

## 1 Ejercicios

### 1.1 Teoría

1. Define brevemente el significado de las siglas SQL.  
S = Structured  
Q = Query  
L = Lenguaje  
Lenguaje de consulta estructurado.
2. El lenguaje SQL se divide en tres partes DDL, DML y DCL.  
DDL: Data definition language.  
DML: Data manipulation language.  
DCL: Data control language.
3. Defina brevemente qué es MySQL Server.  
Es un sistema de gestión de bases de datos relacionales de código abierto.
4. Defina brevemente qué es MySQL WorkBench.  
Está es una herramienta visual e integrada, desarrollada por mysql que nos permite diseñar, modelar, administrar y desarrollar bases de datos mysql de manera gráfica.
5. Defina brevemente qué es phpMyAdmin.  
Es una herramienta de software gratuita, escrita en php que nos permite administrar bases de datos mysql y mariadb a través de una interfaz web.
6. Defina brevemente el significado de LAMP Stack.  
Es una plataforma de desarrollo web de código abierto.
7. Define para qué sirven las siguientes palabras reservadas de la creación de las tablas en MySQL:
  - AUTO\_INCREMENT: Genera automáticamente valores numéricos secuenciales para una columna.
  - CHARACTER SET: Define el conjunto de caracteres utilizado en una tabla.
  - COLLATE: Especifica la forma en que se comparan y ordenan los datos de texto en la tabla.
  - ENGINE: Determina el motor de almacenamiento utilizado por la tabla.
8. ¿Qué diferencias existen entre los motores de almacenamiento InnoDB y MyISAM de MySQL? ¿Cuál de los dos es el que se usa por defecto?  
InnoDB y MyISAM son motores de almacenamiento de MySQL que difieren principalmente en que InnoDB soporta transacciones, claves foráneas y bloqueo a nivel de fila, mientras que MyISAM no lo hace y usa bloqueo a nivel de tabla. InnoDB ofrece mayor seguridad y recuperación ante fallos, mientras que MyISAM es más rápido en consultas de solo lectura. Desde MySQL 5.5, InnoDB es el motor por defecto.

9. ¿Qué diferencia existe entre los sets de caracteres utf8 y utf8mb en MySQL?  
La diferencia entre utf8 y utf8mb4 en MySQL es que utf8 solo admite hasta 3 bytes por carácter, lo que excluye algunos símbolos Unicode, mientras que utf8mb4 admite hasta 4 bytes y representa el conjunto completo de caracteres Unicode, utf8mb4 es el conjunto recomendado actualmente.
10. ¿Qué diferencia hay entre los tipos de datos VARCHAR y CHAR?  
VARCHAR almacena cadenas de longitud variable, ahorrando espacio, mientras que CHAR almacena cadenas de longitud fija, rellena con espacios si es necesario. VARCHAR es más eficiente para textos de longitud cambiante.
11. ¿Qué diferencia hay entre los tipos de datos TEXT y BLOB?  
TEXT se usa para almacenar grandes cantidades de texto, mientras que BLOB almacena datos binarios. Ambos no se indexan completamente y no permiten valores por defecto.
12. ¿Qué diferencia hay entre los tipos de datos DATETIME y TIMESTAMP?  
DATETIME almacena fecha y hora sin zona horaria y es independiente del sistema, mientras que TIMESTAMP se ajusta automáticamente al huso horario del servidor y puede actualizarse automáticamente con la hora actual. TIMESTAMP ocupa menos espacio, pero tiene un rango más limitado de fechas.
13. ¿Qué diferencia hay entre ENUM y SET?  
ENUM permite almacenar un solo valor de una lista predefinida, mientras que SET permite almacenar varios valores de esa lista en un solo campo. ENUM es útil para opciones exclusivas, y SET para múltiples selecciones.
14. ¿Qué tipo de dato usarías si necesitas almacenar un número real exacto sin redondeo?  
Para almacenar un número real exacto sin redondeo, se debe usar el tipo de dato DECIMAL, ideal para valores financieros o precisos.
15. Explica brevemente para qué sirven las cláusulas ON DELETE y ON UPDATE.  
Las cláusulas ON DELETE y ON UPDATE definen cómo se comportan las claves foráneas cuando se elimina o actualiza un registro relacionado en otra tabla. Sirven para mantener la integridad referencial automáticamente.
16. Explica brevemente para qué sirven las siguientes opciones que podemos usar junto a las cláusulas ON DELETE y ON UPDATE:
- RESTRICT: Impide eliminar o actualizar si hay registros relacionados.
  - CASCADE: Propaga la eliminación o actualización a los registros relacionados.
  - SET NULL: Establece en NULL el valor de la clave foránea relacionada.
  - NO ACTION: Igual que RESTRICT, pero depende del motor de almacenamiento.
17. ¿Qué diferencia existe entre TRUNCATE y DELETE?  
TRUNCATE elimina todos los registros de una tabla de forma rápida y no registra cada fila eliminada, mientras que DELETE puede eliminar registros de forma selectiva y sí registra cada eliminación. Además, TRUNCATE reinicia los valores auto incrementables, y no puede usarse con claves foráneas activas.

