tabla <code>tab</code>
SELECT <code>tab1.cx</code> , <code>tab2.cy</code> , <code>tab2.cz</code> , ... FROM <code>tab1</code> , <code>tab2</code> , ... WHERE <code>condicion</code> [ORDER BY <code>tab2.cz</code> , ...] ;	Muestra los valores de los campos de una o varias tablas que cumplan la <code>condicion</code> especificada y, opcionalmente, los ordena por uno o varios campos indicados con <code>ORDER BY</code> (<code>ORDER BY</code> puede añadirse también en los dos ejemplos de <code>SELECT</code> anteriores)
SHOW DATABASES;	Muestra todas las bases de datos ya definidas
SHOW TABLES;	Muestra las tablas de la base de datos en uso
TRUNCATE <code>tab</code> ;	Vacía totalmente la tabla <code>tab</code>
UPDATE <code>tab</code> SET <code>c1=val1</code> , <code>c2=val2</code> , ... WHERE <code>condicion</code> ;	Modifica los valores de campos <code>c1</code> , <code>c2</code> , ... en las filas de la tabla <code>tab</code> que cumplan la <code>condicion</code>
USE <code>db</code> ;	Selecciona la base de datos de nombre <code>db</code> como la que se va a usar a partir de ese momento

Tipos de datos	Valor que toma
INT	número entero (sin comillas)
VARCHAR(<i>nc</i>)	cadena de número variable de caracteres entre 0 y <i>nc</i> (delimitada por comillas)
DATE	fecha en formato "aaaa-mm-dd" (deben incluirse las comillas)
TIME	la hora en formato "hh:mm" (deben incluirse las comillas)

Nota: en caso de problemas

Según se ha dicho en apartados anteriores del enunciado, la base de datos se crea en su servidor personal en la nube, donde quedará guardada para usarla más adelante al hacer la práctica final de la asignatura. Los enunciados de las actividades 4.1 y 4.2 explican paso a paso cómo arrancar su servidor personal en la nube, conectarse a él para ejecutar el comando “mysql” en el servidor, enviar ficheros del ordenador del laboratorio al servidor personal y viceversa, etc. En el Moodle también hay un enlace “FAQ sobre el acceso al servidor personal en la nube” que da respuesta a las dudas más frecuentes.

La documentación citada debe leerse con detalle y se deben seguir las instrucciones de acceso al servidor personal paso a paso sin saltar ninguno. Si aun así encuentra problemas, puede usar el “Foro de preguntas” de la asignatura en Moodle para intercambiar problemas y soluciones entre los propios estudiantes.

En último lugar puede preguntar directamente al profesorado por e-mail a la dirección “ftel@dit.upm.es”. En este caso, para que podamos responder es **imprescindible** que indique claramente en qué paso de las instrucciones ha encontrado el problema, adjuntando una captura de pantalla del último paso que haya hecho correctamente y otra captura que muestre el paso que ha fallado y el mensaje de error que da el sistema.

Si la pregunta no se refiere al servidor personal en la nube sino que se refiere a dudas sobre el contenido que debe tener la base de datos o sobre el lenguaje SQL, debe consultar directamente al profesor de su grupo de clase.

Si no hay más remedio y para evitar que los problemas con el servidor personal impidan entregar a tiempo el formulario de resultados de la práctica, puede crear la base de datos en un ordenador que ya tenga instalado MySQL, o descargarlo e instalarlo desde <https://dev.mysql.com/downloads/mysql/>, rellenar y entregar vía Moodle el formulario de resultados (incluyendo la tarea específica de su grupo de clase que se indicará en la sesión de laboratorio) y guardar una copia de su base de datos como se ha explicado más arriba. En los días siguientes (y, en todo caso, antes de la práctica final) puede resolver los problemas de acceso al servidor personal y crear la base de datos en dicho servidor.

Tener instalado MySQL en su ordenador, además de servir para preparar la práctica 7, es útil para practicar con comandos SQL de cara al examen de la asignatura.