

ANÁLISIS Y DESARROLLO DE SOFTWARE FICHA 3065273

**ACTIVIDAD: BASES DE DATOS RELACIONALES CASO DE
ESTUDIO TIENDA**

**INTEGRANTES:
KEVIN MONTAÑEZ
MIGUEL LOZANO**

**INSTRUCTOR:
LUIS FELIPE RESTREPO**

**CENTRO DESARROLLO AGROINDUSTRIAL Y
EMPRESARIAL SENA VILLETA**

1.Creación Base de datos

The screenshot shows the MySQL Workbench interface. The title bar says "Servidor: 127.0.0.1". Below it is a tab bar with "Bases de datos", "SQL", "Estado actual", and "Cue...". A large text area contains the SQL command: "CREATE DATABASE FestivalMusica;".

```
1 CREATE DATABASE FestivalMusica;
```

2.Creación de tablas

The screenshot shows the MySQL Workbench interface with the database "festivalmusica" selected. The title bar says "Base de datos: festivalmusica". Below it is a tab bar with "Estructura", "SQL", "Buscar", and "Generar una consulta". A large text area contains the SQL command to create the "escenario" table.

```
1 CREATE TABLE escenario (
2 id_Escenario INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,
3 nombre VARCHAR(40) NOT NULL,
4 capacidad INT(8) NOT NULL);
```

The screenshot shows the MySQL Workbench interface with the database "festivalmusica" selected. The title bar says "Base de datos: festivalmusica". Below it is a tab bar with "Estructura", "SQL", "Buscar", "Generar una consulta", and "Exp...". A large text area contains the SQL command to create the "presentacion" table.

```
1 CREATE TABLE presentacion(
2 id_Presentacion INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,
3 id_Artista INT(6),
4 id_Escenario INT (6),
5 fecha DATE NOT NULL,
6 hora TIME NOT NULL,
7 FOREIGN KEY (id_Artista) REFERENCES artistas(id_Artista),
8 FOREIGN KEY (id_Escenario) REFERENCES escenario(id_Escenario));
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 CREATE TABLE clientes(
2 id_Cliente BIGINT(12) PRIMARY KEY AUTO_INCREMENT,
3 nombre VARCHAR(50) NOT NULL,
4 correo VARCHAR(50) NOT NULL,
5 ciudad VARCHAR(40) NOT NULL);
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta Exportar [?](#)

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 CREATE TABLE compraEntrada(
2 id_Compra INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,
3 id_Cliente BIGINT(12),
4 id_Presentacion INT(6),
5 cantidad INT(3) NOT NULL,
6 total DECIMAL(10,2) NOT NULL,
7 FOREIGN KEY (id_Cliente) REFERENCES clientes(id_Cliente),
8 FOREIGN KEY (id_Presentacion) REFERENCES presentacion(id_Presentacion));
```

3.Modificación de tablas

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 ALTER TABLE artistas ADD COLUMN instagram VARCHAR(100);
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 ALTER TABLE escenario ADD COLUMN ubicacion VARCHAR(50);
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 ALTER TABLE clientes ADD COLUMN telefono BIGINT(12);
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 ALTER TABLE clientes DROP COLUMN telefono;
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta Exportar Imprimir

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 CREATE TABLE compraEntrada (
2     id_Compra INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,
3     id_Cliente BIGINT(12),
4     id_Presentacion INT(6),
5     cantidad INT(3),
6     total DECIMAL(10,2),
7     FOREIGN KEY (id_Cliente) REFERENCES clientes(id_Cliente),
8     FOREIGN KEY (id_Presentacion) REFERENCES presentacion(id_Presentacion));
```

4.Insertar Datos

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: ⚡

```
1 INSERT INTO artistas(nombre,genero,pais) VALUES
2 ('Imagine Dragons','Rock Alternativo','EE.UU.'),
3 ('Karol G','Reggaeton','Colombia'),
4 ('ColdPlay','Poprock','Reino Unido'),
5 ('J Balvin','Urbano','Colombia'),
6 ('Dua Lipa','Pop','Reino Unido'),
7 ('Bad Bunny','Trap Latino','Puerto Rico'),
8 ('Artic Monkeys','Indie Rock','Reino Unido'),
9 ('Sebastian Yatra','Pop Latino','Colombia'),
10 ('The Weeknd','R&B','Canada'),
11 ('Rauw Alejandro','Reggaeton','Puerto Rico'),
12 ('Foo Fighters','Rock Altetnativo','EE.UU'),
13 ('BLACKPINK','K-Pop','Corea del Sur');
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: ⚡

```
1 INSERT INTO escenario(nombre,capacidad,ubicacion) VALUES
2 ('Main Stage',25000,'Zona Norte'),
3 ('Beats Stage',20000,'Zona Centro'),
4 ('Rock Arena',18000,'Zona Oeste'),
5 ('Latin Vibes',15000,'Zona Este'),
6 ('Indie Garden',12000,'Zona Sur'),
7 ('Global sounds',10000,'Zona Internacional');
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta Exportar

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: ⚡

