

M5_AE2_ABPRO-Ejercicio grupal

GRUPO 1

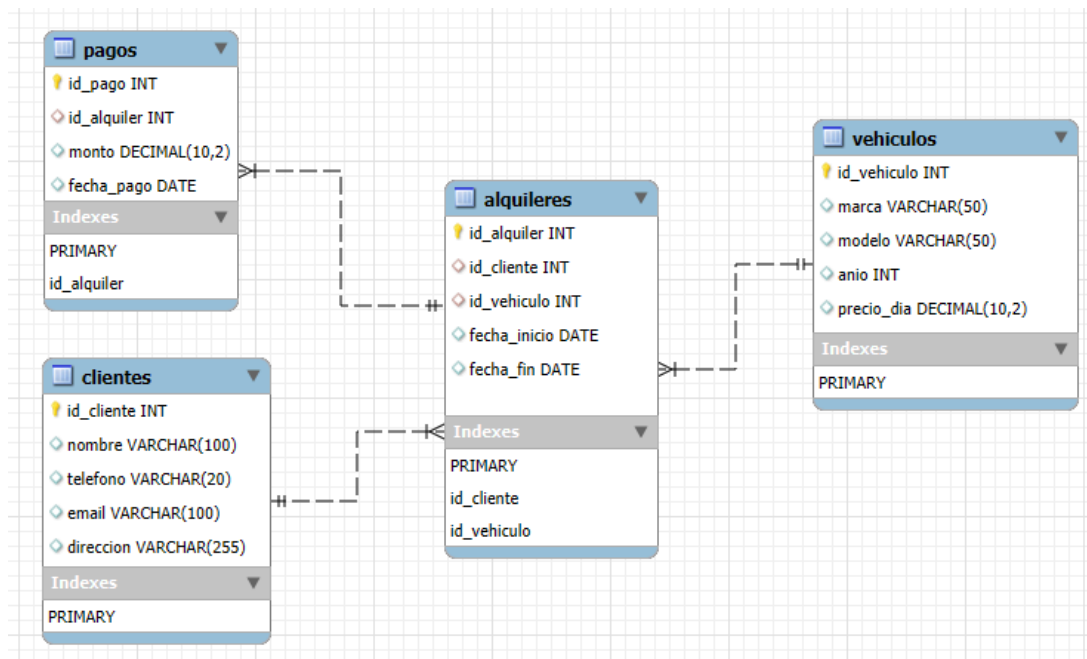
Integrantes:

Milenka Sgombich
Raúl Jaque E.
Felipe Lobiano C
Tomás Pérez Rocha
Jonathan Matus V.

Contexto

Imagina que un grupo de 4 amigos decide crear una plataforma de gestión para un negocio de alquiler de automóviles. Los datos del negocio están organizados en varias tablas dentro de una base de datos. El modelo de datos tiene las siguientes tablas:

1. Clientes
2. Vehículos
3. Alquileres
4. Pagos



Requerimiento

El equipo debe desarrollar consultas SQL para resolver los siguientes problemas a partir del modelo de datos proporcionado. Las consultas deben ser estructuradas y cumplir con las siguientes características:

CÓDIGO:

```
CREATE TABLE Clientes (  
    id_cliente INT PRIMARY KEY,  
    nombre VARCHAR (100),  
    telefono VARCHAR (15),  
    email VARCHAR (100),  
    direccion VARCHAR (100)  
);
```

```
CREATE TABLE Vehiculos (  
    id_vehiculo INT PRIMARY KEY,  
    marca VARCHAR (45),  
    modelo VARCHAR (45),  
    año INT,  
    precio_dia INT  
);
```

```
CREATE TABLE Alquileres (  
    id_alquiler INT PRIMARY KEY,  
    id_cliente INT,  
    id_vehiculo INT,  
    fecha_inicio DATETIME,  
    fecha_fin DATETIME,  
    FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente),  
    FOREIGN KEY (id_vehiculo) REFERENCES Vehiculos(id_vehiculo)  
);
```

```
CREATE TABLE Pagos (  
    id_pago INT PRIMARY KEY,  
    id_alquiler INT,  
    monto DECIMAL(10,2),  
    fecha_pago DATE  
);
```

