

Resolución de los ejercicios

Para llevar a cabo estos desafíos, debes tener abierto **MySQL Workbench**.

Te proporcionamos, en las siguientes *slides*, una solución modelo que puedes usar como referencia.



Resolución del ejercicio 1

- Crear una base de datos con el nombre **BONUS_TRACK**.

```
CREATE DATABASE BONUS_TRACK;
```

- Poner en uso la base de datos generada en el paso anterior.

```
USE BONUS_TRACK;
```



- Crear una **tabla** con el nombre “**AGENDA**” dentro de la base de datos.
- Comentar el **código SQL** generado para la creación de la tabla “**AGENDA**”.

```
CREATE TABLE AGENDA (
    IDCONTACTO INT UNSIGNED PRIMARY KEY,
    NOMBRE VARCHAR(20) NOT NULL,
    APELLIDO VARCHAR(15) NOT NULL,
    DOMICILIO VARCHAR(50),
    TELÉFONO INT NOT NULL)
;
```

```
/*CREATE TABLE AGENDA (
    IDCONTACTO INT UNSIGNED PRIMARY KEY,
    NOMBRE VARCHAR(20) NOT NULL,
    APELLIDO VARCHAR(15) NOT NULL,
    DOMICILIO VARCHAR(50),
    TELÉFONO INT NOT NULL)
;*/
```

- Visualizar las **tablas existentes** en la base de datos para verificar la creación de la tabla “AGENDA”.

```
SHOW TABLES;
```

- Visualizar la **estructura** de la tabla “AGENDA”

```
DESC AGENDA;
```

- Agregar el **campo MAIL** a la tabla. Este campo deberá contener como **máximo 50 caracteres** y su **carga es obligatoria**.

```
ALTER TABLE AGENDA  
ADD MAIL VARCHAR(50) NOT NULL;
```

- Insertar **4 registros** en la tabla “AGENDA”.

```
INSERT INTO AGENDA
VALUES      (1, 'LUCIANO', 'FERNANDEZ', 'MARTIN RODRIGUEZ 1271', 1156485201,
'L.FERNANDEZ@GMAIL.COM'),
            (2, 'SEBASTIÁN', 'GARCÍA', 'LARREA 567', 1152421100,
'SEABSTIAN.GARCIA@HOTMAIL.COM'),
(3, 'CAROLINA', 'SANCHEZ', 'AV. CORRIENTES 2233 - 4ºB', 1132295741,
'CARO.GARCIA@GMAIL.COM'),
(4, 'SABRINA', 'DIAZ', 'CONSTITUCIÓN 3220 - 2ºA', 1128541025,
'SABRI.DIAZ@HOTMAIL.COM')
;
```

Resolución del ejercicio 2

1. Crear una base de datos con el nombre **“LIBRERÍA”**. Poner en uso la base de datos generada.
2. Una vez creadas todas las tablas, insertar un **conjunto de registros** en cada una de ellas, utilizando el archivo con el nombre **“SCRIPT”**. Este archivo está disponible en la sección *Descargas*.

| Continúa en las diapositivas siguientes.

```
CREATE DATABASE LIBRERÍA;
USE LIBRERÍA;

CREATE TABLE autores(
    autor_id varchar(11),
    apellido varchar(40),
    nombre varchar(20),
    telefono varchar(12),
    direccion varchar(40),
    ciudad varchar(20),
    provincia char(2),
    c_postal char(5),
    estado tinyint(1)
);
```

(Continuación):

```
CREATE TABLE ventas(
    local_id char(4),
    factura_nro varchar(20),
    fecha datetime,
    cantidad smallint,
    forma_pago varchar(12),
    libro_id varchar(6)
);
```

```
CREATE TABLE locales(
    local_id char(4),
    nombre varchar(40),
    direccion varchar(40),
    ciudad varchar(20),
    provincia char(2),
    c_postal char(5)
);
```

(Continuación):

```
CREATE TABLE editoriales(  
    editorial_id char(4),  
    nombre varchar(40),  
    ciudad varchar(20),  
    provincia char(2),  
    pais varchar(30)  
);
```

```
CREATE TABLE libros(  
    libro_id varchar(6),  
    titulo varchar(80),  
    categoria char(12),  
    editorial_id char(4),  
    precio double,  
    comentarios varchar(200),  
    fecha_publicacion datetime  
);
```

(Continuación):

```
CREATE TABLE libroautor(  
    autor_id varchar(11),  
    libro_id varchar(6)  
);  
  
CREATE TABLE puestos(  
    puesto_id smallint,  
    descripcion varchar(50)  
);
```

```
CREATE TABLE empleados(  
    empleado_id char(9),  
    nombre varchar(20),  
    apellido varchar(30),  
    puesto_id smallint,  
    editorial_id char(4),  
    fecha_ingreso datetime  
);
```



