

BASES DE DATOS SQL

DISEÑO, CREACIÓN Y MANIPULACIÓN DE UNA BASE DE DATOS SQL

23 de mayo de 2025

Contenido

[DISEÑO DE UN MODELO ENTIDAD-RELACIÓN 2](#_Toc198918563)

[Explicación del Diseño Entidad-Relación 2](#_Toc198918564)

[Diseño del Modelo Relacional para el Sistema de Reservas de Vehículos 4](#_Toc198918565)

[Modelo Relacional Completo 6](#_Toc198918566)

[Consultas SQL 9](#_Toc198918567)

[Grafo relacional expresado en tabla 9](#_Toc198918568)

[CONSULTAS DE INSERCIÓN 15](#_Toc198918569)

[CONSULTAS DE SELECCIÓN BÁSICAS 16](#_Toc198918570)

[CONSULTAS DE AGRUPACIÓN 20](#_Toc198918571)

[SUBCONSULTAS 25](#_Toc198918572)

[CONSULTAS DE ACCIÓN 31](#_Toc198918573)

# DISEÑO DE UN MODELO ENTIDAD-RELACIÓN

## Explicación del Diseño Entidad-Relación

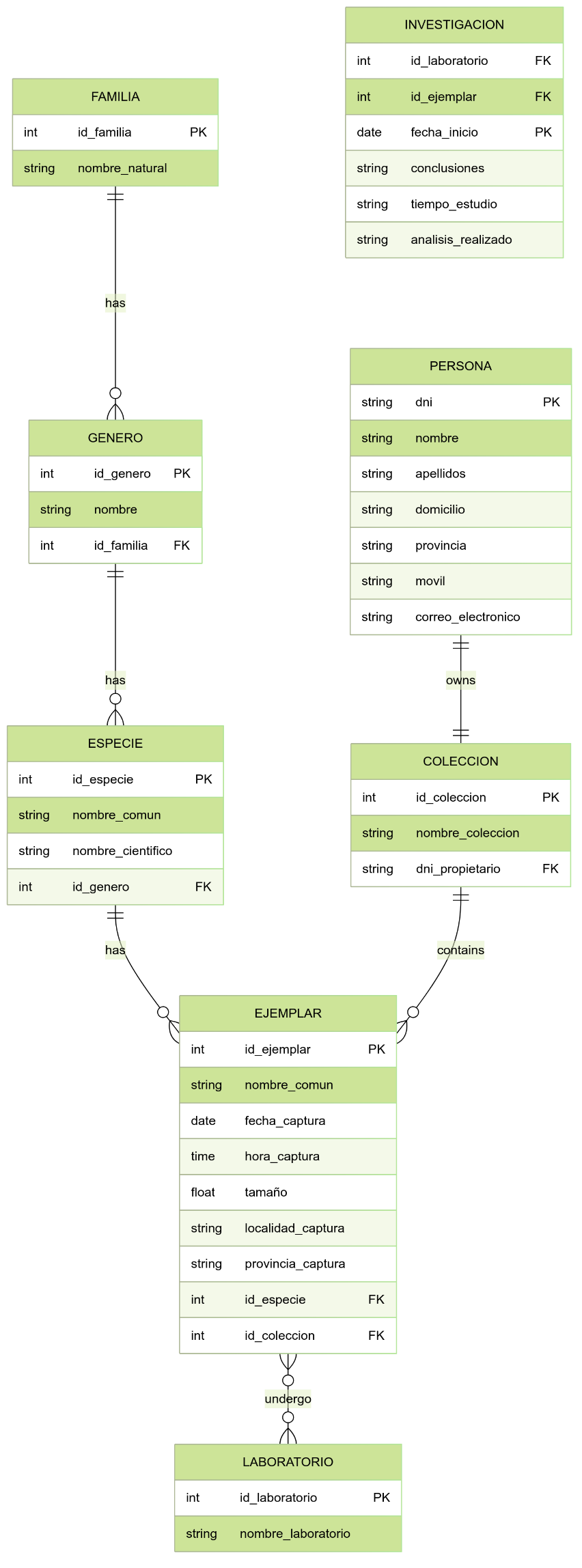
El diseño refleja un sistema de base de datos para el estudio de mariposas.

Las **entidades principales** son:

* **Familia**, **Género** y **Especie**, organizadas jerárquicamente.
* **Ejemplar**, que representa cada mariposa capturada, con datos como tamaño, localidad y fecha de captura.
* **Persona**, propietaria de una única **Colección** que agrupa varios ejemplares.
* **Laboratorio**, donde se estudian los ejemplares mediante la entidad intermedia **Investigación**.

Se han definido claves primarias y foráneas para mantener la integridad referencial, y se han aplicado las cardinalidades adecuadas (1:1, 1:N y N:M) según lo indicado en el enunciado.

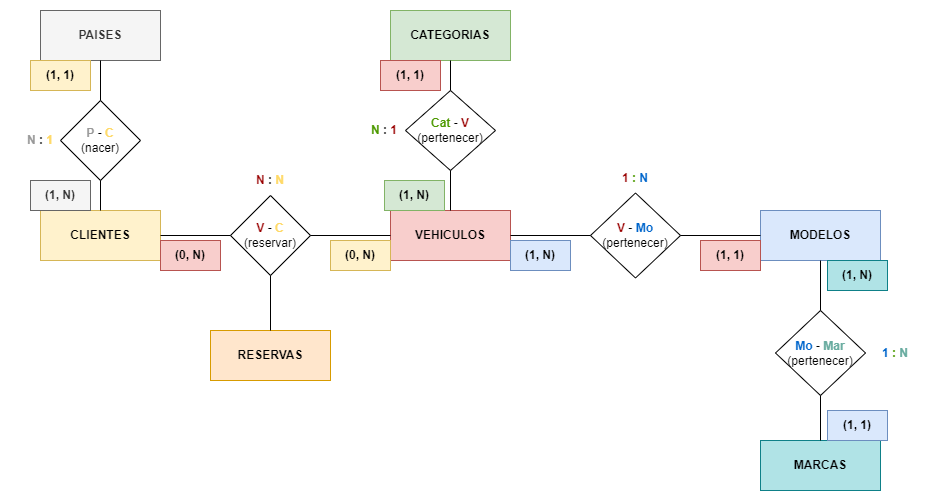
Este modelo permite estructurar toda la información necesaria para gestionar de forma eficaz los datos del estudio biológico.