```
1 INSERT INTO presentacion(id_Artista,id_Escenario,fecha,hora) VALUES
2 (1,3,'2024-04-10','20:00');
```

Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta Exportar

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 INSERT INTO clientes(nombre,correo,ciudad) VALUES
2 ('Laura Martinez','lauramartinez@gmail.com','Medellin'),
3 ('Andres Gomez','andresgomez@gmail.com','Cali'),
4 ('Camila Torres','camilatorres@gmail.com','Barranquilla'),
5 ('Diego Rodrigues','diegorodrigues@gmail.com','Cartagena'),
6 ('Mariana Lopes','marianalopez@gmail.com','Bogota'),
7 ('Sebastian Castro','sebastiancastro@gmail.com','Medellin'),
8 ('Valentina Herrera','valentinaherrera@gmail.com','Bucaramanga'),
9 ('Jorge Morales','jorgemorales@gmail.com','Bogota'),
10 ('Ana Ruiz','anaruiz@gmail.com','Cali'),
11 ('Felipe Ramirez','feliperamirez@gmail.com','Medellin'),
12 ('Daniela Jimenez','danielajimenez@gmail.com','Bogota'),
13 ('Alejandro Rojas','alejandrorojas@gmail.com','Cartagena'),
14 ('Sofia Martinez','sofiamartinezqgmai1com','Bucaramanga'),
15 ('Pablo Fernandez','pablofernandez@gmail.com','Barranquilla'),
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta Exportar

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 INSERT INTO presentacion(id_Artista,id_Escenario,fecha,hora) VALUES
2 (2,4,'2024-07-10','18:00'),
3 (3,1,'2024-07-11','21:00'),
4 (4,4,'2024-07-11','19:00'),
5 (5,1,'2024-07-12','20:30'),
6 (6,2,'2024-07-12','22:00'),
7 (7,5,'2024-07-13','18:30'),
8 (9,6,'2024-07-13','21:00');
```

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta Exportar Imprimir

Ejecutar la(s) consulta(s) SQL en la base de datos festivalmusica: [?](#)

```
1 INSERT INTO compraentrada(id_Cliente,id_Presentacion,cantidad,total) VALUES
2 (1001001001,1,2,300000.00),
3 (1001001002,2,1,150000.00),
4 (1001001003,3,3,450000.00),
5 (1001001004,4,2,300000.00),
6 (1001001005,5,1,180000.00),
7 (1001001006,1,1,150000.00),
8 (1001001007,2,2,300000.00),
9 (1001001008,3,1,150000.00),
10 (1001001009,4,3,450000.00),
11 (1001001010,5,2,360000.00),
12 (1001001011,6,1,180000.00),
13 (1001001012,7,2,300000.00),
14 (1001001013,8,1,150000.00),
15 (1001001014,1,3,450000.00),
```

5. Consultas de SELECT Y WHERE (15 ejercicios)

1. Muestra el nombre y género de los artistas cuyo género es "Pop".

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: artistas

Examinar Estructura SQL Buscar Insertar Exportar

Mostrar ventana de consultas SQL

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0008 segundos.)

```
SELECT artistas.nombre, artistas.genero FROM artistas WHERE genero = 'Pop';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

nombre	genero
Dua Lipa	Pop

Editar Copiar Borrar

SELECT artistas.nombre, artistas.genero FROM artistas WHERE genero = 'Pop';

2. Muestra el nombre, género y país de los artistas ordenados alfabéticamente por nombre

Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0004 segundos.) [nombre: ARTIC MONKEYS... - THE WEEKND...]

```
SELECT artistas.nombre, artistas.genero, artistas.pais FROM artistas ORDER BY artistas.nombre ASC;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ningun

Opciones extra

nombre	genero	pais
Artic Monkeys	Indie Rock	Reino Unido
Bad Bunny	Trap Latino	Puerto Rico
BLACKPINK	K-Pop	Corea del Sur
ColdPlay	Poprock	Reino Unido
Dua Lipa	Pop	Reino Unido
Foo Fighters	Rock Alternativo	EE.UU.
Imagine Dragons	Rock Alternativo	EE.UU.
J Balvin	Urbano	Colombia
Karol G	Reggaeton	Colombia
Rauw Alejandro	Reggaeton	Puerto Rico
Sebastian Yatra	Pop Latino	Colombia
The Weeknd	R&B	Canada

SELECT artistas.nombre, artistas.genero, artistas.pais FROM artistas ORDER BY artistas.nombre ASC;

3.Muestra el nombre y capacidad de los escenarios, ordenados de mayor a menor capacidad.

✓ Mostrando filas 0 - 5 (total de 6, La consulta tardó 0,0008 segundos.) [capacidad: 25000... - 10000...]

```
SELECT escenario.nombre, escenario.capacidad FROM escenario ORDER BY capacidad DESC;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 ▾ Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

nombre	capacidad
Main Stage	25000
Beats Stage	20000
Rock Arena	18000
Latin Vibes	15000
Indie Garden	12000
Global sounds	10000

SELECT escenario.nombre, escenario.capacidad FROM escenario ORDER BY capacidad DESC;

4.Muestra los escenarios cuya capacidad esté entre 10,000 y 20,000, ordenados de forma ascendente por capacidad.

✓ Mostrando filas 0 - 4 (total de 5, La consulta tardó 0,0004 segundos.) [capacidad: 10000... - 20000...]

```
SELECT escenario.nombre, escenario.capacidad FROM escenario WHERE escenario.capacidad BETWEEN 10000 AND 20000 ORDER BY escenario.capacidad ASC;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 ▾ Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna ▾

Opciones extra

nombre	capacidad
Global sounds	10000
Indie Garden	12000
Latin Vibes	15000
Rock Arena	18000
Beats Stage	20000

SELECT escenario.nombre, escenario.capacidad FROM escenario WHERE escenario.capacidad BETWEEN 10000 AND 20000 ORDER BY escenario.capacidad ASC;

5.Muestra el nombre, correo y ciudad de los clientes que viven en "Bogotá", ordenados alfabéticamente por nombre.