-- Tabla Clientes

```
INSERT INTO Clientes (id_cliente, nombre, telefono, email, direccion) VALUES  
(1, 'Juan Pérez', '555-1234', 'juan@mail.com', 'Calle 123'),  
(2, 'Laura Gómez', '555-5678', 'laura@mail.com', 'Calle 456'),  
(3, 'Carlos Sánchez', '555-9101', 'carlos@mail.com', 'Calle 789');
```

```
-- Tabla Vehículos
INSERT INTO Vehiculos (id_vehiculo, marca, modelo, año, precio_dia) VALUES
(1, 'Toyota', 'Corolla', 2020, 30),
(2, 'Honda', 'Civic', 2019, 28),
(3, 'Ford', 'Focus', 2021, 35);

-- Tabla Alquileres
INSERT INTO Alquileres (id_alquiler, id_cliente, fecha_inicio, fecha_fin)
VALUES
(1, 1, '2025-03-10', '2025-03-15'),
(2, 2, '2025-03-12', '2025-03-16'),
(3, 3, '2025-03-20', '2025-03-22');

-- Tabla Pagos
INSERT INTO Pagos (id_pago, id_alquiler, monto, fecha_pago) VALUES
(1, 1, 150.00, '2025-03-12'),
(2, 2, 112.00, '2025-03-13'),
(3, 3, 70.00, '2025-03-20');
```

Consultas a realizar

Consulta 1: Mostrar el nombre, teléfono y email de todos los clientes que tienen un alquiler activo (es decir, cuya fecha actual esté dentro del rango entre fecha_inicio y fecha_fin).

En la base de datos no hay alquileres activos considerando la fecha actual, por lo que devuelve “empty”. Sin embargo, comparto una variante para consultar en un rango específico:

```
+-----+-----+-----+
| nombre | telefono | email |
+-----+-----+-----+
| Juan Pérez | 555-1234 | juan@mail.com |
| Laura Gómez | 555-5678 | laura@mail.com |
+-----+-----+-----+
2 rows in set (0.015 sec)
```

```
SELECT
    c.nombre,
    c.telefono,
    c.email
FROM
    Clientes c
JOIN
    Alquileres a ON c.id_cliente = a.id_cliente
WHERE
    '2025-03-14' BETWEEN a.fecha_inicio AND a.fecha_fin;
```

Consulta 2: Mostrar los vehículos que se alquilaron en el mes de marzo de 2025. Debe mostrar el modelo, marca, y precio_día de esos vehículos.

modelo	marca	precio_dia
Corolla	Toyota	30.00
Civic	Honda	28.00
Focus	Ford	35.00

3 rows in set (0.019 sec)

```
SELECT DISTINCT
  v.modelo,
  v.marca,
  v.precio_dia
FROM
  Vehiculos v
JOIN
  Alquileres a ON v.id_vehiculo = a.id_vehiculo
WHERE
  MONTH(a.fecha_inicio) = 3 AND YEAR(a.fecha_inicio) = 2025;
```

Consulta 3: Calcular el precio total del alquiler para cada cliente, considerando el número de días que alquiló el vehículo (el precio por día de cada vehículo multiplicado por la cantidad de días de alquiler).

nombre	modelo	dias_alquiler	precio_dia	total_a_pagar
Juan Pérez	Civic	5	28.00	140.00
Laura Gómez	Corolla	4	30.00	120.00
Carlos Sánchez	Focus	2	35.00	70.00

```
SELECT
  c.nombre,
  v.modelo,
  DATEDIFF(a.fecha_fin, a.fecha_inicio) AS dias_alquiler,
  v.precio_dia,
  DATEDIFF(a.fecha_fin, a.fecha_inicio) * v.precio_dia AS total_a_pagar
FROM Clientes c
JOIN Alquileres a ON c.id_cliente = a.id_cliente
JOIN Vehiculos v ON a.id_vehiculo = v.id_vehiculo;
```

Consulta 4: Encontrar los clientes que no han realizado ningún pago (no tienen registros en la tabla Pagos). Muestra su nombre y email.

```
mysql> SELECT c.nombre, c.email
-> FROM Clientes c
-> WHERE c.id_cliente NOT IN (
->     SELECT a.id_cliente
->     FROM Alquileres a
->     INNER JOIN Pagos p ON a.id_alquiler = p.id_alquiler
-> );
Empty set (0.037 sec)
```

```
SELECT c.nombre, c.email
FROM Clientes c
WHERE c.id_cliente NOT IN (
    SELECT a.id_cliente
    FROM Alquileres a
    INNER JOIN Pagos p ON a.id_alquiler = p.id_alquiler
);
```

Consulta 5: Calcular el promedio de los pagos realizados por cada cliente. Muestra el nombre del cliente y el promedio de pago.

```
+-----+-----+
| nombre      | promedio_pago |
+-----+-----+
| Juan Pérez  | 150.000000    |
| Laura Gómez | 112.000000    |
| Carlos Sánchez | 70.000000    |
+-----+-----+
3 rows in set (0.132 sec)
```

```
SELECT c.nombre, AVG(p.monto) AS promedio_pago
FROM Clientes c
JOIN Alquileres a ON c.id_cliente = a.id_cliente
JOIN Pagos p ON a.id_alquiler = p.id_alquiler
GROUP BY c.nombre;
```