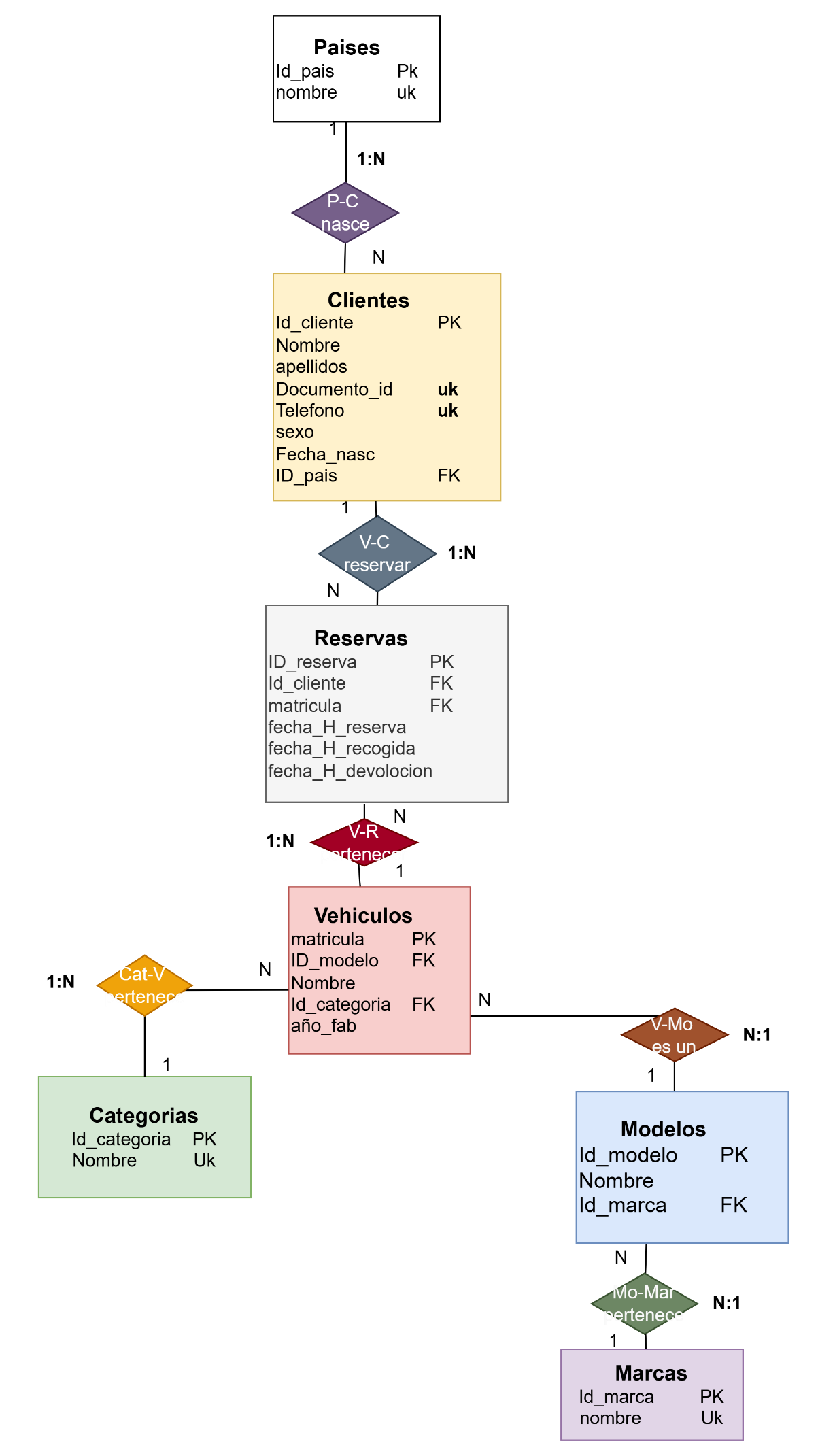


# Diseño del Modelo Relacional para el Sistema de Reservas de Vehículos

Este ejercicio consiste en diseñar el modelo relacional a partir de un modelo entidad-relación clásico. El sistema permite gestionar reservas de vehículos por parte de clientes, incluyendo la información sobre países, categorías de vehículos, marcas, modelos y datos relevantes de las reservas.

🔹Diagrama Entidad-Relación





### Modelo Relacional Completo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del atributo** | **Tipo de dato** | **Clave** | **Restricción** | **Descripción** |
| id\_pais | INT | PK |  | Identificador único del país |
| nombre | VARCHAR(100) |  | UK | Nombre del país |

**Tabla Países**

**Tabla Clientes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del atributo** | **Tipo de dato** | **Clave** | **Restricción** | **Descripción** |
| id\_cliente | INT | PK |  | Identificador del cliente |
| nombre | VARCHAR(50) |  |  | Nombre del cliente |
| apellidos | VARCHAR(100) |  |  | Apellidos del cliente |
| documento\_id | VARCHAR(20) |  | UK | DNI o documento identificativo |
| telefono | VARCHAR(15) |  |  | Número de teléfono |
| sexo | CHAR(1) |  |  | Sexo del cliente (M/F/...) |
| fecha\_nacimiento | DATE |  |  | Fecha completa de nacimiento |
| id\_pais | INT | FK |  | País de nacimiento / PAISES(id\_pais) |

**Tabla Categorías**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del atributo** | **Tipo de dato** | **Clave** | **Restricción** | **Descripción** |
| id\_categoria | INT | PK |  | Identificador de la categoría |
| nombre | VARCHAR(50) |  | UK | Nombre de la categoría (Ej: Turismo) |

**Tabla Marcas**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del atributo** | **Tipo de dato** | **Clave** | **Restricción** | **Descripción** |
| id\_marca | INT | PK |  | Identificador de la marca |
| nombre | VARCHAR(50) |  | UK | Nombre de la marca (Ford...) |

**Tabla Modelos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del atributo** | **Tipo de dato** | **Clave** | **Descripción** |
| id\_modelo | INT | PK | Identificador del modelo |
| nombre | VARCHAR(50) |  | Nombre del modelo (Ej: Focus) |
| id\_marca | INT | FK | Marca del modelo → MARCAS(id\_marca) |

**Tabla Vehículos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del atributo** | **Tipo de dato** | **Clave** | **Descripción** |
| id\_modelo | INT | PK | Identificador del modelo |
| nombre | VARCHAR(50) |  | Nombre del modelo (Ej: Focus) |
| id\_marca | INT | FK | Marca del modelo → MARCAS(id\_marca) |
| año\_fab | INT |  | Año de fabricación del vehículo |
| Id\_categoria | INT | FK | Identificador de la categoría del vehículo (Referencia a CATEGORIAS) |

**Tabla Reservas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del atributo** | **Tipo de dato** | **Clave** | **Descripción** |
| id\_reserva | INT | PK | Identificador de la reserva |
| id\_cliente | INT | FK | Cliente que realiza la reserva → CLIENTES(id\_cliente) |
| matricula | VARCHAR(10) | FK | Vehículo reservado → VEHICULOS(matricula) |
| fecha\_h\_reserva | DATETIME |  | Cuándo se hizo la reserva |
| fecha\_h\_recogida | DATETIME |  | Cuándo se recoge el vehículo |
| fecha\_h\_devolucion | DATETIME |  | Cuándo se devuelve el vehículo |

Las claves foráneas aseguran la integridad referencial entre las tablas.

Las relaciones 1:N están representadas con claves foráneas desde el lado N hacia el lado 1.

Se han añadido todos los atributos que se indican expresamente en el enunciado, aunque no estén en el diagrama E-R.

**PK** (Primary Key - Clave Primaria)

**FK** (Foreign Key - Clave Foránea)

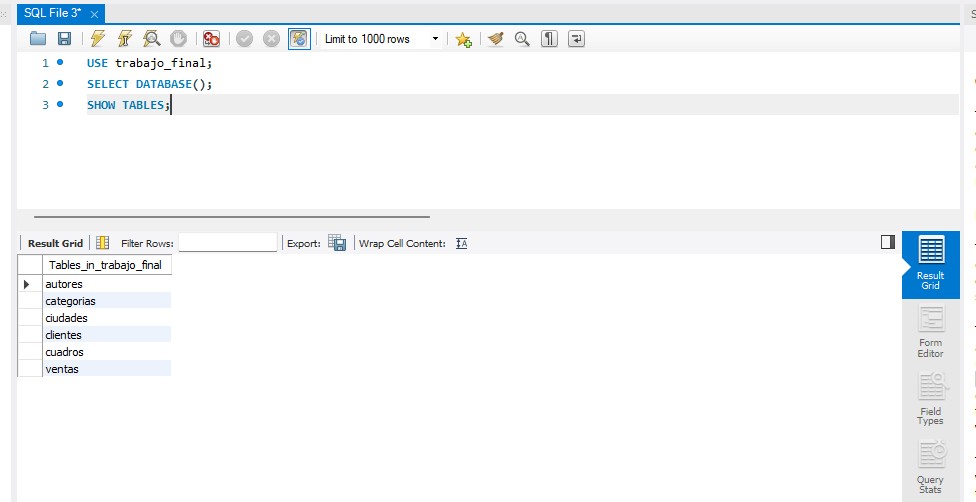
**UK** (UNIQUE)

