

Taller 4 bases de datos

Juan José García Aguirre

Poblar las tablas:

Para cambiar las tablas, se utiliza principalmente el comando *Insert into tabla*, entonces se poblaron las tablas editorial, libro, autor, cliente, libro_autor, teléfono_clientes:

Editorial:

```
INSERT INTO Editorial
VALUES ('Anagrama', 'Barcelona', 'Calle Bailén', '12345678910'),
      ('Gallimard', 'Paris', '5 rue Sébastien Bottin', '9874563217'),
      ('Viking', 'New york', '375 Hudson St', '1223334444'),
      ('Santillana', 'Madrid', 'Calle de Juan Ignacio', '321456987');
```

Libro:

```
INSERT INTO libro VALUES
('ISBN1', 'Los detectives salvajes', 640, 'Anagrama'),
('ISBN2', '2666', 912, 'Anagrama'), ('ISBN3', 'Rayuela', 656, 'Gallimard'),
('ISBN4', 'Cien años de soledad', 432, 'Gallimard'),
('ISBN5', 'Moby Dick', 720, 'Viking'),
('ISBN6', 'La Odisea', 432, 'Viking'),
('ISBN7', 'Don Quijote de la Mancha', 1056, 'Santillana'),
('ISBN8', 'El Aleph', 224, 'Santillana'),
('ISBN9', 'La ciudad y los perros', 496, 'Anagrama'),
('ISBN10', 'La casa de los espíritus', 496, 'Anagrama'),
('ISBN11', 'En busca del tiempo perdido', 4215, 'Gallimard'),
('ISBN12', 'El extranjero', 160, 'Gallimard'),
('ISBN13', 'La saga/fuga de J. B.', 672, 'Viking'),
('ISBN14', 'El corazón es un cazador solitario', 400, 'Viking'),
('ISBN15', 'Crónica de una muerte anunciada', 128, 'Santillana'),
('ISBN16', 'Cien años de soledad', 432, 'Santillana'),
('ISBN17', 'La insostenible levedad del ser', 352, 'Anagrama'),
('ISBN18', 'El amor en los tiempos del cólera', 512, 'Anagrama'),
('ISBN19', 'Rayuela', 656, 'Gallimard'),
('ISBN20', '2666', 912, 'Gallimard');
```

Autor:

```
INSERT INTO autor VALUES
(1, '1927-03-06', 'Colombiano', 'Gabriel García Márquez'),
(2, '1953-02-28', 'Estadounidense', 'Paul Auster'),
(3, '1942-07-28', 'Chileno', 'Roberto Bolaño'),
(4, '1937-06-28', 'Argentino', 'Juan José Saer'),
(5, '1926-10-15', 'Francesa', 'Michel Foucault');
```

Cliente:

```
INSERT INTO cliente VALUES
('123456789', 'Juan Pérez'),
('987654321', 'María Rodríguez'),
('456123789', 'Luis García'),
('789123456', 'Ana Martínez'),
('321654987', 'Pedro López'),
('456789123', 'Laura González'),
('987321654', 'Carlos Ramírez');
```

Libro_autor:

```
INSERT INTO libro_autor VALUES
('ISBN1', 1),
('ISBN2', 2),
('ISBN3', 3),
('ISBN4', 4),
('ISBN5', 5),
('ISBN6', 1),
('ISBN7', 2),
('ISBN8', 3),
('ISBN9', 4),
('ISBN10', 5);
```

Libro_cliente:

```
INSERT INTO libro_cliente VALUES
('ISBN1', 123456789),
('ISBN8', 123456789),
('ISBN5', 321654987),
('ISBN10', 456123789),
('ISBN3', 456123789),
('ISBN6', 456789123),
('ISBN4', 789123456),
('ISBN7', 987321654),
('ISBN2', 987654321),
('ISBN9', 987654321);
```