Consulta 6: Mostrar los vehículos que están disponibles para alquilar en una fecha específica (por ejemplo, 2025-03-18). Debe mostrar el modelo, marca y precio_día. Si el vehículo está ocupado, no se debe incluir.

```
+-----+-----+-----+
| marca | modelo | precio_dia |
+-----+-----+-----+
| Toyota | Corolla | 30.00      |
| Honda  | Civic  | 28.00      |
| Ford   | Focus  | 35.00      |
+-----+-----+-----+
3 rows in set (0.083 sec)
```

```

SELECT v.marca, v.modelo, v.precio_dia
FROM Vehiculos v
WHERE v.id_vehiculo NOT IN (
SELECT a.id_vehiculo
FROM Alquileres a
WHERE '2025-03-18' BETWEEN a.fecha_inicio AND a.fecha_fin
);

```

Consulta 7: Encontrar la marca y el modelo de los vehículos que se alquilaron **más de una vez** en el mes de marzo de 2025.

```

mysql> SELECT v.marca , v.modelo
-> FROM Vehiculos v
-> INNER JOIN Alquileres a ON v.id_vehiculo = a.id_vehiculo
-> WHERE YEAR (a.fecha_inicio) = 2025 AND MONTH (a.fecha_inicio) = 3
-> GROUP BY v.id_vehiculo, v.marca, v.modelo
-> HAVING COUNT(a.id_alquiler) > 1;
Empty set (0.041 sec)

```

Ningún vehículo fue alquilado más de una vez en marzo de 2025, por lo que devuelve “empty”.

```

SELECT v.marca , v.modelo
FROM Vehiculos v
INNER JOIN Alquileres a ON v.id_vehiculo = a.id_vehiculo
WHERE YEAR (a.fecha_inicio) = 2025 AND MONTH (a.fecha_inicio) = 3
GROUP BY v.id_vehiculo, v.marca, v.modelo
HAVING COUNT(a.id_alquiler) > 1;

```

Consulta 8: Mostrar el total de monto pagado por cada cliente. Debe mostrar el nombre del cliente y la cantidad total de pagos realizados (suma del monto de los pagos).

```

+-----+-----+
| nombre      | total_pagado |
+-----+-----+
| Juan Pérez  | 150.00       |
| Laura Gómez | 112.00       |
| Carlos Sánchez | 70.00       |
+-----+-----+
3 rows in set (0.012 sec)

```

```

SELECT c.nombre, SUM(p.monto) AS total_pagado
FROM Clientes c
INNER JOIN alquileres a ON c.id_cliente = a.id_cliente
INNER JOIN Pagos p ON a.id_alquiler = p.id_alquiler
GROUP BY c.id_cliente, c.nombre;

```

Consulta 9: Mostrar los clientes que alquilaron el vehículo Ford Focus (con id_vehiculo = 3). Debe mostrar el nombre del cliente y la fecha del alquiler.

nombre	fecha_inicio	fecha_fin
Carlos Sánchez	2025-03-20	2025-03-22

1 row in set (0.042 sec)

```
SELECT c.nombre, a.fecha_inicio, a.fecha_fin
FROM Alquileres a
JOIN Clientes c ON a.id_cliente = c.id_cliente
WHERE a.id_vehiculo = 3;
```

Consulta 10: Realizar una consulta que muestre el nombre del cliente y el total de días alquilados de cada cliente, ordenado de mayor a menor total de días. El total de días es calculado como la diferencia entre fecha_inicio y fecha_fin.

nombre	total_dias
Juan Pérez	5
Laura Gómez	4
Carlos Sánchez	2

3 rows in set (0.083 sec)

```
SELECT c.nombre, SUM(DATEDIFF(a.fecha_fin, a.fecha_inicio)) AS total_dias
FROM Alquileres a
JOIN Clientes c ON a.id_cliente = c.id_cliente
GROUP BY c.id_cliente
ORDER BY total_dias DESC;
```

Requisitos

1. Utiliza llaves primarias y foráneas para establecer relaciones entre las tablas.
2. Debes utilizar funciones de agregación como SUM(), AVG(), COUNT(), entre otras.
3. Utiliza condiciones de selección adecuadas con WHERE y operadores lógicos.
4. Realiza consultas con JOINS, por ejemplo, para obtener información de varias tablas relacionadas.
5. Leer el modelo de datos correctamente y realizar consultas de selección complejas.

6. Asegúrate de que las consultas sean eficientes y devuelvan solo la información requerida.

Evaluación

El equipo será evaluado con base en los siguientes criterios:

1. Correcta aplicación de consultas SQL: Se evaluará si las consultas cumplen correctamente con los requerimientos.
2. Uso adecuado de funciones SQL: El uso de funciones como SUM(), AVG(), COUNT(), y JOIN será evaluado.
3. Optimización de consultas: Las consultas deben ser claras y eficientes.
4. Presentación y formato del archivo SQL: Las consultas deben estar bien estructuradas y legibles.