# Consultas SQL

## Grafo relacional expresado en tabla

|  |
| --- |
| **TABLAS Y ATRIBUTOS** |
| **1. AUTORES** |
| * id\_autor INT, PK, NOT NULL, AUTO\_INCREMENT |
| * nombre VARCHAR(45), NOT NULL |
| * apellidos VARCHAR(60), NOT NULL |
| **2. CATEGORIAS** |
| * id\_categoria INT, PK, NOT NULL, AUTO\_INCREMENT |
| * nombre VARCHAR(60), NOT NULL, UNIQUE |
| * observaciones TEXT |
| **3. CIUDADES** |
| * id\_ciudad INT, PK, NOT NULL, AUTO\_INCREMENT |
| * nombre VARCHAR(60), NOT NULL, UNIQUE |
| **4. CLIENTES** |
| * id\_cliente INT, PK, NOT NULL, AUTO\_INCREMENT |
| * nombre VARCHAR(45), NOT NULL |
| * apellido1 VARCHAR(60), NOT NULL |
| * apellido2 VARCHAR(60), NOT NULL |
| * fecha\_nacimiento DATE, NOT NULL |
| * email VARCHAR(140), NOT NULL, UNIQUE |
| * sexo ENUM(‘Hombre’, ‘Mujer’, ‘No indicado’) |
| * ciudad\_nacimiento INT, FK, NOT NULL (→ CIUDADES) |
| * observaciones TEXT |
| **5. CUADROS** |
| * id\_cuadro INT, PK, NOT NULL, AUTO\_INCREMENT |
| * titulo VARCHAR(100), NOT NULL, UNIQUE |
| * anyo\_realizacion YEAR, NOT NULL |
| * categoria INT, FK, NOT NULL (→ CATEGORIAS) |
| * autor INT, FK, NOT NULL (→ AUTORES) |
| **6. VENTAS** |
| * id\_venta INT, PK, NOT NULL, AUTO\_INCREMENT |
| * cliente INT, FK, NOT NULL (→ CLIENTES) |
| * cuadro INT, FK, NOT NULL (→ CUADROS) |
| * cantidad\_ejemplares INT, NOT NULL |
| * precio\_ejemplar FLOAT, NOT NULL |
| * fecha\_venta DATE, NOT NULL |

CONSULTAS DE CREACIÓN, ALTERACIÓN Y BORRADO

* Crear la base de datos, usarla ver las tablas

USE trabajo\_final;

SELECT DATABASE();

SHOW TABLES;

* Crear las entidades propuestas en el grafo relacional

CREATE TABLE ventas (

id\_ventas INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

cliente INT NOT NULL,

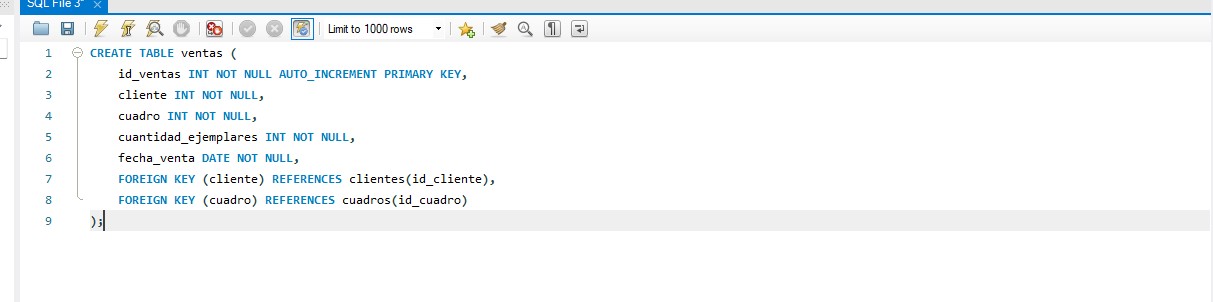
cuadro INT NOT NULL,

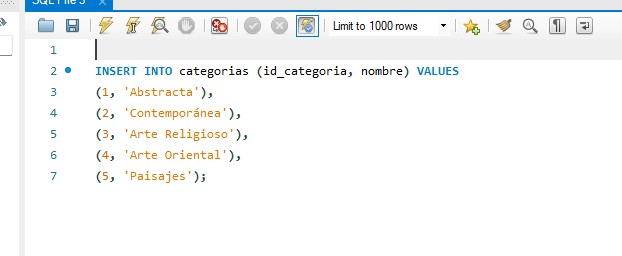
cuantidad\_ejemplares INT NOT NULL,

fecha\_venta DATE NOT NULL,

FOREIGN KEY (cliente) REFERENCES clientes(id\_cliente),

FOREIGN KEY (cuadro) REFERENCES cuadros(id\_cuadro)

);



* INSERT INTO categorías (id\_categoria, nombre) VALUES

(1, 'Abstracta'),

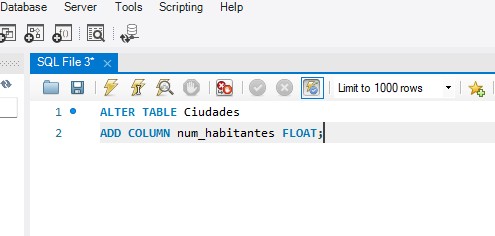
(2, 'Contemporánea'),

(3, 'Arte Religioso'),

(4, 'Arte Oriental'),

(5, 'Paisajes');

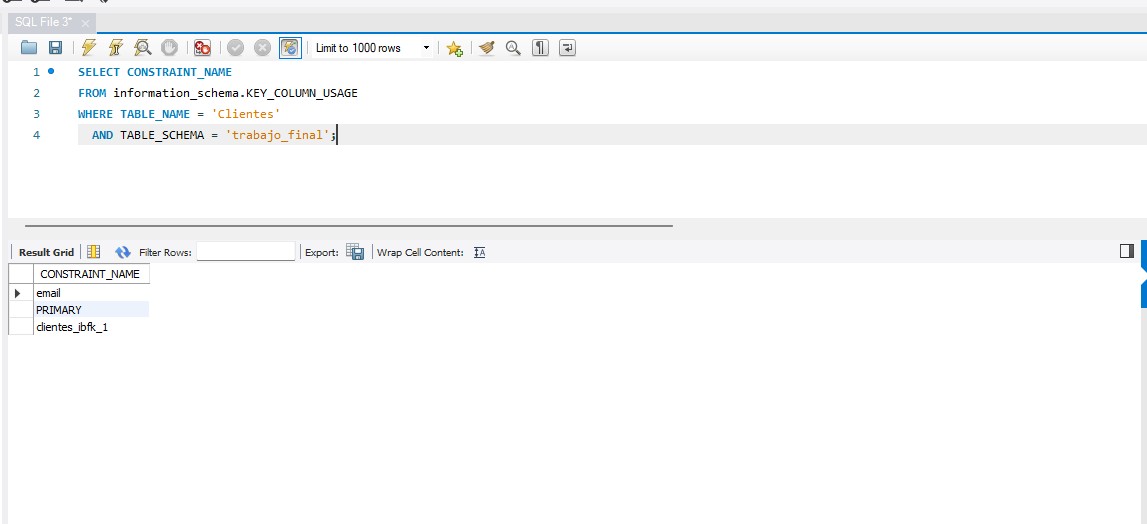
* Añadir campo num\_habitantes a Ciudades