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** festivalmusica
- Table:** clientes
- Query Result:**
 - Showing rows 0 - 8 (total of 9). The query took 0,0013 seconds.
 - SQL Query: `SELECT clientes.nombre, clientes.correo, clientes.ciudad FROM clientes WHERE ciudad = 'Bogota' ORDER BY nombre ASC;`
 - Buttons: Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
 - Table Headers: nombre, correo, ciudad
 - Table Data:

	nombre	correo	ciudad
<input type="checkbox"/>	Daniela Jimenez	danielajimenez@gmail.com	Bogota
<input type="checkbox"/>	Isabella Gonzales	isabellagonzales@gmail.com	Bogota
<input type="checkbox"/>	Jorge Morales	jorgemorales@gmail.com	Bogota
<input type="checkbox"/>	Juan Perez	juanperez@gmail.com	Bogota
<input type="checkbox"/>	Kevin Fernandez	kevinfernandez@gmail.com	Bogota
<input type="checkbox"/>	Maicol Bernal	maicolbernal@gmail.com	Bogota
<input type="checkbox"/>	Mariana Lopes	marianolopez@gmail.com	Bogota
<input type="checkbox"/>	Schneider Manrique	schneidermanrique@gmail.com	Bogota
<input type="checkbox"/>	Sofia Camelo	sofiacamelo@gmail.com	Bogota
- Table Options:** Mostrar todo, Número de filas: 25, Filtrar filas: Buscar en esta tabla, Ordenar según la clave: Ninguna
- Extra Options:** Opciones extra

`SELECT clientes.nombre, clientes.correo, clientes.ciudad FROM clientes WHERE ciudad = 'Bogota' ORDER BY nombre ASC;`

6.Muestra el nombre, correo y ciudad de los clientes que viven en "Cali" o "Barranquilla".

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** festivalmusica
- Table:** clientes
- Query Result:**
 - Showing rows 0 - 8 (total of 9). The query took 0,0002 seconds.
 - SQL Query: `SELECT clientes.nombre, clientes.correo, clientes.ciudad FROM clientes WHERE clientes.ciudad IN ('cali', 'barranquilla');`
 - Buttons: Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
 - Table Headers: nombre, correo, ciudad
 - Table Data:

	nombre	correo	ciudad
<input type="checkbox"/>	Andres Gomez	andresgomez@gmail.com	Cali
<input type="checkbox"/>	Camila Torres	camilatorres@gmail.com	Barranquilla
<input type="checkbox"/>	Ana Ruiz	anaruiz@gmail.com	Cali
<input type="checkbox"/>	Pablo Fernandes	pablofernandez@gmail.com	Barranquilla
<input type="checkbox"/>	Maria Fernanda Perez	mariaperez@gmail.com	Cali
<input type="checkbox"/>	Valeria Castillo	valeriacastillo@gmail.com	Barranquilla
<input type="checkbox"/>	Valentina Camacho	valentinacamacho@gmail.com	Cali
<input type="checkbox"/>	Jorge Torres	jorgetorres@gmail.com	Cali
<input type="checkbox"/>	Luis Lozano	luislozano@gmail.com	Cali
 - Table Options:** Mostrar todo, Número de filas: 25, Filtrar filas: Buscar en esta tabla, Ordenar según la clave: Ninguna
 - Extra Options:** Opciones extra

```
SELECT clientes.nombre, clientes.correo, clientes.ciudad FROM clientes WHERE clientes.ciudad IN ('cali', 'barranquilla');
```

7. Muestra los nombres de los clientes cuyo nombre empiece con la letra "A".

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: clientes

Examinar Estructura SQL Buscar Insertar Exportar

Mostrar ventana de consultas SQL

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0007 segundos.)

```
SELECT clientes.nombre FROM clientes WHERE nombre LIKE 'A%';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

nombre
Andres Gomez
Ana Ruiz
Alejandro Rojas
Albert Gutierrez

```
SELECT clientes.nombre FROM clientes WHERE nombre LIKE 'A%';
```

8. Muestra los clientes cuyo correo contenga la palabra "music".

NINGUNO

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0002 segundos.)

```
SELECT clientes.nombre, clientes.correo FROM clientes WHERE clientes.correo LIKE '%music%';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

nombre correo

Operaciones sobre los resultados de la consulta

Crear vista

```
SELECT clientes.nombre, clientes.correo FROM clientes WHERE clientes.correo LIKE '%music%';
```

9. Muestra los artistas cuyo género sea "Rock Alternativo" o "Indie Rock", ordenados alfabéticamente por nombre.

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Servidor 127.0.0.1, Base de datos: festivalmusica, Tabla: artistas, Examinar, Estructura, SQL, Buscar, Insertar, Exportar, Importar, Privilegios.
- Message Bar:** Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0012 segundos.) [nombre: ARTIC MONKEYS... - IMAGINE DRAGONS...]
- SQL Editor:** SELECT * FROM artistas WHERE genero = 'Rock Alternativo' OR genero = 'Indie Rock' ORDER BY nombre ASC;
- Table View:** Shows two rows of artist data:

	id_Artista	nombre	genero	pais	instagram
<input type="checkbox"/>	7	Artic Monkeys	Indie Rock	Reino Unido	NULL
<input type="checkbox"/>	1	Imagine Dragons	Rock Alternativo	EE.UU.	NULL
- Filter and Order Options:** Mostrar todo, Número de filas: 25, Filtrar filas: Buscar en esta tabla, Ordenar según la clave: Ninguna.