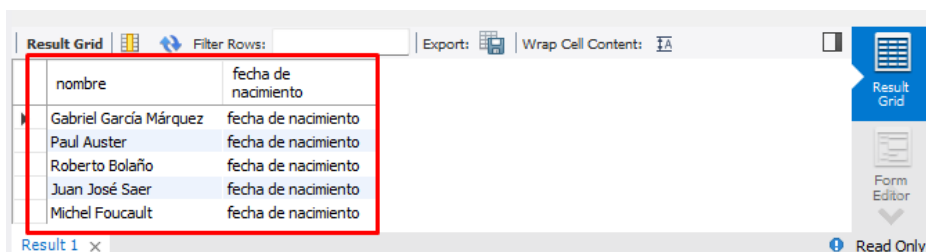
Telefono_clientes:

```
INSERT INTO telefono_cliente VALUES
('123456789', '+1234567890'),
('123456789', '+9876543210'),
('987654321', '+1111111111'),
('987654321', '+2222222222'),
('456123789', '+3333333333'),
('456123789', '+4444444444'),
('789123456', '+5555555555'),
('789123456', '+6666666666'),
('321654987', '+7777777777'),
('321654987', '+8888888888'),
('456789123', '+9999999999'),
('456789123', '+0000000000');
```

Consultas:

Consulta para conocer nombre y fecha de nacimiento de autores

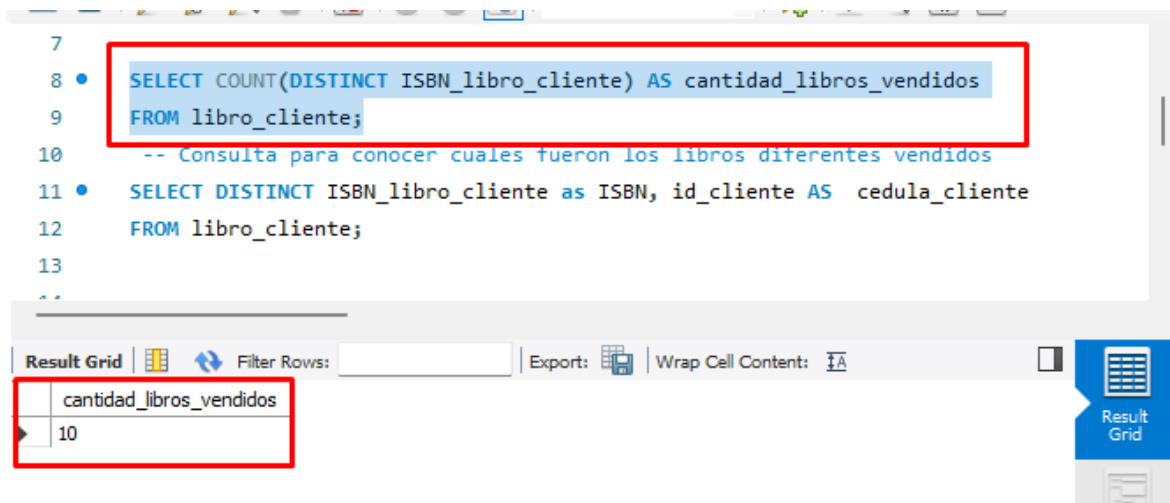
```
-- Consulta para conocer nombre y fecha de nacimiento de autores
SELECT nombre, "fecha de nacimiento"
FROM autor;
```



The screenshot shows a database interface with a 'Result Grid' tab. A table with two columns, 'nombre' and 'fecha de nacimiento', is displayed. The data rows are: Gabriel García Márquez, Paul Auster, Roberto Bolaño, Juan José Saer, and Michel Foucault, all with 'fecha de nacimiento' as the date. A red box highlights the table header and the first row of data.

nombre	fecha de nacimiento
Gabriel García Márquez	fecha de nacimiento
Paul Auster	fecha de nacimiento
Roberto Bolaño	fecha de nacimiento
Juan José Saer	fecha de nacimiento
Michel Foucault	fecha de nacimiento

Consulta para conocer cantidad de libros diferentes vendidos



The screenshot shows a database interface with a SQL editor and a 'Result Grid' tab. The SQL editor contains two queries. The first query, highlighted with a red box, is: `SELECT COUNT(DISTINCT ISBN_libro_cliente) AS cantidad_libros_vendidos FROM libro_cliente;`. The second query is: `SELECT DISTINCT ISBN_libro_cliente as ISBN, id_cliente AS cedula_cliente FROM libro_cliente;`. The 'Result Grid' tab shows a table with one column, 'cantidad_libros_vendidos', and one row with the value '10'. A red box highlights the table header and the data row.

```
7
8 • SELECT COUNT(DISTINCT ISBN_libro_cliente) AS cantidad_libros_vendidos
9   FROM libro_cliente;
10
11 -- Consulta para conocer cuales fueron los libros diferentes vendidos
12 • SELECT DISTINCT ISBN_libro_cliente as ISBN, id_cliente AS cedula_cliente
13   FROM libro_cliente;
14
15
```

cantidad_libros_vendidos
10

Consulta para conocer el nombre de cada cliente acompañado de sus numeros telefónicos

```

15  -- Consulta para conocer el nombre de cada cliente acompañado de sus numeros t
16  • SELECT c.nombre, t.numero
17  FROM cliente AS c
18  LEFT JOIN telefono_cliente AS t ON c.cedula = t.cedula_cliente;
19