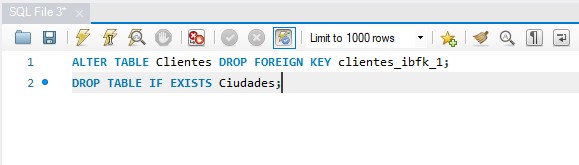
ALTER TABLE ciudades

ADD COLUMN num\_habitantes FLOAT;

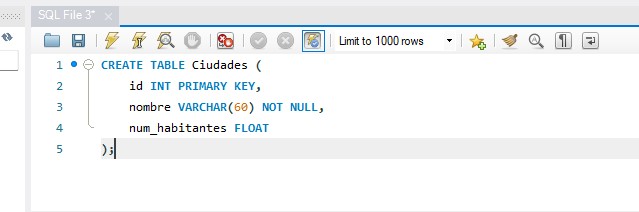
* Pasos para borrar la entidad Ciudades correctamente



* Primero necesitamos **quitar la relación de clave foránea** que une Clientes.ciudad\_nacimiento con Ciudades.id.



* Primero se elimina la restricción de clave foránea desde Clientes, ya que Ciudades está referenciada por esa tabla. Luego se borra la tabla Ciudades con DROP TABLE.
* Crear nuevamente la entidad Ciudades con los datos del grafo relacional



CREATE TABLE Ciudades (

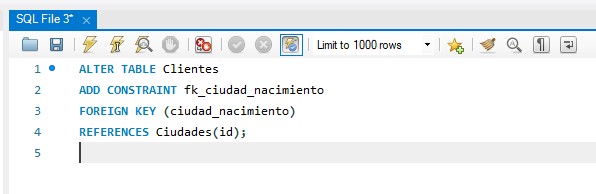
id INT PRIMARY KEY,

nombre VARCHAR(60) NOT NULL,

num\_habitantes FLOAT

);

* Clave foránea en el atributo ciudad\_nacimiento de la entidad Clientes de forma que forme la realización con la entidad Ciudades



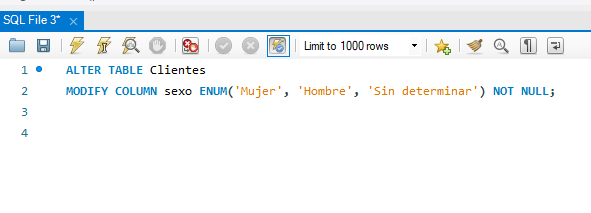
ALTER TABLE Clientes

ADD CONSTRAINT fk\_ciudad\_nacimiento

FOREIGN KEY (ciudad\_nacimiento)

REFERENCES Ciudades(id);

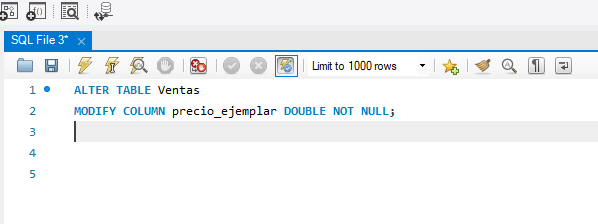
* Reemplazar "No indicado" por "Sin determinar" en el ENUM



ALTER TABLE Clientes

MODIFY COLUMN sexo ENUM('Mujer', 'Hombre', 'Sin determinar') NOT NULL;

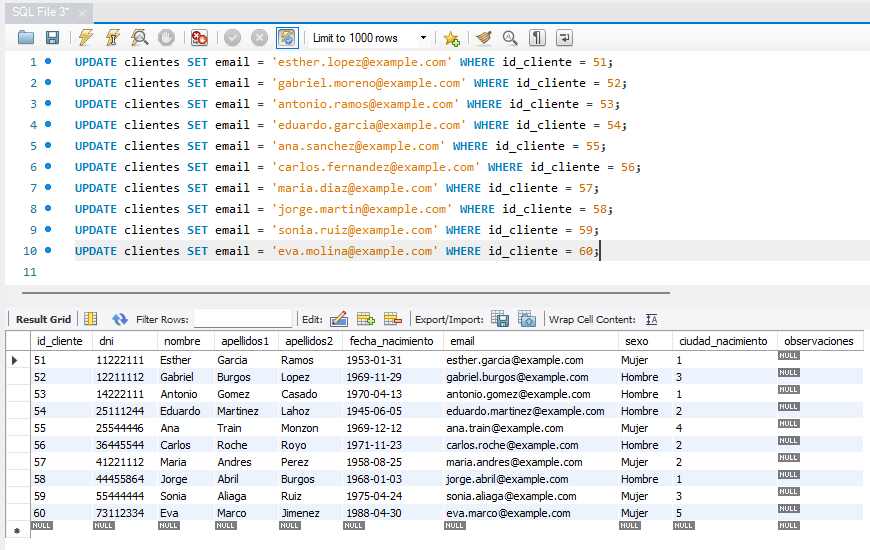
* Permite cambiar el tipo de dato asociado al atributo “precio\_ejemplar” de la entidad Ventas de Float a Double, mantenimiento la restricción NOT NULL



ALTER TABLE Ventas

MODIFY COLUMN precio\_ejemplar DOUBLE NOT NULL;

# CONSULTAS DE INSERCIÓN



UPDATE clientes SET email = 'esther.lopez@example.com' WHERE id\_cliente = 51;

UPDATE clientes SET email = 'gabriel.nomen@example.com' WHERE id\_cliente = 52;

UPDATE clientes SET email = 'antonio.ramos@example.com' WHERE id\_cliente = 53;

UPDATE clientes SET email = 'eduardo.garcia@example.com' WHERE id\_cliente = 54;

UPDATE clientes SET email = 'ana.sanchez@example.com' WHERE id\_cliente = 55;

UPDATE clientes SET email = 'carlos.fernandez@example.com' WHERE id\_cliente = 56;

UPDATE clientes SET email = 'maria.diaz@example.com' WHERE id\_cliente = 57;

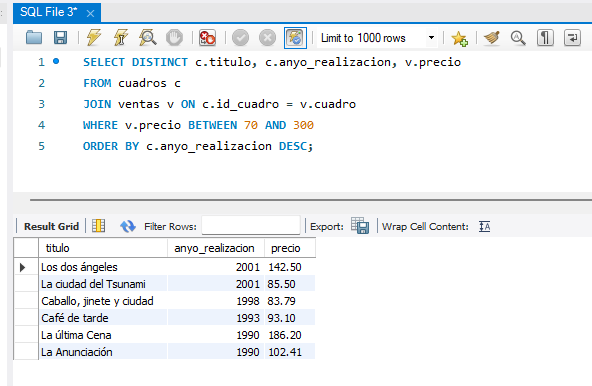
UPDATE clientes SET email = 'jorge.martin@example.com' WHERE id\_cliente = 58;

UPDATE clientes SET email = 'sonia.ruiz@example.com' WHERE id\_cliente = 59;

UPDATE clientes SET email = 'eva.nolina@example.com' WHERE id\_cliente = 60;

# CONSULTAS DE SELECCIÓN BÁSICAS

* Consulta de selección que muestre el título, año de realización y precio de aquellos cuadros que hayan sido vendidos por un precio que esté comprendido entre los 70 y 300 €. Será necesario ordenar los resultados por año de realización descendente.