SELECT * FROM artistas WHERE genero = 'Rock Alternativo' OR genero = 'Indie Rock'
ORDER BY nombre ASC;

10. Muestra las presentaciones programadas para el día "2024-07-12", ordenadas de menor a mayor hora.

The screenshot shows the MySQL Workbench interface with the following details:

- Message Bar:** Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0003 segundos.) [hora: 22:00:00... - 20:30:00...]
- SQL Editor:** SELECT presentacion.id_Presentacion, presentacion.fecha, presentacion.hora FROM presentacion WHERE presentacion.fecha = '2024-07-12' ORDER BY presentacion.hora DESC;
- Table View:** Shows two rows of presentation data:

	id_Presentacion	fecha	hora
<input type="checkbox"/>	6	2024-07-12	22:00:00
<input type="checkbox"/>	5	2024-07-12	20:30:00
- Filter and Order Options:** Mostrar todo, Número de filas: 25, Filtrar filas: Buscar en esta tabla, Ordenar según la clave: Ninguna.

SELECT presentacion.id_Presentacion, presentacion.fecha, presentacion.hora FROM presentacion WHERE presentacion.fecha = '2024-07-12'
ORDER BY presentacion.hora DESC;

11. Muestra los nombres de los artistas que no son de "Colombia".

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: artistas

Examinar Estructura SQL Buscar Insertar

Mostrar ventana de consultas SQL

Mostrando filas 0 - 8 (total de 9, La consulta tardó 0,0014 segundos.)

```
SELECT artistas.nombre FROM artistas WHERE pais != 'Colombia';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [

Mostrar todo | Número de filas: 25 ▾ Filtrar filas: Buscar e

Opciones extra

	nombre
<input type="checkbox"/>	Imagine Dragons
<input type="checkbox"/>	ColdPlay
<input type="checkbox"/>	Dua Lipa
<input type="checkbox"/>	Bad Bunny
<input type="checkbox"/>	Artic Monkeys
<input type="checkbox"/>	The Weeknd
<input type="checkbox"/>	Rauw Alejandro
<input type="checkbox"/>	Foo Fighters
<input type="checkbox"/>	BLACKPINK

```
SELECT artistas.nombre FROM artistas WHERE pais != 'Colombia';
```

12. Muestra el nombre y correo de los clientes, ordenados de forma descendente por ciudad.

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: clientes

Examinar Estructura SQL Buscar Insertar Exportar

Mostrar ventana de consultas SQL

Mostrando filas 0 - 24 (total de 30, La consulta tardó 0,0255 segundos.)

```
SELECT clientes.nombre,clientes.correo FROM clientes ORDER BY ciudad DESC;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

1 > >> | Mostrar todo | Número de filas: 25 | Filtrar filas: Bus

Opciones extra

	nombre	correo
<input type="checkbox"/>	Nicolas Salazar	nicolassalazar@gmail.com
<input type="checkbox"/>	Laura Martinez	lauramartinez@gmail.com
<input type="checkbox"/>	Albert Gutierrez	albertgutierrez@gmail.com
<input type="checkbox"/>	Felipe Ramirez	feliperamirez@gmail.com
<input type="checkbox"/>	Stiven Mendoza	stivenendoza@gmail.com
<input type="checkbox"/>	Sebastian Castro	sebastiancastro@gmail.com
<input type="checkbox"/>	Juan Mahecha	juanmahecha@gmail.com
<input type="checkbox"/>	Diego Rodrigues	diegorodrigues@gmail.com
<input type="checkbox"/>	Santiago Mendoza	santiagoendoza@gmail.com
<input type="checkbox"/>	Alejandro Rojas	alejandrorojas@gmail.com
<input type="checkbox"/>	Valentina Camacho	valentinacamacho@gmail.com
<input type="checkbox"/>	Ana Ruiz	anaruiz@gmail.com
<input type="checkbox"/>	Jorge Torres	jorgetorres@gmail.com
<input type="checkbox"/>	Maria Fernanda Perez	mariaperez@gmail.com
<input type="checkbox"/>	Andres Gomez	andresgomez@gmail.com

SELECT clientes.nombre,clientes.correo FROM clientes ORDER BY ciudad DESC;

13. Muestra las 5 primeras filas de la tabla cliente.

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: clientes

Examinar Estructura SQL Buscar Insertar Exportar Importar

Mostrar ventana de consultas SQL

Mostrando filas 0 - 4 (total de 5, La consulta tardó 0,0066 segundos.)

```
SELECT * FROM clientes LIMIT 5;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Opciones extra

		id_Cliente	nombre	correo	ciudad
<input type="checkbox"/>	Editar Copiar Borrar	1001001001	Juan Perez	juanperez@gmail.com	Bogota
<input type="checkbox"/>	Editar Copiar Borrar	1001001002	Laura Martinez	lauramartinez@gmail.com	Medellin
<input type="checkbox"/>	Editar Copiar Borrar	1001001003	Andres Gomez	andresgomez@gmail.com	Cali
<input type="checkbox"/>	Editar Copiar Borrar	1001001004	Camila Torres	camilatorres@gmail.com	Barranquilla
<input type="checkbox"/>	Editar Copiar Borrar	1001001005	Diego Rodrigues	diegorodrigues@gmail.com	Cartagena

SELECT * FROM clientes LIMIT 5;

14. Muestra los clientes que no viven en "Bogotá", "Medellín" o "Cali".

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: clientes

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios

Mostrar ventana de consultas SQL

Mostrando filas 0 - 7 (total de 8, La consulta tardó 0,0004 segundos.)