```

nombre	numero
Juan Pérez	+1234567890
Juan Pérez	+9876543210
Pedro López	+7777777777
Pedro López	+8888888888
Luis García	+3333333333
Laura González	+0000000000
Laura González	+9999999999
Ana Martínez	+5555555555
Ana Martínez	+6666666666
Carlos Ramírez	+4444444444
María Rodríguez	+1111111111
María Rodríguez	+2222222222

Consulta para ver el titulo del libro acompañado de su autor o autores

```

19  -----
20  -- Consulta para ver el titulo del libro acompañado de su autor o autores
21  • SELECT L.titulo ,A.nombre
22  FROM libro AS L
23  INNER JOIN libro_Autor AS LA ON (L.ISBN = LA.ISBN_libro)
24  INNER JOIN autor AS A ON (A.id = LA.id_autor);

```

titulo	nombre
Los detectives salvajes	Gabriel García Márquez
La Odisea	Gabriel García Márquez
2666	Paul Auster
Don Quijote de la Mancha	Paul Auster
Rayuela	Roberto Bolaño
El Aleph	Roberto Bolaño
Cien años de soledad	Juan José Saer
La ciudad y los perros	Juan José Saer
La casa de los espíritus	Michel Foucault
Moby Dick	Michel Foucault

Consulta para conocer el número de ventas de libros por editorial

```

26 -- Consulta para conocer el número de ventas de libros por editorial
27 • SELECT Ed.nombre, COUNT(Ed.nombre) AS contador_ventas
28 FROM editorial AS Ed
29 INNER JOIN libro AS L ON (Ed.nombre = L.nombre_editorial)
30 GROUP BY Ed.nombre
31

```

nombre	contador_ventas
Anagrama	6
Gallimard	6
Santillana	3
Viking	5

Vistas:

La primera vista se llama EditorialesConMenosVentas, puesto que es una información valiosa a la hora de revisar que editorial es la que esta presentando menores ventas, para así reducir el numero de libros comprados a esta misma editorial.

```

CREATE VIEW `EditorialesConMenosVentas` AS
SELECT Ed.nombre, COUNT(Ed.nombre) AS contador_ventas
FROM editorial AS Ed
INNER JOIN libro AS L ON (Ed.nombre = L.nombre_editorial)
GROUP BY Ed.nombre
ORDER BY contador_ventas ASC;

```

Result Grid		Filter Rows:
nombre	contador_ventas	
Santillana	3	
Viking	5	Resultado vista
Anagrama	6	
Gallimard	6	

En la segunda vista, se enlistan los clientes con mas compras de libros con la intención de hacer mejores ofertas para así aumentar su índice de satisfacción, además de planear planes de fidelización.

```

CREATE VIEW `ClientesConMasCompras` AS
SELECT c.*, COUNT(c.cedula) AS cantidad_compras
FROM libro_cliente AS lc
LEFT JOIN cliente AS c ON lc.id_cliente = c.cedula
GROUP BY c.cedula
ORDER BY cantidad_compras DESC;

```

Result Grid			
	cedula	nombre	cantidad_compras
▶	123456789	Juan Pérez	2
	456123789	Luis García	2
	987654321	María Rodríguez	2
	321654987	Pedro López	1
	456789123	Laura González	1
	789123456	Ana Martínez	1
	987321654	Carlos Ramírez	1

Actividad 2:

Se define la base de datos basada en el MR de hospital:

A partir del modelo MR presentado en hospital se crea la database “hospital”, con las tablas definidas ya en el MR presentado:

- En primer lugar, se crean las tablas que no tienen llaves foráneas, las tablas:
 - Medicos
 - Procedimientos
 - Medicamentos
- El segundo paso es definir las tablas que tienen llave foránea, las tablas:
 - Pacientes
 - Enfermeros
 - Facturas
 - Teléfono_paciente
 - Teléfono_medicos
 - Telefono_enfermeros
 - Paciente_medicamento
 - Medico_procedimiento

En el archivo con el nombre **DB_hospital.sql** se encuentra el script para la generación de las tablas con los atributos respectivos del MR presentados.