18. ¿Para qué se utiliza el atributo UNIQUE?

El atributo UNIQUE se utiliza para garantizar que los valores en una columna (o combinación de columnas) sean únicos, es decir, que no se repitan.  
Se usa comúnmente para asegurar la integridad de datos como correos electrónicos o identificadores.

19. ¿Qué diferencia hay entre INDEX y KEY? ¿Para qué se utilizan?

INDEX y KEY son sinónimos en MySQL; ambos se usan para crear índices que aceleran las búsquedas y consultas en una tabla. Permiten que MySQL localice datos más eficientemente, especialmente en columnas utilizadas en condiciones WHERE o JOIN.

20. ¿Para qué se utiliza el atributo ZEROFILL?

El atributo ZEROFILL rellena con ceros a la izquierda el valor numérico hasta alcanzar el ancho especificado, útil para mostrar valores con formato fijo.

21. ¿Qué diferencia existe entre declarar una columna como INT (11) y INT (11) ZEROFILL?

INT (11) define solo la visualización del número (no su rango), y no tiene efecto sin ZEROFILL; mientras que INT (11) ZEROFILL mostrará los valores con ceros a la izquierda para llenar los 11 dígitos. Funcionalmente ambos almacenan el mismo valor, pero el segundo afecta cómo se presenta.

22. ¿Qué significan las llaves y los corchetes en la siguiente descripción de notación?

```
1 CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] nombre_base_datos;
```

- Las llaves con la barra nos indican que debemos elegir una sola opción ya sea database o schema.
- Los corchetes nos indican que el contenido es opcional “if not exists” lo podemos omitir sin causar errores.

23. ¿Qué comando tendría que escribir desde un terminal para conectarnos con el root con contraseña 1234, a un MySQL Server remoto que se está ejecutando en una máquina con la IP 10.10.1.5?

El comando que debemos emplear para conectarnos es “mysql – h 10.10.1.5 -u root -p1234”.

24. ¿Qué puerto es el que utiliza por defecto el servicio de MySQL Server?

El puerto que MySQL Server utiliza por defecto es el 3306.

25. Cuando instalamos MySQL Server está configurado por defecto para qué sólo acepte conexiones locales y que no permita conexiones remotas. ¿Qué valor tendríamos que poner en la directiva de configuración bind-address para permitir conexiones remotas?

```
1 [mysqld]
2 bind-address = 127.0.0.1
```

1 [mysqld] 2 bind-address = 0.0.0.0, este valor permite que el servidor MySQL escuche en todas las interfaces de red, es decir, acepta conexiones desde cualquier dirección IP externa.

26. Los usuarios de MySQL Server se almacenan en las tablas `mysql.user`. La clave primaria de esta tabla está formada por los valores `user` y `host`, de modo que cada fila identificada por nombre de usuario y el host desde el que puede conectarse. Según la siguiente tabla, ¿desde qué máquinas se podrá conectar el usuario `root` ‘%’?

1	+	-----+	-----+
2		user	host
3	+	-----+	-----+
4		root	%
5		root	localhost
6		debian-sys-maint	localhost
7		mysql.session	localhost
8		mysql.sys	localhost
9	+	-----+	-----+

El carácter `%` es un comodín que representa cualquier dirección IP o nombre de host. Por lo tanto, el usuario `root` definido con `host = %` puede conectarse desde cualquier máquina remota, siempre y cuando:

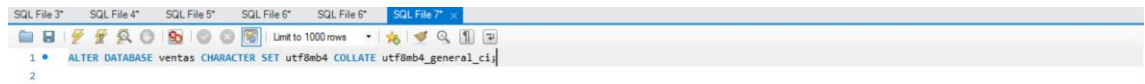
- El servidor MySQL lo permita.
- Las credenciales sean correctas.
- El puerto 3306 por defecto esté accesible.

## 1.2 Práctica con MySQL

Ejecute el siguiente script para crear una base de datos llamada `ventas`.