```
SELECT clientes.nombre,clientes.ciudad FROM clientes WHERE clientes.ciudad NOT IN('Bogota','Medellin','Cali');
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	nombre	ciudad	
<input type="checkbox"/>	Editar Copiar Borrar	Camila Torres	Barranquilla
<input type="checkbox"/>	Editar Copiar Borrar	Diego Rodrigues	Cartagena
<input type="checkbox"/>	Editar Copiar Borrar	Valentina Herrera	Bucaramanga
<input type="checkbox"/>	Editar Copiar Borrar	Alejandro Rojas	Cartagena
<input type="checkbox"/>	Editar Copiar Borrar	Sofia Martinez	Bucaramanga
<input type="checkbox"/>	Editar Copiar Borrar	Pablo Fernandes	Barranquilla
<input type="checkbox"/>	Editar Copiar Borrar	Santiago Mendoza	Cartagena
<input type="checkbox"/>	Editar Copiar Borrar	Valeria Castillo	Barranquilla

SELECT clientes.nombre,clientes.ciudad FROM clientes WHERE clientes.ciudad NOT IN('Bogota','Medellin','Cali');

15. Muestra los nombres de los artistas cuyo nombre tenga exactamente 2 palabras.

The screenshot shows the phpMyAdmin interface for a database named 'festivalmusica'. The 'artistas' table is selected. A green message at the top indicates 10 rows were found in 0.2603 seconds. The SQL query used is:

```
SELECT artistas.nombre FROM artistas WHERE LENGTH(nombre) - LENGTH(REPLACE(nombre, ' ', '')) = 1;
```

Below the query, there are options to edit, copy, or delete each row. The results show ten artists:

	nombre
<input type="checkbox"/>	Imagine Dragons
<input type="checkbox"/>	Karol G
<input type="checkbox"/>	J Balvin
<input type="checkbox"/>	Dua Lipa
<input type="checkbox"/>	Bad Bunny
<input type="checkbox"/>	Artic Monkeys
<input type="checkbox"/>	Sebastian Yatra
<input type="checkbox"/>	The Weeknd
<input type="checkbox"/>	Rauw Alejandro
<input type="checkbox"/>	Foo Fighters

```
SELECT artistas.nombre FROM artistas WHERE LENGTH(nombre) -  
LENGTH(REPLACE(nombre,' ','')) = 1;
```

6. Consultas UPDATE (5 ejercicios)

1. Modifica el nombre del escenario con idEscenario = 2 a "Beats Arena".

The screenshot shows the phpMyAdmin interface for the 'festivalmusica' database. The 'escenario' table is selected. A green message indicates 1 row was affected in 0.0766 seconds. The SQL query used is:

```
UPDATE escenario SET nombre = 'Beats Arena' WHERE id_Escenario = 2;
```

Below the query, there are options to edit, copy, or create PHP code for each row. The results show one row updated:

	nombre
	Beats Arena

```
UPDATE escenario SET nombre = 'Beats Arena' WHERE id_Escenario = 2;
```

2. Modifica el correo del cliente con idCliente = 1001001002 a

"lauramusic@gmail.com".

```
✓ 1 fila afectada. (La consulta tardó 0,0008 segundos.)  
UPDATE clientes SET clientes.correo = 'lauramusica@gmail.com' WHERE clientes.id_cliente=1001001002;  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

UPDATE clientes SET clientes.correo = 'lauramusica@gmail.com' WHERE clientes.id_Cliente = 1001001002;

3.Modifica la capacidad del escenario con idEscenario = 1 a 30,000.



Mostrar ventana de consultas SQL

```
✓ 1 fila afectada. (La consulta tardó 0,0035 segundos.)
```

```
UPDATE escenario SET capacidad = 30000 WHERE id_Escenario = 1;
```

[Editar en línea] [Editar] [Crear código PHP]

UPDATE escenario SET capacidad = 30000 WHERE id_Escenario = 1;

4.Modifica el género del artista con idArtista = 4 a "Reggaetón".

```
✓ 1 fila afectada. (La consulta tardó 0,0004 segundos.)  
UPDATE artistas SET artistas.genero = 'reggaeton' WHERE artistas.id_Artista = 4;  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

UPDATE artistas SET artistas.genero = 'Reggaeton' WHERE artistas.id_Artista = 4;

5.Modifica el correo de todos los clientes de la ciudad "Bogotá" para que terminen en "@festivalmusic.com".

```
✓ 9 filas afectadas. (La consulta tardó 0,0010 segundos.)  
UPDATE clientes SET correo = CONCAT(SUBSTRING_INDEX(correo,'@',1), '@festivalmusic.com') WHERE ciudad = 'Bogota';  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

UPDATE clientes SET correo =
CONCAT(SUBSTRING_INDEX(correo,'@',1), '@festivalmusic.com') WHERE ciudad =
'Bogota';

7. Consultas DELETE (5 ejercicios)

1.Elimina la compra de entradas con idCompra = 3.

✓ 1 fila eliminada. (La consulta tardó 0,0005 segundos.)

```
DELETE FROM compraentrada WHERE compraentrada.id_Compra=3;
```

[Editar en línea] [Editar] [Crear código PHP]

DELETE FROM compraentrada WHERE compraentrada.id_Compra = 3;

2.Elimina todas las compras realizadas por el cliente con idCliente = 1001001001.

The screenshot shows the MySQL Workbench interface. The title bar says "Servidor: 127.0.0.1 » Base de datos: festivalmusica". Below it, there are tabs for "Estructura", "SQL", "Buscar", and "Generar una consulta". A button "Mostrar ventana de consultas SQL" is visible. The main area displays a success message and the executed SQL query.

✓ 1 fila eliminada. (La consulta tardó 0,0010 segundos.)

```
DELETE FROM compraentrada WHERE id_Cliente = 1001001001;
```

[Editar en línea] [Editar] [Crear código PHP]

DELETE FROM compraentrada WHERE id_cliente = 1001001001;

3.Elimina la presentación con idPresentacion = 8.

The screenshot shows an error message from MySQL Workbench. It includes the error code #1451, the SQL query being run, and the MySQL error message.

Error

consulta SQL: [Copiar](#)

```
DELETE FROM presentacion WHERE presentacion.id_Presentacion=8;
```

MySQL ha dicho: [?](#)

```
#1451 - Cannot delete or update a parent row: a foreign key constraint fails ('festivalmusica`.`compraentrada`, CONSTRAINT `compraentrada_ibfk_2` FOREIGN KEY (`id_Presentacion`) REFERENCES `presentacion` (`id_Presentacion`))
```

DELETE FROM presentacion WHERE presentacion.id_Presentacion = 8;

4.Elimina el artista con idArtista = 10.

← Servidor: 127.0.0.1 » Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta

Mostrar ventana de consultas SQL

✓ 1 fila eliminada. (La consulta tardó 0,0009 segundos.)

```
DELETE FROM artistas WHERE id_Artista = 10;
```

[Editar en línea] [Editar] [Crear código PHP]

DELETE FROM artistas WHERE id_Artista = 10;

5.Elimina todos los clientes que viven en "Cali".

Error

consulta SQL: [Copiar](#)

```
DELETE FROM clientes WHERE clientes.ciudad = 'cali';
```

MySQL ha dicho: [?](#)

```
#1451 - Cannot delete or update a parent row: a foreign key constraint fails (`festivalmusica`.`compraentrada`, CONSTRAINT `compraentrada_ibfk_1` FOREIGN KEY (`id_Cliente`) REFERENCES `clientes` (`id_Cliente`))
```

DELETE FROM clientes WHERE clientes.ciudad = 'Cali';

8.Consultas JOIN (10 ejercicios)

1.Muestra el nombre del cliente, nombre del artista, fecha y hora de la presentación para cada compra de entradas.

✓ Mostrando filas 0 - 28 (total de 29, La consulta tardó 0.0004 segundos.)

```
SELECT clientes.nombre AS nombre_cliente, artistas.nombre AS nombre_artista, presentacion.fecha, presentacion.hora FROM compraentrada JOIN clientes ON compraentrada.id_Cliente = clientes.id_Cliente JOIN presentacion ON compraentrada.id_Presentacion = presentacion.id_Presentacion JOIN artistas ON presentacion.id_Artista = artistas.id_Artista;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: Filtrar filas: Ordenar según la clave:

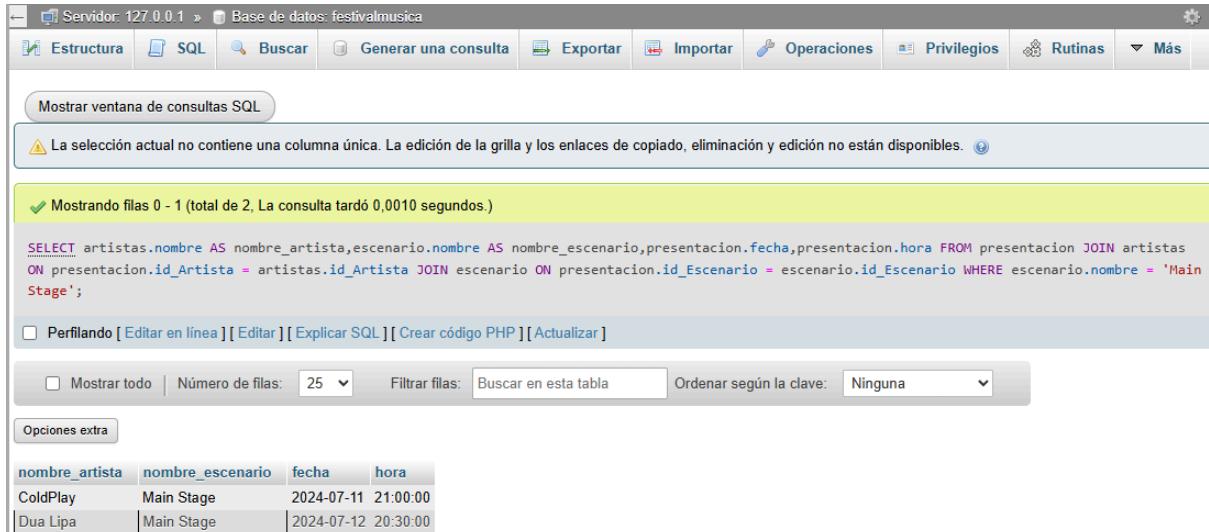
Opciones extra

nombre_cliente	nombre_artista	fecha	hora
Juan Perez	Imagine Dragons	2024-04-10	20:00:00
Mariana Lopes	Imagine Dragons	2024-04-10	20:00:00
Sofia Martinez	Imagine Dragons	2024-04-10	20:00:00
Jorge Torres	Imagine Dragons	2024-04-10	20:00:00
Laura Martinez	Karol G	2024-07-10	18:00:00
Sebastian Castro	Karol G	2024-07-10	18:00:00
Pablo Fernandez	Karol G	2024-07-10	18:00:00
Juan Mahecha	Karol G	2024-07-10	18:00:00
Valentina Herrera	ColdPlay	2024-07-11	21:00:00
Isabella Gonzales	ColdPlay	2024-07-11	21:00:00
Luis Lozano	ColdPlay	2024-07-11	21:00:00
Camila Torres	J Balvin	2024-07-11	19:00:00

SELECT clientes.nombre AS nombre_cliente, artistas.nombre AS nombre_artista, presentacion.fecha, presentacion.hora FROM compraentrada JOIN clientes ON compraentrada.id_Cliente = clientes.id_Cliente

JOIN presentacion ON compraentrada.id_Presentacion = presentacion.id_Presentacion
 JOIN artistas ON presentacion.id_Artista = artistas.id_Artista;

2. Muestra el nombre del artista, nombre del escenario, fecha y hora de las presentaciones que se realizan en el escenario "Main Stage".