-- 1. Cuadros vendidos entre 70€ y 300€, ordenados por año desc

SELECT titulo, anyo\_realizacion, precio

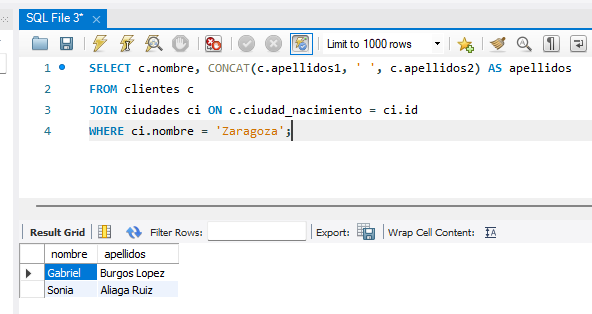
FROM cuadros c

JOIN ventas v ON c.id\_cuadro = v.cuadro

WHERE precio BETWEEN 70 AND 300

ORDER BY anyo\_realizacion DESC;

* Consulta de selección que muestre el nombre y apellidos de todos los clientes que residan en Zaragoza



-- 2. Clientes que residan en Zaragoza

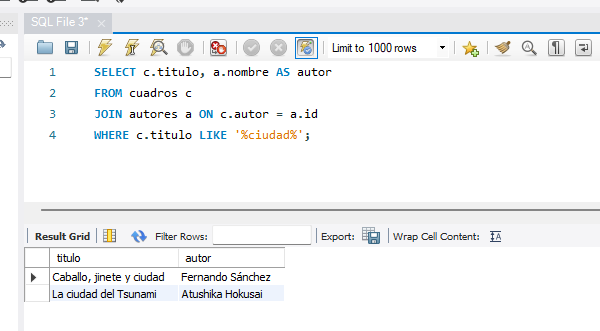
SELECT cl.nombre, CONCAT(cl.apellidos1, '', cl.apellidos2) AS apellidos

FROM clientes cl

JOIN ciudades ci ON cl.ciudad\_nacimiento = ci.id

WHERE ci.nombre = 'Zaragoza';

* Consulta de selección que muestre el título y autor de los cuadros que contengan en su título la palabra “ciudad”.



-- 3. Cuadros cuyo título contiene "ciudad"

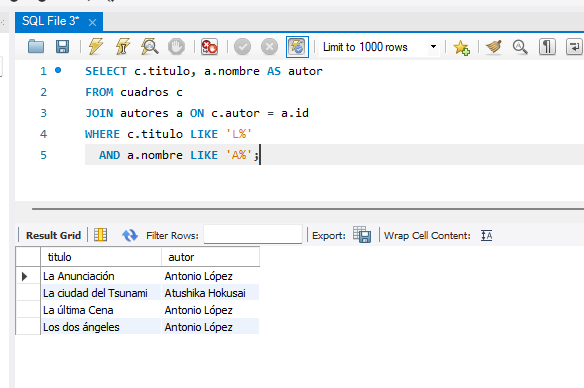
SELECT c.titulo, a.nombre AS autor

FROM cuadros c

join autores a ON c.autor = a.id

WHERE titulo LIKE '%ciudad%';

* Consulta de selección que muestre el título y autor de aquellos cuadros cuya primera letra del título sea “L” y que pertenezcan a un autor cuyo nombre comience por “A”.



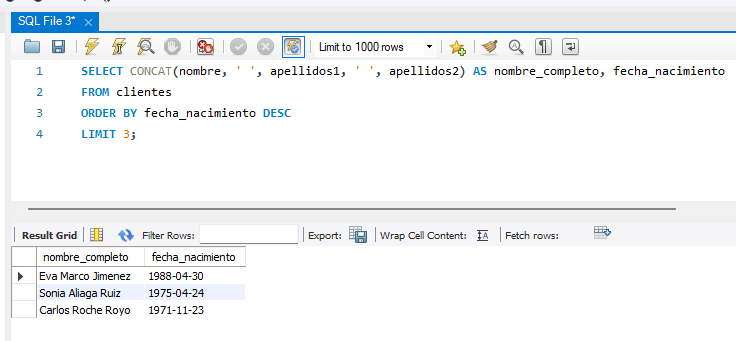
-- 4. Cuadros cuyo título empieza por "L" y autor empieza por "A"  
SELECT c.titulo, a.nombre AS autor

FROM cuadros c

JOIN autores a ON c.autor = a.id

WHERE c.titulo LIKE 'L%' AND a.nombre LIKE 'A%';

* Consulta de selección que muestre el nombre completo (nombre y apellido en una misma columna) de los tres clientes más jóvenes.



-- 5. Tres clientes más jóvenes

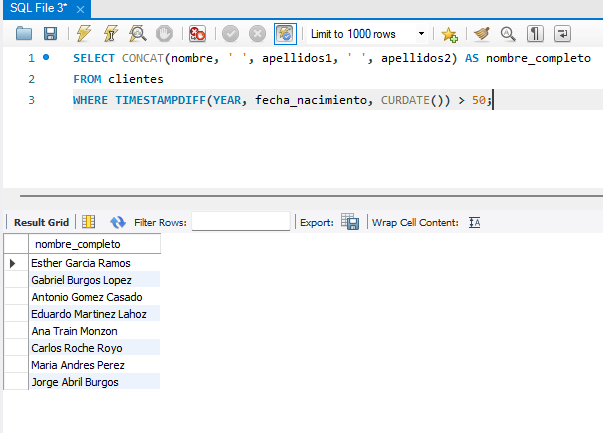
SELECT CONCAT(nombre, ' ', apellidos1, ' ', apellidos2) AS nombre\_completo, fecha\_nacimiento

FROM clientes

ORDER BY fecha\_nacimiento DESC

LIMIT 3;

* Consulta de selección que muestre el nombre completo de los clientes (nombre y apellido en una misma columna) de todos los clientes que a día de hoy tengan más de 50 años.



-- 6. Clientes mayores de 50 años

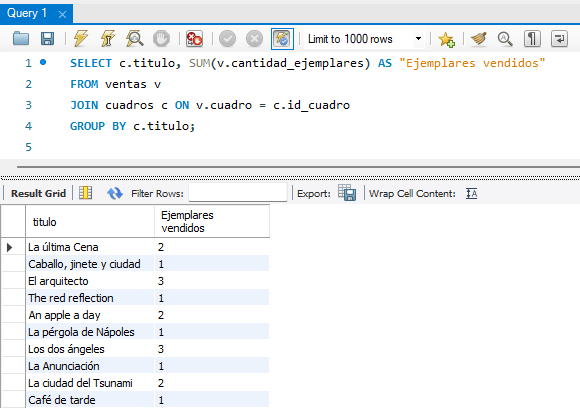
SELECT CONCAT(nombre, ' ', apellidos1, ' ', apellidos2) AS nombre\_completo

FROM clientes

WHERE TIMESTAMPDIFF(YEAR, fecha\_nacimiento, CURDATE()) > 50;

# CONSULTAS DE AGRUPACIÓN

* Consulta de selección que muestre el título de los cuadros y la cantidad que se han vendido de cada uno. Este último dato se debe mostrar bajo el alias “Ejemplares vendidos”



-- 1. Título de los cuadros y cantidad vendida

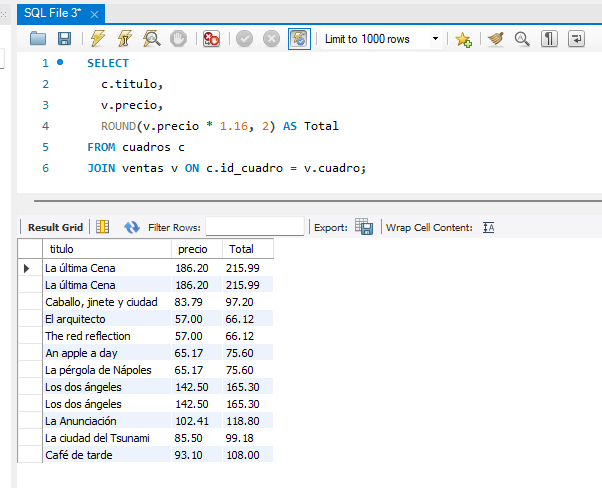
SELECT c.titulo, SUM(v.cantidad\_ejemplares) AS "Ejemplares vendidos"

FROM ventas v

JOIN cuadros c ON v.cuadro = c.id\_cuadro

GROUP BY c.titulo;

* Consulta de selección que muestre el título de los cuadros, su precio de venta y en una nueva columna llamada “Total” el coste final asociado a cada cuadro si se les aplicase un IVA del 16%.



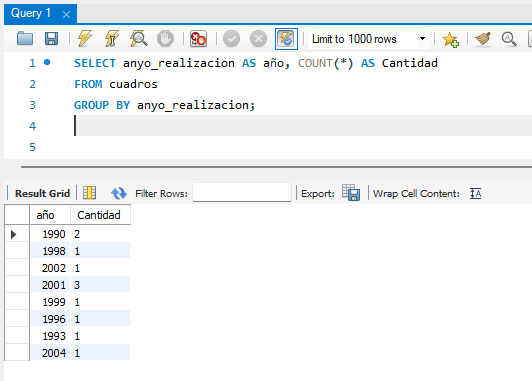
-- 2. Coste final con IVA

SELECT c.titulo, v.precio, ROUND(v.precio \* 1.16, 2)AS Total

FROM ventas v

JOIN cuadros c ON v.cuadro = c.id\_cuadro;

* Consulta de selección que muestre los años en los que se ha pintado cada uno de los cuadros y la cantidad de cuadros diferentes pintados durante esos períodos con el alias “Cantidad”



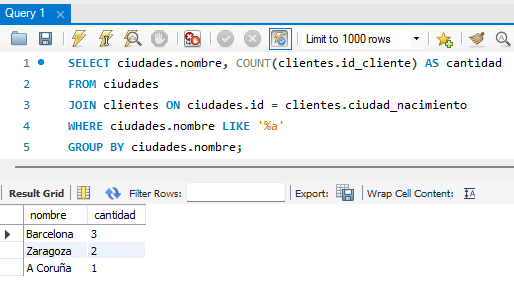
-- 3. Años y cantidad de cuadros

SELECT anyo\_realizacion AS año, COUNT(\*) AS Cantidad

FROM cuadros

GROUP BY anyo\_realizacion;

* Consulta de selección que muestre la cantidad de clientes que tenemos registrados para cada ciudad, pero sólo de aquellas ciudades cuyo nombre finalicen con la letra “a”.



-- 4. Clientes por ciudad que terminan en "a"

SELECT ciudades.nombre, COUNT(clientes.id\_cliente) AS cantidad

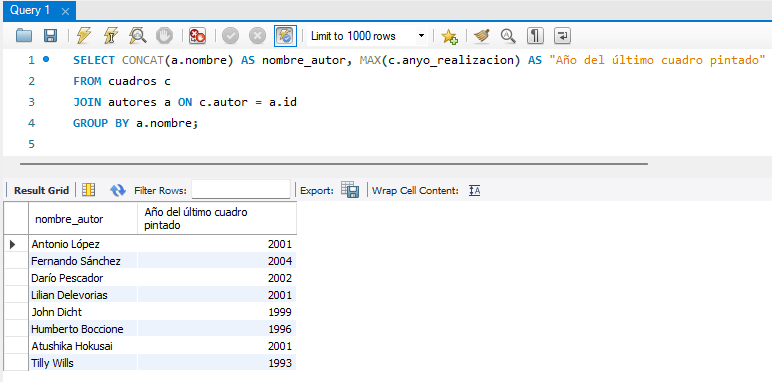
FROM ciudades

JOIN clientes ON ciudades.id = clientes.ciudad\_nacimiento

WHERE ciudades.nombre LIKE '%a'

GROUP BY ciudades.nombre;

* Consulta de selección que muestre el nombre completo de los autores (nombre y apellido en una misma columna) y el año del cuadro más reciente que han pintado con el alias “Año del último cuadro pintado”.



-- 5. Año del cuadro más reciente por autor

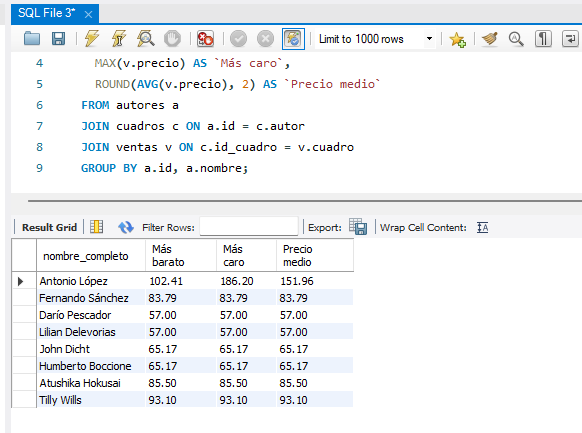
SELECT CONCAT(a.nombre) AS nombre\_autor, MAX(c.anyo\_realizacion) AS "Año del último cuadro pintado"

FROM cuadros c

JOIN autores a ON c.autor = a.id

GROUP BY a.nombre;

* Consulta de selección que muestre el precio del cuadro más barato con el alias “Más barato”, el precio del cuadro más caro con el alias “Más caro” y el precio medio de los cuadros con el alias “Precio medio” de cada uno de los autores, mostrando también el nombre completo de los mismos.



-- 6. Precio mínimo, máximo y medio por autor

SELECT a.nombre,

MIN(v.precio) AS "Más barato",

MAX(v.precio) AS "Más caro",

round(avg(v.precio), 2) AS "Precio medio"

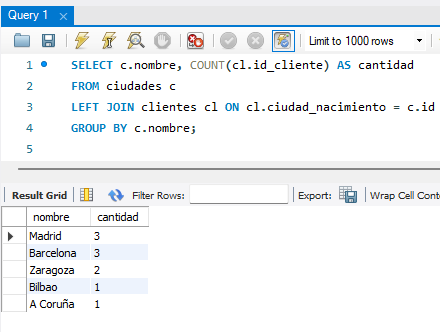
FROM ventas v

JOIN cuadros c ON v.cuadro = c.id\_cuadro

JOIN autores a ON c.autor = a.id

GROUP BY a.nombre;

* Consulta de selección que muestre la cantidad de clientes que tenemos de cada ciudad, incluyendo las ciudades dónde aún no haya clientes.



-- 7. Clientes por ciudad incluyendo ciudades sin clientes

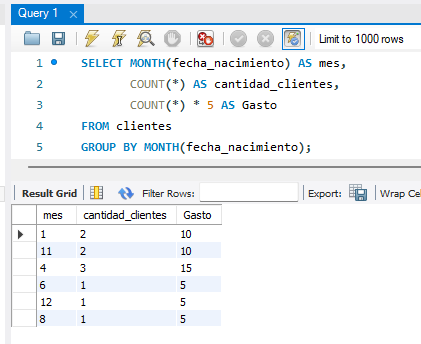
SELECT c.nombre, COUNT(cl.id\_cliente) AS cantidad

FROM ciudades c

LEFT JOIN clientes cl ON cl.ciudad\_nacimiento = c.id

GROUP BY c.nombre;

* La empresa desea enviar un correo electrónico a cada cliente por su cumpleaños con un código de descuento y desea que realicemos una consulta de selección que permita mostrar los diferentes meses del año, la cantidad de clientes que tenemos que cumplan años en cada uno de los meses y en una columna llamada “Gasto” la cantidad de dinero que perderá la empresa con esta medida, suponiendo que cada descuento supone una pérdida de 5€.