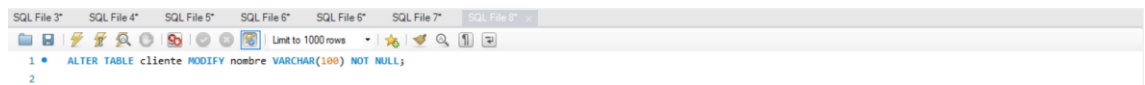
```
1 DROP DATABASE IF EXISTS ventas;
2 CREATE DATABASE ventas CHARACTER SET latin1;
3 USE ventas;
4
5 CREATE TABLE cliente (
6   id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
7   nombre VARCHAR(25),
8   primer_apellido VARCHAR(15) NOT NULL,
9   ciudad VARCHAR(100),
10  categoria INT UNSIGNED
11 );
12
13 CREATE TABLE comercial (
14   id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
15   nombre VARCHAR(100) NOT NULL,
16   apellidol VARCHAR(100) NOT NULL,
17   apellido2 VARCHAR(100),
18   ciudad VARCHAR(100),
19   comision FLOAT
20 );
```

1. Una vez ejecutado el script y ha creado la base de datos en MySQL Server, ¿qué sentencia SQL debería ejecutar para modificar el encoding de la base de datos y utilizarla utf8mb4 en lugar de latin1?

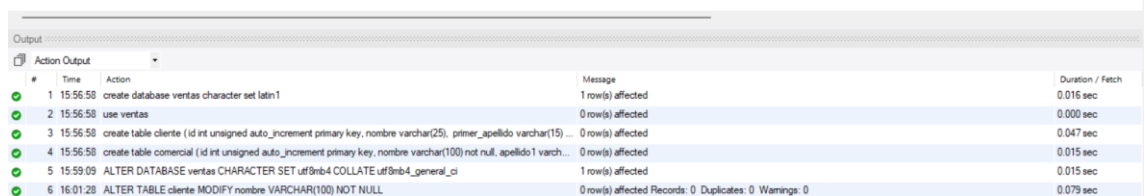


```
1 ALTER DATABASE ventas CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
2
```

2. Modifique la columna nombre de la tabla cliente para que pueda almacenar cadenas de hasta 100 caracteres y para que no pueda ser NULL.



```
1 ALTER TABLE cliente MODIFY nombre VARCHAR(100) NOT NULL;
2
```



#	Time	Action	Message	Duration / Fetch
1	15:56:58	create database ventas character set latin1	1 row(s) affected	0.016 sec
2	15:56:58	use ventas	0 row(s) affected	0.000 sec
3	15:56:58	create table cliente (id int unsigned auto_increment primary key, nombre varchar(25), primer_apellido varchar(15) ...	0 row(s) affected	0.047 sec
4	15:56:58	create table comercial (id int unsigned auto_increment primary key, nombre varchar(100) not null, apellido1 varchar...	0 row(s) affected	0.015 sec
5	15:59:09	ALTER DATABASE ventas CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci	1 row(s) affected	0.015 sec
6	16:01:28	ALTER TABLE cliente MODIFY nombre VARCHAR(100) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.079 sec

3. ¿Qué comando puede ejecutar para comprobar que el cambio que se ha realizado en el paso anterior se ha ejecutado correctamente?

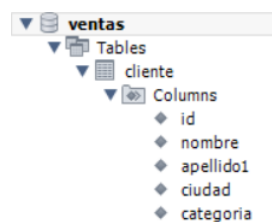
The screenshot shows the SQL Studio interface with the command `DESCRIBE cliente;` entered in the SQL File 9\* tab. The Result Grid displays the structure of the `cliente` table:

Field	Type	Null	Key	Default	Extra
id	int unsigned	NO	PRI	auto_increment	
nombre	varchar(100)	NO			
primer_apellido	varchar(15)	NO			
ciudad	varchar(100)	YES			
categoria	int unsigned	YES			