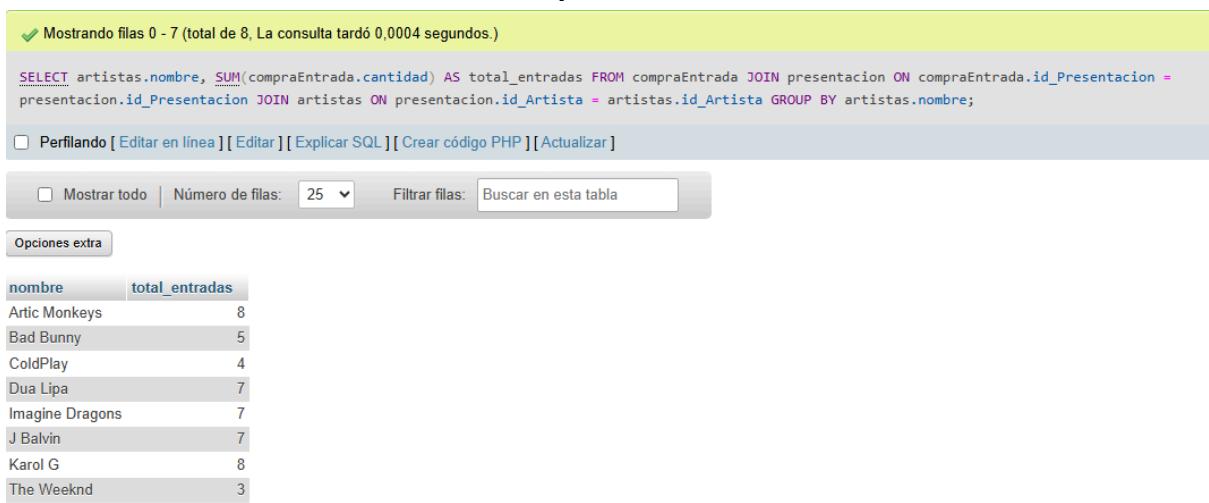
```

SELECT artistas.nombre AS nombre_artista,escenario.nombre AS nombre_escenario,presentacion.fecha,presentacion.hora FROM presentacion JOIN artistas ON presentacion.id_Artista = artistas.id_Artista JOIN escenario ON presentacion.id_Escenario = escenario.id_Escenario WHERE escenario.nombre = 'Main Stage';
  
```

nombre_artista	nombre_escenario	fecha	hora
ColdPlay	Main Stage	2024-07-11	21:00:00
Dua Lipa	Main Stage	2024-07-12	20:30:00

SELECT artistas.nombre AS nombre_artista,escenario.nombre AS nombre_escenario,
 presentacion.fecha,presentacion.hora FROM presentacion
 JOIN artistas ON presentacion.id_Artista = artistas.id_Artista
 JOIN escenario ON presentacion.id_Escenario = escenario.id_Escenario
 WHERE escenario.nombre = 'Main Stage';

3. Muestra el total de entradas vendidas por cada artista.



```

SELECT artistas.nombre, SUM(compraEntrada.cantidad) AS total_entradas FROM compraEntrada JOIN presentacion ON compraEntrada.id_Presentacion = presentacion.id_Presentacion JOIN artistas ON presentacion.id_Artista = artistas.id_Artista GROUP BY artistas.nombre;
  
```

nombre	total_entradas
Artic Monkeys	8
Bad Bunny	5
ColdPlay	4
Dua Lipa	7
Imagine Dragons	7
J Balvin	7
Karol G	8
The Weeknd	3

SELECT artistas.nombre,SUM(compraEntrada.cantidad) AS total_entradas
 FROM compraEntrada JOIN presentacion ON compraEntrada.id_Presentacion =
 presentacion.id_Presentacion JOIN artistas ON presentacion.id_Artista = artistas.id_Artista
 GROUP BY artistas.nombre;

4. Muestra los nombres de los clientes, su ciudad, y la cantidad de entradas que han comprado para cada presentación.

← Servidor: 127.0.0.1 → Base de datos: festivalmusica

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas

Mostrando filas 0 - 24 (total de 29, La consulta tardó 0,0013 segundos.)

```
SELECT clientes.nombre AS nombre_cliente,clientes.ciudad,presentacion.id_Presentacion AS id_presentacion,COUNT(*) AS cantidad_entradas_compradas
FROM clientes JOIN compraentrada ON clientes.id_Cliente = compraentrada.id_Cliente JOIN presentacion ON compraentrada.id_Presentacion =
presentacion.id_Presentacion GROUP BY clientes.nombre,clientes.ciudad,presentacion.id_Presentacion;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

1 > >> Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

nombre_cliente	ciudad	id_presentacion	cantidad_entradas_compradas
Albert Gutierrez	Medellin	6	1
Alejandro Rojas	Cartagena	8	1
Ana Ruiz	Cali	5	1
Andres Gomez	Cali	3	1
Camila Torres	Barranquilla	4	1
Daniela Jimenez	Bogota	7	1
Diego Rodrigues	Cartagena	5	1
Felipe Ramirez	Medellin	6	1
Isabella Gonzales	Bogota	3	1
Jorge Morales	Bogota	4	1
Jorge Torres	Cali	1	1
Juan Mahecha	Medellin	2	1
Kevin Fernandez	Bogota	5	1
Consola	Medellin	2	1

```
SELECT clientes.nombre AS nombre_cliente,clientes.ciudad,presentacion.id_Presentacion
AS id_presentacion,COUNT(*) AS cantidad_entradas_compradas FROM clientes JOIN
compraentrada ON clientes.id_Cliente = compraentrada.id_Cliente JOIN presentacion ON
compraentrada.id_Presentacion = presentacion.id_Presentacion GROUP BY
clientes.nombre,clientes.ciudad,presentacion.id_Presentacion;
```

5. Muestra los nombres de los clientes que han comprado más de 5 entradas en total.

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0004 segundos.)

```
SELECT clientes.nombre, SUM(compraEntrada.cantidad) AS total_entradas FROM compraEntrada JOIN clientes ON compraEntrada.id_Cliente= clientes.id_Cliente
GROUP BY clientes.nombre HAVING total_entradas>5;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

```
SELECT clientes.nombre, SUM(compraEntrada.cantidad) AS total_entradas FROM
compraEntrada JOIN clientes ON compraEntrada.id_Cliente= clientes.id_Cliente GROUP
BY clientes.nombre HAVING total_entradas > 5;
```

6. Muestra la cantidad de presentaciones programadas por cada escenario.

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: escenario

[Examinar](#) [Estructura](#) [SQL](#) [Buscar](#) [Insertar](#) [Exportar](#) [Importar](#) [Privilegios](#) [Operaciones](#)

Mostrando filas 0 - 5 (total de 6, La consulta tardó 0,0009 segundos.)

```
SELECT escenario.nombre,COUNT(presentacion.id_Presentacion) AS cantidad_presentaciones FROM escenario LEFT JOIN presentacion ON escenario.id_Escenario = presentacion.id_Escenario GROUP BY escenario.nombre;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

nombre	cantidad_presentaciones
Beats Stage	1
Global sounds	1
Indie Garden	1
Latin Vibes	2
Main Stage	2
Rock Arena	1

```
SELECT escenario.nombre,COUNT(presentacion.id_Presentacion) AS cantidad_presentaciones FROM escenario LEFT JOIN presentacion ON escenario.id_Escenario = presentacion.id_Escenario GROUP BY escenario.nombre;
```

7.Muestra la cantidad de artistas registrados por cada país.

Mostrando filas 0 - 6 (total de 7, La consulta tardó 0,0003 segundos.)

```
SELECT artistas.pais, COUNT(*) AS total_artistas FROM artistas GROUP BY artistas.pais;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

pais	total_artistas
Canada	1
Colombia	3
Corea del Sur	1
EE.UU	1
EE.UU.	1
Puerto Rico	2
Reino Unido	3

```
SELECT artistas.pais, COUNT(*) AS total_artistas FROM artistas GROUP BY artistas.pais;
```

8.Muestra la suma total de dinero recaudado por cada artista.

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: artistas

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Se

Mostrando filas 0 - 7 (total de 8, La consulta tardó 0,0006 segundos.)

```
SELECT artistas.nombre,SUM(compraentrada.total) AS total_dinero_recaudado FROM artistas JOIN presentacion ON artistas.id_Artista = presentacion.id_Artista JOIN compraentrada ON presentacion.id_Presentacion = compraentrada.id_Presentacion GROUP BY artistas.nombre;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

nombre	total_dinero_recaudado
Artic Monkeys	1200000.00
Bad Bunny	810000.00
ColdPlay	1050000.00
Dua Lipa	1200000.00
Imagine Dragons	750000.00
J Balvin	1050000.00
Karol G	1200000.00
The Weeknd	450000.00

```
SELECT artistas.nombre,SUM(compraentrada.total) AS total_dinero_recaudado FROM artistas JOIN presentacion ON artistas.id_Artista = presentacion.id_Artista JOIN compraentrada ON presentacion.id_Presentacion = compraentrada.id_Presentacion GROUP BY artistas.nombre;
```

9. Muestra el total de compras realizadas por los clientes de la ciudad de "Bogotá".

Mostrando filas 0 - 8 (total de 9, La consulta tardó 0,0015 segundos.)

```
SELECT clientes.nombre, COUNT(compraEntrada.id_Compra) AS total_compras FROM compraEntrada JOIN clientes ON compraEntrada.id_Cliente = clientes.id_Cliente WHERE clientes.ciudad = 'Bogotá' GROUP BY clientes.nombre;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

nombre	total_compras
Daniela Jimenez	1
Isabella Gonzales	1
Jorge Morales	1
Juan Perez	1
Kevin Fernandez	1
Maicol Bernal	1
Mariana Lopes	1
Schneider Manrique	1
Sofia Camelo	1

```
SELECT clientes.nombre, COUNT(compraEntrada.id_Compra) AS total_compras FROM compraEntrada JOIN clientes ON compraEntrada.id_Cliente = clientes.id_Cliente WHERE clientes.ciudad = 'Bogotá' GROUP BY clientes.nombre;
```

10. Muestra la cantidad total de entradas vendidas por cada ciudad de cliente.

Servidor: 127.0.0.1 » Base de datos: festivalmusica » Tabla: clientes

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguir

Mostrando filas 0 - 5 (total de 6, La consulta tardó 0,0006 segundos.)

```
SELECT clientes.ciudad,SUM(compraentrada.cantidad) AS total_entradas_vendidas FROM clientes JOIN compraentrada ON clientes.id_Cliente = compraentrada.id_Cliente GROUP BY clientes.ciudad ORDER BY total_entradas_vendidas DESC;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Buscar en esta tabla

Opciones extra

ciudad	total_entradas_vendidas
Cali	13
Bogota	13
Medellin	11
Barranquilla	6
Bucaramanga	4
Cartagena	3

```
SELECT clientes.ciudad,SUM(compraentrada.cantidad) AS total_entradas_vendidas FROM clientes JOIN compraentrada ON clientes.id_Cliente = compraentrada.id_Cliente GROUP BY clientes.ciudad ORDER BY total_entradas_vendidas DESC;
```