-- 8. Cumpleaños por mes y gasto

SELECT MONTH(fecha\_nacimiento) AS mes,

COUNT(\*) AS cantidad\_clientes,

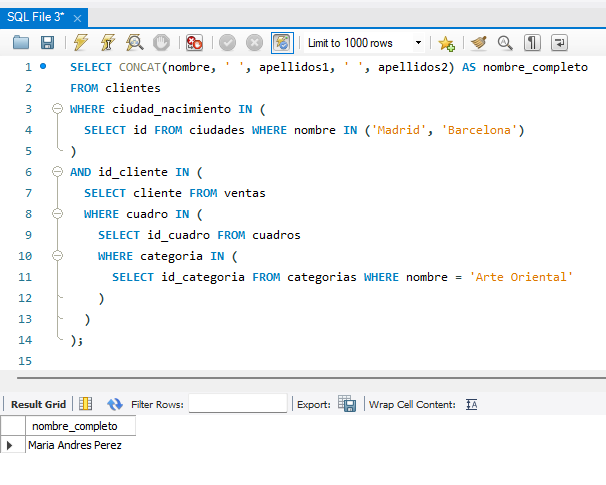
COUNT(\*) \* 5 AS Gasto

FROM clientes

GROUP BY MONTH(fecha\_nacimiento);

# SUBCONSULTAS

Consulta de selección que muestre el nombre completo de los clientes que residan en Madrid o Barcelona y hayan comprado cuadros de la categoría “Arte Oriental”.



-- 1. Clientes de Madrid o Barcelona que compraron cuadros de "Arte Oriental"

SELECT CONCAT(nombre, ' ', apellidos1, ' ', apellidos2) AS nombre\_completo

FROM clientes

WHERE ciudad\_nacimiento IN (

SELECT id FROM ciudades WHERE nombre IN ('Madrid', 'Barcelona')

) AND id\_cliente IN (

SELECT cliente FROM ventas WHERE cuadro IN (

SELECT id\_cuadro FROM cuadros WHERE categoria = (

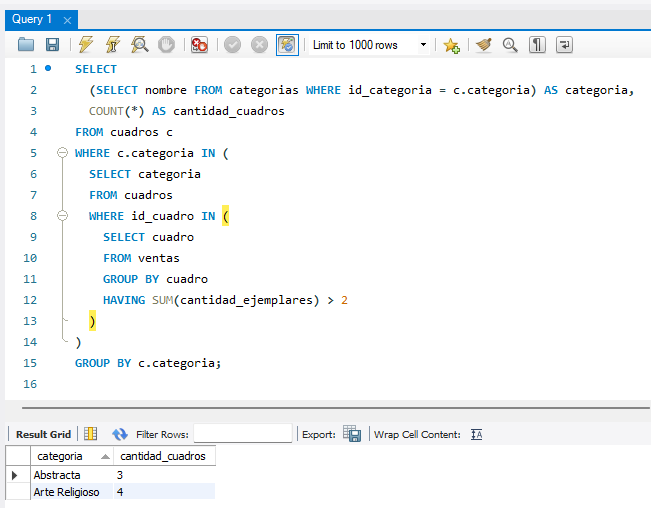
SELECT id\_categoria FROM categorias WHERE nombre = 'Arte Oriental'

)

)

);

* Consulta de selección que muestre el nombre de las categorías registradas y la cantidad de cuadros que hay asociados a cada una de ellas, siempre y cuando hayan vendido más de 2 ejemplares



-- 2. Categorías con más de 2 ejemplares vendidos

SELECT

(SELECT nombre FROM categorias WHERE id\_categoria = c.categoria) AS categoria,

COUNT(\*) AS cantidad\_cuadros

FROM cuadros c

WHERE c.categoria IN (

SELECT categoria

FROM cuadros

WHERE id\_cuadro IN (

SELECT cuadro

FROM ventas

GROUP BY cuadro

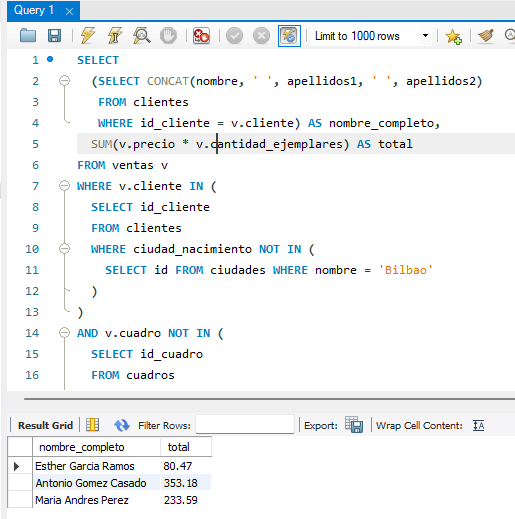
HAVING SUM(cantidad\_ejemplares) > 2

)

)

GROUP BY c.categoria;

* Consulta de selección que muestre el nombre completo de los clientes y el coste total que han gastado en la compra de cuadros, siempre y cuando los clientes no residan en Bilbao y los cuadros no sean de la categoría “Abstracta”



-- 3. Clientes y coste total si no son de Bilbao ni compraron cuadros "Abstracta"

SELECT

(SELECT CONCAT(nombre, ' ', apellidos1, ' ', apellidos2)

FROM clientes

WHERE id\_cliente = v.cliente) AS nombre\_completo,

SUM(v.precio \* v.cantidad\_ejemplares) AS total

FROM ventas v

WHERE v.cliente IN (

SELECT id\_cliente

FROM clientes

WHERE ciudad\_nacimiento NOT IN (

SELECT id FROM ciudades WHERE nombre = 'Bilbao'

)

)

AND v.cuadro NOT IN (

SELECT id\_cuadro

FROM cuadros

WHERE categoria = (

SELECT id\_categoria

FROM categorias

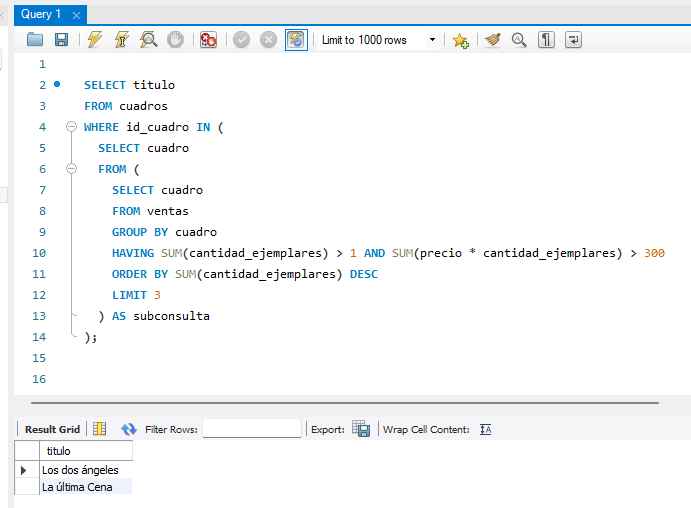
WHERE nombre = 'Abstracta'

)

)

GROUP BY v.cliente;

* Consulta de selección que muestre el título de los tres cuadros más vendidos hasta el momento, siempre que hayan vendido más de una unidad y superen los 300€ entre todas las ventas del ejemplar.



-- 4. Tres cuadros más vendidos (>1 unidad y >300€ en total)

SELECT titulo

FROM cuadros

WHERE id\_cuadro IN (

SELECT cuadro

FROM (

SELECT cuadro

FROM ventas

GROUP BY cuadro

HAVING SUM(cantidad\_ejemplares) > 1 AND SUM(precio \* cantidad\_ejemplares) > 300

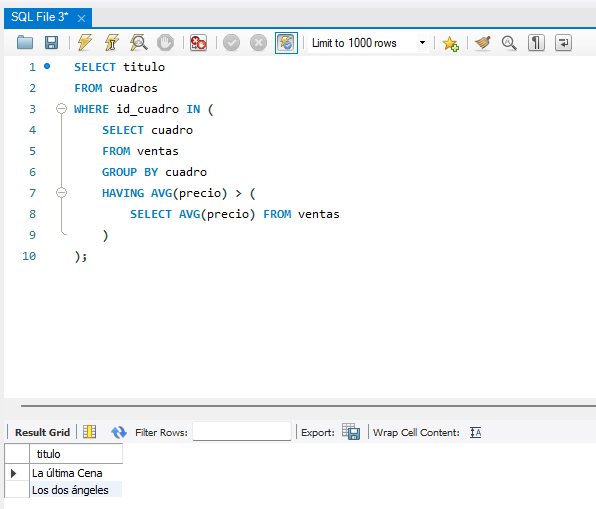
ORDER BY SUM(cantidad\_ejemplares) DESC

LIMIT 3

) AS subconsulta

);

* Consulta de selección que muestre el título de los cuadros cuyo precio supere la media del precio de todos los cuadros.



-- 5. Cuadros con precio superior a la media

SELECT titulo

FROM cuadros

WHERE id\_cuadro IN (

SELECT cuadro

FROM ventas

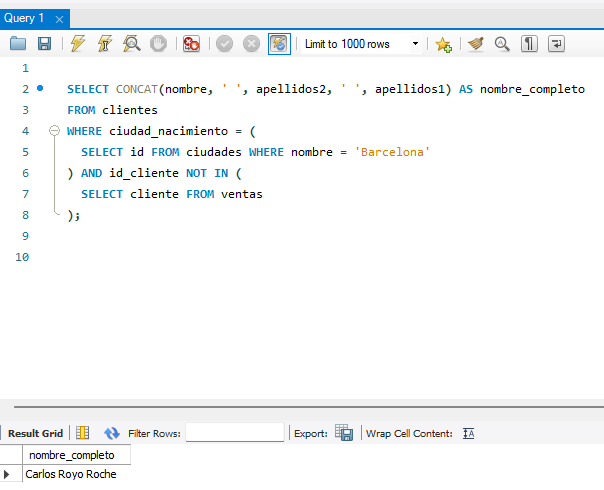
WHERE precio > (

SELECT AVG(precio) FROM ventas

)

);

* Consulta de selección que muestre el nombre completo de los clientes que aún no hayan comprado ningún cuadro y residan en Barcelona.



-- 6. Clientes que no han comprado cuadros y viven en Barcelona

SELECT CONCAT(nombre, ' ', apellidos2, ' ', apellidos1) AS nombre\_completo

FROM clientes

WHERE ciudad\_nacimiento = (

SELECT id FROM ciudades WHERE nombre = 'Barcelona'

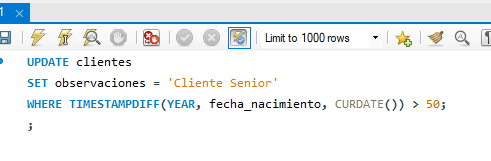
) AND id\_cliente NOT IN (

SELECT cliente FROM ventas

);

# CONSULTAS DE ACCIÓN

* Consulta de acción que permita modificar el campo observaciones para que a todos los clientes cuya edad supere los 50 años les aparezca el texto “Cliente Senior”



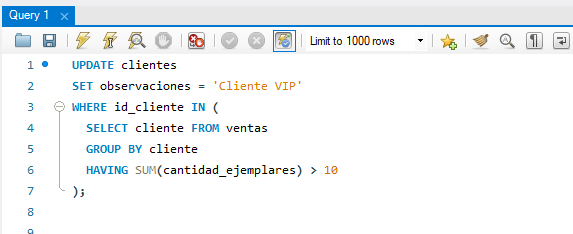
-- 1. Marcar clientes mayores de 50 como "Cliente Senior"

UPDATE clientes

SET observaciones = 'Cliente Senior'

WHERE TIMESTAMPDIFF(YEAR, fecha\_nacimiento, CURDATE()) > 50;

* Consulta de acción que permita modifica el campo observaciones para que a todos los clientes que hayan comprado más de 10 ejemplares les aparezca el texto “Cliente VIP”



-- 2. Marcar como "Cliente VIP" a quienes compraron más de 10 ejemplares

UPDATE clientes

SET observaciones = 'Cliente VIP'

WHERE id\_cliente IN (

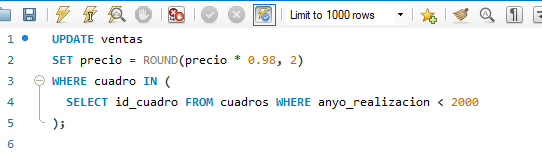
SELECT cliente FROM ventas

GROUP BY cliente

HAVING SUM(cantidad\_ejemplares) > 10

);

* Consulta de acción que permita modificar el precio de los cuadros, de forma que todos aquellos que se hayan pintado antes del año 2000 tengan una rebaja del 2%.



-- 3. Rebajar un 2% a cuadros pintados antes del año 2000

UPDATE ventas

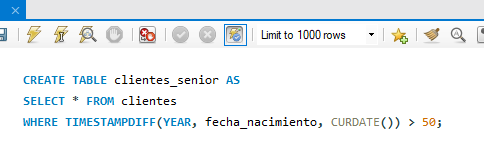
SET precio = ROUND(precio \* 0.98, 2)

WHERE cuadro IN (

SELECT id\_cuadro FROM cuadros WHERE anyo\_realizacion < 2000

);

* Nueva tabla llamada clientes\_senior que contenga todos los datos de los clientes senior existentes.



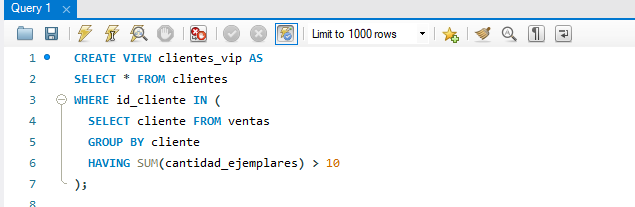
-- 4. Crear tabla clientes\_senior con datos

CREATE TABLE clientes\_senior AS

SELECT \* FROM clients

WHERE TIMESTAMPDIFF(YEAR, fecha\_nacimiento, CURDATE()) > 50;

* Vista llamada clientes\_vip que contenga todos los datos de los clientes vip existentes.



-- 5. Vista clientes\_vip

CREATE VIEW clientes\_vip AS

SELECT \* FROM clientes

WHERE id\_cliente IN (

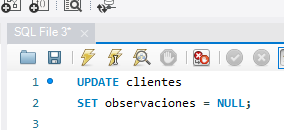
SELECT cliente FROM ventas

GROUP BY cliente

HAVING SUM(cantidad\_ejemplares) > 10

);

* Consulta de acción que permita borrar todas las observaciones realizadas en la entidad clientes.



-- 6. Borrar observaciones en clientes

UPDATE clientes

SET observaciones = NULL;