Resolución del ejercicio 3

- A partir de lo realizado en el Ejercicio 2, luego de insertar los registros, se debe definir **claves primarias (primary key)** en **cada tabla**.
- Examinar los datos ingresados, e **insertar 3 registros** en cada una de las tablas utilizando la sentencia **INSERT** del conjunto de sentencias **DML (Data Manipulation Language)** del lenguaje SQL.
- Tener en cuenta que cada tabla tiene un campo con el sufijo **_id** que identifica al campo clave.



```
alter table autores add primary key(autor_id);
alter table puestos add primary key(puesto_id);
alter table empleados add primary key(empleado_id);
alter table libroautor add primary key(autor_id, libro_id);
alter table locales add primary key(local_id);
alter table editoriales add primary key(editorial_id);
alter table libros add primary key(libro_id);
alter table ventas add primary key(local_id, factura_nro, libro_id);
```

Resolución del ejercicio 4

- Mostrar todo el **contenido** de la tabla ***TOP_SPOTIFY***.

```
SELECT * FROM TOP_SPOTIFY;
```

- En el resultado de la consulta, solo se deben observar las columnas ***ARTISTA***, ***TÍTULO*** y

```
SELECT ARTISTA, TITULO, GENERO FROM TOP_SPOTIFY;
```



- Ordenar **alfabéticamente** el resultado de la consulta según los **géneros musicales**.

```
SELECT ARTISTA, TITULO, GENERO FROM TOP_SPOTIFY ORDER BY GENERO;
```

- En el caso de aquellos géneros que se repiten, ordenar **alfabéticamente** los **nombres de los artistas**.

```
SELECT ARTISTA, TITULO, GENERO FROM TOP_SPOTIFY  
ORDER BY GENERO, ARTISTA;
```

Resolución del ejercicio 5

- Con base en el ejercicio anterior, **mostrar todos los registros de la tabla TOP_SPOTIFY**. En el resultado, solo se deben observar las columnas **ARTISTA**, **TÍTULO** y **GÉNERO**. Ordenar **alfabéticamente** el resultado según los nombres de los **artistas** y el nombre de las **canciones**. Mostrar únicamente las **10** primeras canciones.

```
SELECT ARTISTA, TITULO, GENERO FROM TOP_SPOTIFY  
ORDER BY ARTISTA, TITULO LIMIT 10;
```



- Modificar la consulta anterior para mostrar únicamente las canciones ubicadas desde la **posición 11 hasta la 15 inclusive**.

```
SELECT ARTISTA, TITULO, GENERO FROM TOP_SPOTIFY  
ORDER BY ARTISTA, TITULO LIMIT 5 OFFSET 10;
```

- Dada la tabla **TOP_SPOTIFY**, obtener una lista de todas aquellas canciones pertenecientes a la cantante llamada **Madonna**. Debes mostrar **todos los campos** de la tabla en el resultado de la consulta.

```
SELECT * FROM TOP_SPOTIFY WHERE ARTISTA = 'MADONNA';
```

- A partir de la tabla **TOP_SPOTIFY**, obtener una lista de todas aquellas canciones que pertenecen al género **Pop**. Mostrar **todos los campos** de la tabla en el resultado de la consulta y ordenar **alfabéticamente** el resultado según el nombre de las **canciones**.

```
SELECT * FROM TOP_SPOTIFY WHERE GENERO = 'POP' ORDER BY TITULO;
```



- De la tabla ***TOP_SPOTIFY***, obtener una lista de todas las canciones pertenecientes al género ***Pop*** lanzadas durante el **año 2015**. Mostrar **todos los campos** de la tabla en el resultado de la consulta y ordenar dicho resultado **alfabéticamente** según los nombres de los **artistas** y los nombres de las **canciones**.

```
SELECT * FROM TOP_SPOTIFY WHERE GENERO = 'POP'  
AND ANO = 2015  
ORDER BY ARTISTA, TITULO;
```

- A partir de la tabla **TOP_SPOTIFY**, obtener una lista de todas aquellas canciones lanzadas **antes del año 2011** y que pertenezcan al género **Dance Pop**.
Mostrar **todos los campos** de la tabla en el resultado de la consulta y ordenar dicho resultado **alfabéticamente** según los nombres de las **canciones**.

```
SELECT * FROM TOP_SPOTIFY  
WHERE ANO < 2011 AND GENERO = 'DANCE POP'  
ORDER BY TITULO;
```



¡Terminaste el módulo!

Todo listo para rendir el examen