5 registros por tabla:

Los registros se hacen para cada tabla, utilizando la herramienta chatGPT como generador de texto, para facilitar la creación de los registros, en el archivo **registros_db.sql**, está el script utilizado para generar los 5 registros para cada tabla, el id en todas las tablas es un entero de 3 dígitos, donde el primer dígito es el orden en el que se fueron creando los registros. A continuación, un ejemplo para la tabla

pacientes:

```
INSERT INTO Pacientes VALUES
(401, 'María', 'González', 'Calle 123', 301),
(402, 'Juan', 'Martínez', 'Avenida 456', 302),
(403, 'Ana', 'López', 'Calle Principal', 303),
(404, 'Carlos', 'Rodríguez', 'Avenida Central', 304),
(405, 'Laura', 'Fernández', 'Calle 789', 305);
```

Consultas:

Para el caso de la primera consulta se definen las consultas según los requerimientos en el archivo **Consultas_db**, ahí se pueden encontrar tanto la consulta de medicamentos por paciente, y enfermeros por procedimiento.

Vistas:

Por último, se definen las 3 vistas solicitadas en el archivo con el nombre **vistas_db**, con los siguientes nombres y justificaciones para cada una:

- **Resumen_facturación**: esta vista se crea para el caso en el que el familiar del paciente desea conocer el valor total hasta determinada fecha en el que va el proceso en el hospital, para así darse una idea del valor final.
- **Procedimientos_pacientes**: Esta vista es para que, por ejemplo, una persona encargada de revisar los procedimientos llevado a cabo en el hospital conozca a que pacientes se realizaron, y quienes fueron los responsables de dicho procedimiento.
- **Factura_completa**: esta vista es para que al final del proceso medico realizado al paciente, se le haga el cobro respectivo, enlistando cuales fueron los procedimientos y medicamentos por los cuales se le está realizando el cobro.

Actividad 3:

Procedimientos para librería:

A continuación, se crean 4 procedimientos diferentes para la tabla “editorial”, que se encuentra en la database creada anteriormente de una librería. En agregar y actualizar se solicitarán 4 atributos: Nombre, Ciudad, complemento, teléfono; Siendo estos los atributos necesarios para crear una nueva editorial, o en determinado caso actualizar alguno de ellos. Los procedimientos se encuentran en el archivo “procedimientos_libreria”


- **Agregar:**

Para este caso se solicitan los atributos necesarios, y se utiliza el comando “Insert info” para insertar la información de una nueva editorial en la tabla.

```
CREATE PROCEDURE agregar_editorial(  
    IN nombre_editorial VARCHAR(50),  
    IN ciudad VARCHAR(30),  
    IN complemento VARCHAR(100),  
    IN Telefono VARCHAR(20)  
)  
BEGIN  
    INSERT INTO Editorial (nombre, ciudad, complemento, Telefono)  
    VALUES (nombre_editorial, ciudad, complemento, Telefono);  
END//
```

- **Actualizar:**

Como se mencionó, se solicitan los atributos necesarios, y por medio del comando “Update”, se actualiza la información perteneciente a la editorial, el nombre no porque esta es la clave de la tabla.

```
CREATE PROCEDURE actualizar_editorial(  
    IN nombre_editorial VARCHAR(50),  
    IN nueva_ciudad VARCHAR(30),  
    IN nuevo_complemento VARCHAR(100),  
    IN nuevo_Telefono VARCHAR(20)  
)  
BEGIN  
    UPDATE Editorial  
    SET ciudad = nueva_ciudad,  
        complemento = nuevo_complemento,  
        Telefono = nuevo_Telefono  
    WHERE nombre = nombre_editorial;   
END//
```

- **Consultar:**

Para este procedimiento simplemente se solicita el nombre de la editorial como clave de la tabla, y por medio del comando “Select *”, se selecciona toda la información perteneciente a la editorial


```

CREATE PROCEDURE consultar_editorial(
    IN nombre_editorial VARCHAR(50)
)
BEGIN
    SELECT * FROM Editorial WHERE nombre = nombre_editorial;
END//

```

- **Borrar:**

Para este procedimiento simplemente se solicita el nombre de la editorial como clave de la tabla, y por medio del comando “Delete”, se elimina toda la información perteneciente a la editorial de la tabla.

```

CREATE PROCEDURE borrar_editorial(
    IN nombre_editorial VARCHAR(50)
)
BEGIN
    DELETE FROM Editorial WHERE nombre = nombre_editorial;
END//

```

Triggers librería (Archivo “Trigger_librería”):

En primer lugar, se crea la tabla “control_de_cambios_librería”, para ahí recopilar la información perteneciente a los cambios, se le incluyen los 3 atributos solicitados: usuario, acción y fecha (La fecha tiene un comando para guardar la información del momento exacto del cambio):

```

CREATE TABLE IF NOT EXISTS control_de_cambios_librería (
    usuario VARCHAR(50) NOT NULL,
    accion ENUM('Agregar', 'Eliminar') NOT NULL,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

A continuación, se deben crear dos triggers, por medio del comando “Create trigger”, este trigger se asocia a la tabla editorial y al evento “Insert”, por medio de el comando “For each row”, se logra que tenga en cuenta en cada línea de la tabla editorial, después inserta a la tabla contro de cambios el usuario, la acción (La fecha se agrega al crear la línea):

```

CREATE TRIGGER agregar_editorial_trigger AFTER INSERT ON Editorial
FOR EACH ROW
BEGIN
    INSERT INTO control_de_cambios_librería (usuario, accion)
    VALUES (USER(), 'Agregar');
END//
-

```

Por último, se crea un trigger que se asocia a la tabla "Editorial" y al evento "DELETE". Al utilizar el comando "FOR EACH ROW", el trigger se ejecutará para cada fila eliminada de la tabla "Editorial". Dentro del trigger, se registra el usuario que realizó la acción y la acción misma (en este caso, "eliminar").

```

CREATE TRIGGER eliminar_editorial_trigger AFTER DELETE ON Editorial
FOR EACH ROW
BEGIN
    INSERT INTO control_de_cambios_librería (usuario, accion)
    VALUES (USER(), 'Eliminar');
END//
-

```

Procedimientos para hospital:

Se crean 4 procedimientos diferentes para la tabla "Medicos". En agregar y actualizar se solicitarán los atributos de la tabla: Nombre, Apellido y especialidad; Siendo estos los atributos necesarios para crear una nueva línea, o en determinado caso actualizar alguno de ellos. Los procedimientos se encuentran en el archivo "procedimientos_hospital":

- **Agregar:**

```

CREATE PROCEDURE agregar_medico(
    IN nombre_medico VARCHAR(45),
    IN apellido_medico VARCHAR(45),
    IN especialidad VARCHAR(45)
)
BEGIN
    INSERT INTO Medicos (nombre_medico, apellido_medico, especialidad)
    VALUES (nombre_medico, apellido_medico, especialidad);
END //
-

```

- **Actualizar:**

```

CREATE PROCEDURE actualizar_medico(
    IN id_medico INT,
    IN nuevo_nombre_medico VARCHAR(45),
    IN nuevo_apellido_medico VARCHAR(45),
    IN nueva_especialidad VARCHAR(45)
)
BEGIN
    UPDATE Medicos
    SET nombre_medico = nuevo_nombre_medico,
        apellido_medico = nuevo_apellido_medico,
        especialidad = nueva_especialidad
    WHERE id_medicos = id_medico;
END //
-

```

- **Consultar:**

```

CREATE PROCEDURE consultar_medico(
    IN id_medico INT
)
BEGIN
    SELECT * FROM Medicos WHERE id_medicos = id_medico;
END //
-

```

- **Eliminar:**

```

CREATE PROCEDURE eliminar_medico(
    IN id_medico INT
)
BEGIN
    DELETE FROM Medicos WHERE id_medicos = id_medico;
END //
-

```

Triggers hospital (Archivo “Trigger_hospital”):

Se crea la tabla para el control de cambios:

```
CREATE TABLE IF NOT EXISTS control_de_cambios_hospital (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    usuario VARCHAR(50),  
    accion VARCHAR(20),  
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
) ENGINE=InnoDB;
```

Se crean los dos triggers:

```
CREATE TRIGGER agregar_medico_trigger  
AFTER INSERT ON Medicos  
FOR EACH ROW  
BEGIN  
    INSERT INTO control_de_cambios_hospital (usuario, accion)  
    VALUES (USER(), 'agregar');  
END //
```

```
CREATE TRIGGER eliminar_medico_trigger  
AFTER DELETE ON Medicos  
FOR EACH ROW  
BEGIN  
    INSERT INTO control_de_cambios_hospital (usuario, accion)  
    VALUES (USER(), 'eliminar');  
END //
```