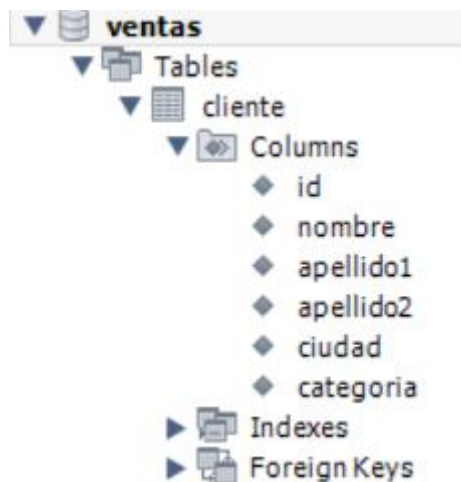
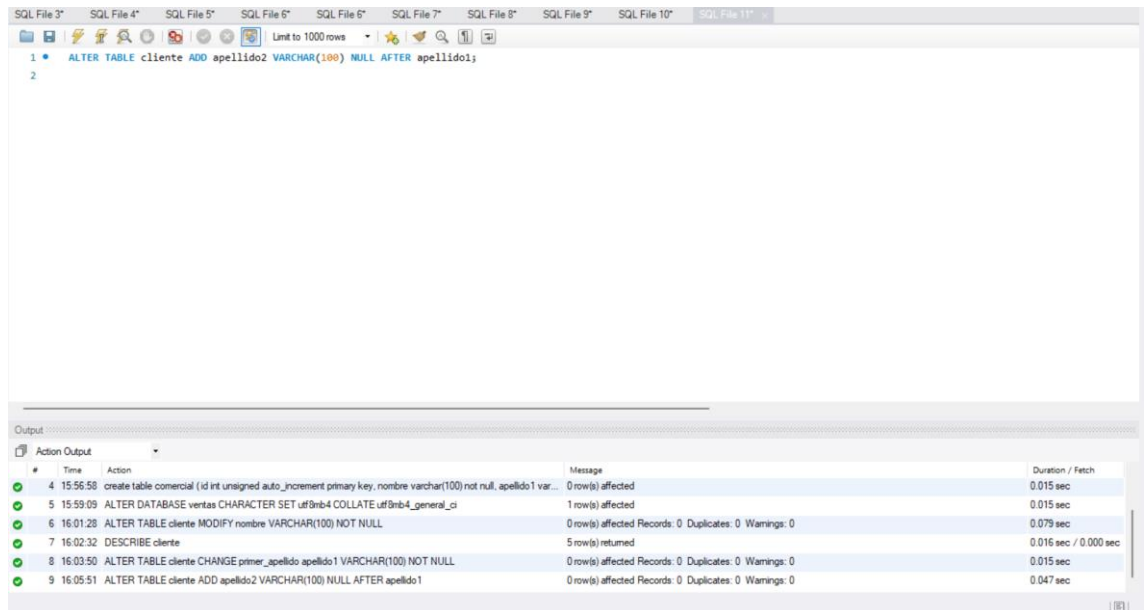
The Action Output pane shows the execution log, including the `DESCRIBE cliente` command which returned 5 rows.

4. Modifica el nombre de la columna `primer_apellido` y asígnele `apellido1`. También tendrá que permitir que pueda almacenar hasta 100 caracteres y que no pueda ser un valor NULL.

The screenshot shows the SQL Studio interface with the command `ALTER TABLE cliente CHANGE primer_apellido apellido1 VARCHAR(100) NOT NULL;` entered in the SQL File 10\* tab. The Action Output pane shows the execution log, including the `ALTER TABLE` command which affected 0 rows.



5. Añada una nueva columna a la tabla cliente para poder almacenar el segundo apellido. La columna se debe llamar apellido2 debe estar entre la columna apellido1 y ciudad, puede almacenar hasta 100 caracteres y puede ser NULL.



6. Elimine la columna categoría de la tabla cliente.

The screenshot shows an SQL IDE interface with a toolbar at the top containing icons for file operations, execution, and search. Below the toolbar, a SQL script is entered in a text area:

```
1 ALTER TABLE cliente DROP COLUMN categoria;
2
```

Below the script, the 'Output' panel is visible, showing a table of execution results:

#	Time	Action	Message	Duration / Fetch
5	15:59:09	ALTER DATABASE ventas CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci	1 row(s) affected	0.015 sec
6	16:01:28	ALTER TABLE cliente MODIFY nombre VARCHAR(100) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.079 sec
7	16:02:32	DESCRIBE cliente	5 row(s) returned	0.016 sec / 0.000 sec
8	16:03:50	ALTER TABLE cliente CHANGE primer_apellido apellido1 VARCHAR(100) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
9	16:05:51	ALTER TABLE cliente ADD apellido2 VARCHAR(100) NULL AFTER apellido1	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
10	16:07:21	ALTER TABLE cliente DROP COLUMN categoria	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

7. Modifique la columna comisión de la tabla comercial para que almacene por defecto el valor 10.

The screenshot shows the same SQL IDE interface. The SQL script in the text area is now:

```
1 ALTER TABLE comercial MODIFY comision FLOAT DEFAULT 10;
2
```

The 'Output' panel shows the execution results, including the new command:

#	Time	Action	Message	Duration / Fetch
6	16:01:28	ALTER TABLE cliente MODIFY nombre VARCHAR(100) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.079 sec
7	16:02:32	DESCRIBE cliente	5 row(s) returned	0.016 sec / 0.000 sec
8	16:03:50	ALTER TABLE cliente CHANGE primer_apellido apellido1 VARCHAR(100) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
9	16:05:51	ALTER TABLE cliente ADD apellido2 VARCHAR(100) NULL AFTER apellido1	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
10	16:07:21	ALTER TABLE cliente DROP COLUMN categoria	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
11	16:10:13	ALTER TABLE comercial MODIFY comision FLOAT DEFAULT 10	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec