**AUTORENTAL**

# 

**LUIS MIGUEL CAICEDO BERMON**

**T2**

**PEDRO FELIPE GÓMEZ BONILLA**

**CAMPUSLANDS**

**SANDBOX  
RUTA JAVA**

**TIBÚ**

**2024**

Tabla de Contenidos

[**Introducción 4**](#_xy776j9pw1un)

[**Caso de Estudio 4**](#_x5wjmxr9s7lx)

[**Planificación 4**](#_8muzr1pu3mlz)

[Construcción del Modelo Conceptual 4](#_akgvn7l12xs2)

[Descripción 5](#_qxuss0va0z8)

[Gráfica 5](#_u45t31dnlow2)

[Descripción Técnica 5](#_gww41npptmyg)

[Construcción del Modelo Lógico 6](#_nzoi4crbe63p)

[Descripción 6](#_46z7wrz43fjb)

[Gráfica 6](#_cx639fevpp57)

[Descripción Técnica 6](#_vemj55jop1dw)

[Normalización del Modelo Lógico 7](#_j975ki11pvw9)

[Primera Forma Normal (1FN) 7](#_ipsnvw5v2v0u)

[Descripción 7](#_yudex7qmv5l6)

[Gráfica 7](#_u1g2e96ymz30)

[Descripción Técnica 7](#_rjmpapoxneod)

[Segunda Forma Normal (2FN) 7](#_3fyocgo001fx)

[Descripción 8](#_n31b5bkefr23)

[Gráfica 8](#_mywvu5xf7mjo)

[Descripción Técnica 8](#_i9ctpni5mu37)

[Tercera Forma Normal (3FN) 8](#_aeip5fb7t3s5)

[Descripción 8](#_v6plqfy3484u)

[Gráfica 8](#_bcqznd6b310i)

[Descripción Técnica 9](#_qaz530fwxr49)

[Construcción del Modelo Físico 9](#_muq8ro9qbje6)

[Descripción 9](#_5g32igevz4r8)

[Código 9](#_ryjwvstap29b)

[Descripción Técnica 10](#_f9n2yx58hlja)

[Diagrama E-R 10](#_l2vl1lqjv6nt)

[Descripción 10](#_b542o9532ses)

[Gráfica 10](#_8wg77tt5rsnt)

[Descripción Técnica 11](#_whj2rpgnzs4v)

[Tablas 11](#_s3pvkylovom)

[Descripción 11](#_ye40r8z6ajh3)

[Gráfica 11](#_mgogeuideyfr)

[Descripción Técnica 11](#_4tau2idq5j2j)

[Relaciones entre Tablas 11](#_v02u5svki3o6)

[Descripción 12](#_mn4p1kzcs5d)

[Gráfica 12](#_ahrg4l6677cx)

[Descripción Técnica 12](#_v3u2pmnyvhxt)

[Inserción de Datos 12](#_khdudtfa7c4u)

[Descripción 12](#_r9p210nyg848)

[Gráfica 12](#_wpdzy6svxf2h)

[Descripción Técnica 13](#_cudxay80qe23)

[**Referencias 13**](#_ywn691xazve)

# 

# Introducción

**AutoRental** es una empresa de alquiler de vehículos que necesita una BBDD para gestionar sus empleados, vehículos, clientes, sucursales y los alquileres que se realicen.

Esta empresa cuenta con 5 sucursales en diferentes ciudades y se proyecta a expandirse a otras ciudades del país y cuenta con una flota propia de vehículos de diferentes tipos, modelos (año), capacidad, etc.

# 

# Caso de Estudio

La empresa donde usted trabaja ha sido contratada para desarrollar un sistema de

información para una empresa de alquiler de vehículos llamada AutoRental, y usted ha sido

designado para diseñar una base de datos para ese sistema de información.

AutoRental cuenta con 5 sucursales en diferentes ciudades y se proyecta a expandirse a otras

ciudades del país y cuenta con una flota propia de vehículos de diferentes tipos, modelos

(año), capacidad, etc.

Los clientes de AutoRental podrán alquilar un vehículo en una sucursal y entregarlo en otra

sucursal.

AutoRental ofrece descuentos sobre diferentes tipos de vehículos a lo largo del año.

Los valores de alquiler dependen del tipo de vehículo (sedán, compacto, camioneta platón,

camioneta lujo, deportivo, etc) y se cobran por días y/o semanas. Por ejemplo, si un alquila

un vehículo por 9 días, el valor cotizado será de 1 semana y 2 días.

Si un cliente entrega el vehículo pasada la fecha de entrega contratada, se cobrarán los días

adicionales con un incremento del 8%.

**La base de datos deberá cumplir almacenar la siguiente información:**

• Sucursales: ciudad y dirección donde se ubica, teléfono fijo, celular y correo electrónico.

• Empleados: sucursal donde labora, cédula, nombres, apellidos, dirección y ciudad de

residencia, celular y correo electrónico.

• Clientes: cédula, nombres, apellidos, dirección y ciudad de residencia, celular y correo

electrónico.

• Vehículos: tipo de vehículo, placa, referencia, modelo, puertas, capacidad, sunroof, motor,

color.

• Alquileres: vehículo, cliente, empleado, sucursal y fecha de salida, sucursal y fecha de

llegada, fecha esperada de llegada, valor de alquiler por semana, valor de alquiler por día,

porcentaje de descuento, valor cotizado y valor pagado.

# Planificación

## Construcción del Modelo Conceptual

### Descripción

En este modelo fueron creadas 7 tablas con sus respectivos atributos:

**Sucursales:**

**ciudad**

**ubicación**

**teléfono fijo**

**celular**

**correo**

**Clientes:**

**nombres**

**apellidos**

**cedula**

**ciudad de residencia**

**dirección**

**celular**

**correo**

**Empleados:**

**sucursal**

**cedula**

**nombres**

**apellidos**

**ciudad**

**ubicación**

**celular**

**correo**

**Vehículos**

**tipo de vehículo**

**palca**

**referencia**

**modelo**

**puertas**

**capacidad**

**sunroof**

**motor**

**color**

**Tipo de vehículos:**

**tipo**

**valor de alquiler por semana**

**valor de alquiler por dia**

**Descuentos**

**tipo de vehículo**

**fecha de inicio**

**fecha de fin**

**porcentaje de descuento**

**Alquileres**

**vehículo**

**empleado**

**sucursal de salida**

**fecha de salida**

**sucursal de llegada**

**fecha esperada de llegada**

**fecha de llegada**

**valor cotizado**

**valor pagado**

### Gráfica

## Construcción del Modelo Lógico

### Descripción

El modelo lógico está basado en el modelo conceptual el cual fue de ayuda para identificar los tipos de datos en cada tabla agregando llaves primarias que permiten mejor identificación de los datos, llaves foráneas para relacionar las tablas y su cardinalidad.

**Sucursales:**

id int primary key

ciudad varchar(45)

direccion varchar(45)

teléfono int

celular int

correo varchar(45)

**Empleados:**

id int primary key

id\_sucursal int foreign key

nombre1 varchar(45)

nombre2 varchar(45),

apellido1 varchar(45)

apellido2 varchar(45),

direccion varchar(45),

ciudad varchar(45)

celular int

correo varchar(45)

**Clientes:**

id int primary key

cedula int

nombre1 varchar(45)

nombre2 varchar(45),

apellido1 varchar(45)

apellido2 varchar(45),

direccion varchar(45),

ciudad varchar(45)

celular int

correo varchar(45)

**Tipo de vehículos:**

id int primary key

tipo varchar(45)

valor\_semana int

valor\_dia int

**Descuentos**

id int primary key

id\_tipo int foreign key

fecha\_inicio date

fecha\_fin date

porcentaje\_descuento int

**Vehículos**

id int primary key

id\_tipo int foreign key

placa varchar(45)

referencia varchar(45)

modelo varchar(45)

puertas int

capacidad int

sunroof boolean

motor varchar(45)

color varchar(45)

**Alquileres**

id int primary key

id\_vehiculo int foreign key

id\_cliente int foreign key

id\_empleado int foreign key

id\_sucursal\_salida int foreign key

fecha\_salida date

id\_sucursal\_llegada int foreign key

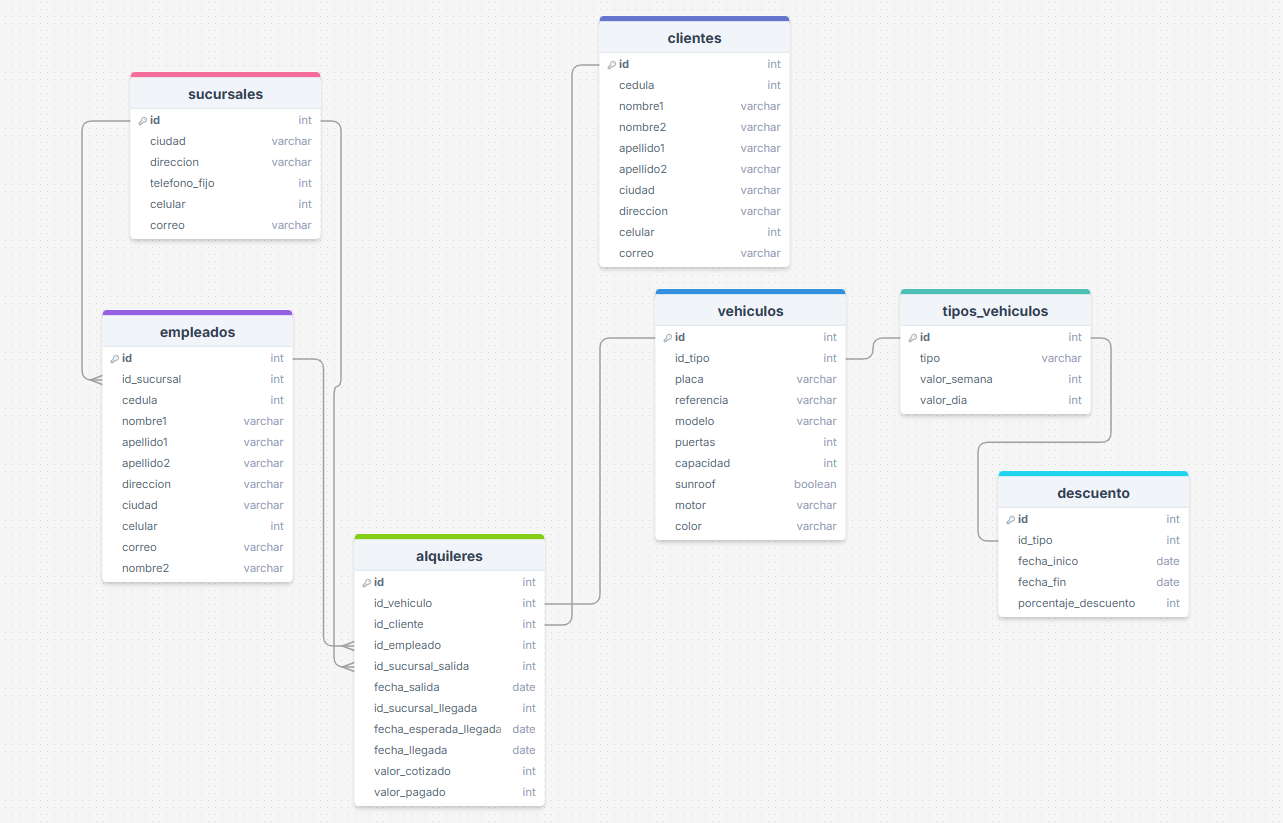
fecha\_esperada\_llegada date

fecha\_llegada date

valor\_cotizado int

valor\_pagado int

### Gráfica



## Normalización del Modelo Lógico

La normalización consiste en verificar que no hayan datos redundantes y que cumplan con las tres formas normales las cuales son las siguientes.

### Primera Forma Normal (1FN)

* Eliminar los grupos repetidos de las tablas individuales.
* Crear una tabla independiente para cada conjunto de datos relacionados.
* Identifique cada conjunto de datos relacionados con una clave principal.

#### Descripción

**sucursales:**

se encuentra en primera forma porque cumple con las normas ya que no tienen datos redundantes y la tabla es independiente.

**Empleados:**

se encuentra en primera forma porque cumple con las normas ya que no tienen datos redundantes y la tabla es independiente.

**Clientes:**

se encuentra en primera forma porque cumple con las normas ya que no tienen datos redundantes y la tabla es independiente.

**Tipo de vehículos:**

se encuentra en primera forma porque cumple con las normas ya que no tienen datos redundantes y la tabla es independiente.

**Descuentos**

se encuentra en primera forma porque cumple con las normas ya que no tienen datos redundantes y la tabla es independiente.

**Vehículos**

se encuentra en primera forma porque cumple con las normas ya que no tienen datos redundantes y la tabla es independiente.

**Alquileres**

se encuentra en primera forma porque cumple con las normas ya que no tienen datos redundantes y la tabla es independiente.

#### Gráfica

#### Descripción Técnica

### Segunda Forma Normal (2FN)

* Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.
* Relacione estas tablas con una clave externa.

Los registros no deben depender de nada más que de la clave principal de una tabla (una clave compuesta, si es necesario).

#### Descripción

**sucursales:**

se encuentra en segunda forma porque cumple con las normas ya que los registros no dependen de nada más que la llave principal y se relaciona con las demás tablas con una llave externa

**Empleados:**

se encuentra en segunda forma porque cumple con las normas ya que los registros no dependen de nada más que la llave principal y se relaciona con las demás tablas con una llave externa

**Clientes:**

se encuentra en segunda forma porque cumple con las normas ya que los registros no dependen de nada más que la llave principal y se relaciona con las demás tablas con una llave externa

**Tipo de vehículos:**

se encuentra en segunda forma porque cumple con las normas ya que los registros no dependen de nada más que la llave principal y se relaciona con las demás tablas con una llave externa

**Descuentos**

se encuentra en segunda forma porque cumple con las normas ya que los registros no dependen de nada más que la llave principal y se relaciona con las demás tablas con una llave externa

**Vehículos**

se encuentra en segunda forma porque cumple con las normas ya que los registros no dependen de nada más que la llave principal y se relaciona con las demás tablas con una llave externa

**Alquileres**

se encuentra en segunda forma porque cumple con las normas ya que los registros no dependen de nada más que la llave principal y se relaciona con las demás tablas con una llave externa

#### Gráfica

#### Descripción Técnica

### Tercera Forma Normal (3FN)

* Eliminar los campos que no dependen de la clave.

Los valores de un registro que no forman parte de la clave de ese registro no pertenecen a la tabla.

#### Descripción

**sucursales:**

se encuentra en tercera forma porque cumple con las normas ya que no hay campos que no dependan de la llave primaria

**Empleados:**

se encuentra en tercera forma porque cumple con las normas ya que no hay campos que no dependan de la llave primaria

**Clientes:**

se encuentra en tercera forma porque cumple con las normas ya que no hay campos que no dependan de la llave primaria

**Tipo de vehículos:**

se encuentra en tercera forma porque cumple con las normas ya que no hay campos que no dependan de la llave primaria

**Descuentos**

se encuentra en tercera forma porque cumple con las normas ya que no hay campos que no dependan de la llave primaria

**Vehículos**

se encuentra en tercera forma porque cumple con las normas ya que no hay campos que no dependan de la llave primaria

**Alquileres**

se encuentra en tercera forma porque cumple con las normas ya que no hay campos que no dependan de la llave primaria

#### Gráfica

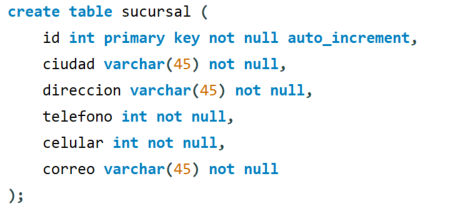
## Construcción del Modelo Físico

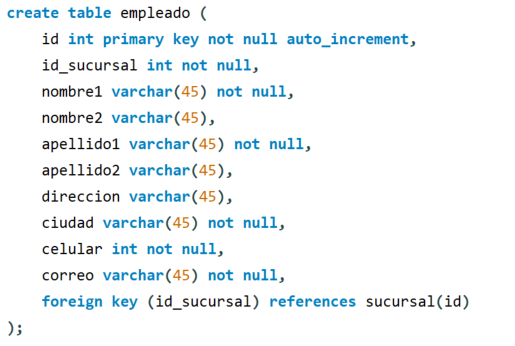
### Descripción

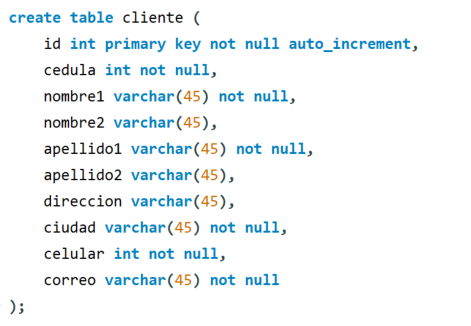
Este código puede crear la BBDD con las tablas necesarias para su funcionamiento teniendo en cuenta las relaciones entre ellas y los tipos de datos correspondientes.

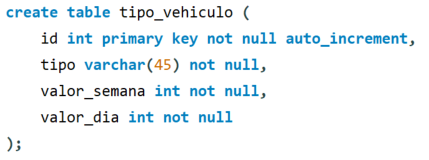
### Código

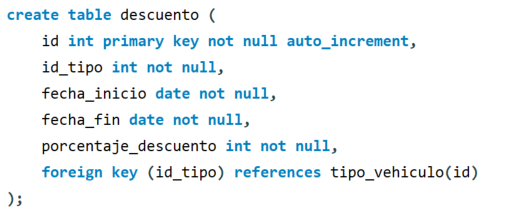


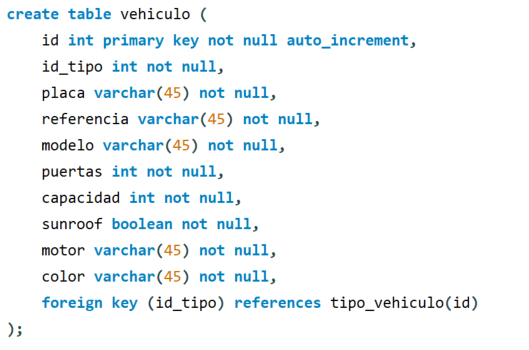


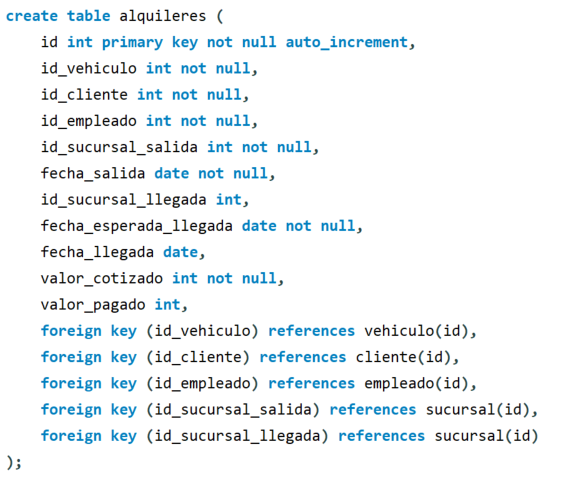










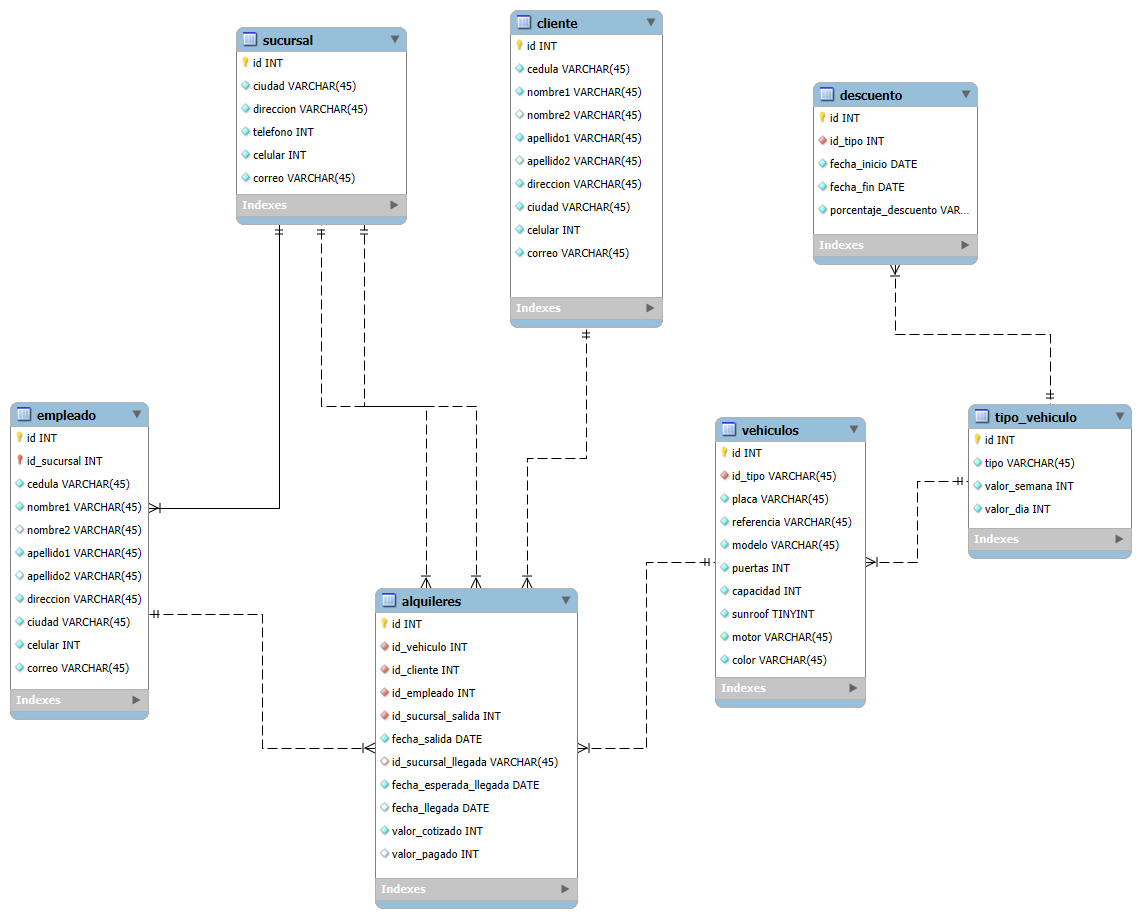


## Diagrama E-R

#### Descripción

Este diagrama permite visualizar todas las tablas en la BBDD con sus columnas y el tipo de dato que llevan se pueden ver las llaves primarias y también las foráneas y las relaciones entre ellas.

#### Gráfica



**inserciones de datos**

-- 1. Tabla `sucursal`

INSERT INTO sucursal (ciudad, direccion, telefono, celular, correo) VALUES

('Quito', 'Av. Amazonas 123', 2567890, 987654321, 'quito@sucursal.com'),

('Guayaquil', 'Calle 9 456', 4001234, 987123456, 'guayaquil@sucursal.com'),

('Cuenca', 'Calle Larga 789', 7234567, 987987654, 'cuenca@sucursal.com'),

('Manta', 'Av. Malecon 321', 3124567, 988765432, 'manta@sucursal.com'),

('Loja', 'Calle Central 654', 5512345, 987654123, 'loja@sucursal.com'),

('Ambato', 'Av. Cevallos 789', 3456789, 987654987, 'ambato@sucursal.com'),

('Riobamba', 'Calle Bolívar 321', 4567890, 987123987, 'riobamba@sucursal.com'),

('Tulcán', 'Calle Sucre 456', 5678901, 987654321, 'tulcan@sucursal.com'),

('Machala', 'Av. 25 de Junio 123', 6789012, 988765432, 'machala@sucursal.com'),

('Ibarra', 'Calle 24 de Mayo 789', 7890123, 987987654, 'ibarra@sucursal.com');

-- 2. Tabla `tipo\_vehiculo`

INSERT INTO tipo\_vehiculo (tipo, valor\_semana, valor\_dia) VALUES

('Sedán', 350, 60),

('SUV', 500, 80),

('Camioneta', 600, 100),

('Deportivo', 1000, 150),

('Pickup', 550, 90),

('Hatchback', 300, 50),

('Minivan', 650, 120),

('Coupé', 800, 130),

('Convertible', 1200, 200),

('Furgoneta', 400, 70),

('Minibús', 750, 130),

('Furgón', 500, 85),

('Camion', 1000, 170),

('SUV Compacta', 400, 75),

('Monovolumen', 650, 110),

('Auto eléctrico', 900, 160),

('Deportivo convertible', 1300, 210),

('Camper', 800, 135),

('Moped', 150, 25),

('Motocicleta', 250, 45);

-- 3. Tabla `descuento`

INSERT INTO descuento (id\_tipo, fecha\_inicio, fecha\_fin, porcentaje\_descuento) VALUES

(1, '2024-01-01', '2024-01-31', 10),

(2, '2024-02-01', '2024-02-28', 15),

(3, '2024-03-01', '2024-03-31', 20),

(4, '2024-04-01', '2024-04-30', 5),

(5, '2024-05-01', '2024-05-31', 12),

(6, '2024-06-01', '2024-06-30', 18),

(7, '2024-07-01', '2024-07-31', 14),

(8, '2024-08-01', '2024-08-31', 22),

(9, '2024-09-01', '2024-09-30', 25),

(10, '2024-10-01', '2024-10-31', 10),

(11, '2024-11-01', '2024-11-30', 15);

-- 4. Tabla `empleado`

INSERT INTO empleado (id\_sucursal, nombre1, nombre2, apellido1, apellido2, direccion, ciudad, celular, correo) VALUES

(1, 'Juan', 'Carlos', 'Pérez', 'Gómez', 'Av. 6 de Diciembre 456', 'Quito', 987654321, 'juanperez@empresa.com'),

(2, 'María', 'Isabel', 'Rodríguez', 'Díaz', 'Calle Guayaquil 789', 'Guayaquil', 987123456, 'mariarodriguez@empresa.com'),

(3, 'Luis', 'Fernando', 'Martínez', 'López', 'Calle Victoria 123', 'Cuenca', 988765432, 'luismartinez@empresa.com'),

(4, 'Ana', 'Paz', 'Jiménez', 'Moreno', 'Av. Malecon 321', 'Manta', 987654123, 'anajimenez@empresa.com'),

(5, 'Carlos', 'Eduardo', 'Mendoza', 'Silva', 'Calle Central 654', 'Loja', 987123654, 'carlosmendoza@empresa.com'),

(1, 'Verónica', 'Paola', 'González', 'Ruiz', 'Calle Pichincha 123', 'Quito', 987654789, 'veronicagonzalez@empresa.com'),

(2, 'Eduardo', 'Ángel', 'Sánchez', 'Ramírez', 'Calle Bolívar 456', 'Guayaquil', 987654321, 'eduardosanchez@empresa.com'),

(3, 'Andrea', 'Lina', 'Fernández', 'Vega', 'Av. 12 de Octubre 789', 'Cuenca', 988765432, 'andreafernandez@empresa.com'),

(4, 'Ricardo', 'José', 'Hernández', 'Paredes', 'Calle Malecón 567', 'Manta', 987654123, 'ricardohernandez@empresa.com'),

(5, 'Patricia', 'Esther', 'Ruiz', 'Molina', 'Calle Sucre 234', 'Loja', 987654987, 'patriciaruiz@empresa.com'),

(6, 'Martín', 'Eduardo', 'López', 'Torres', 'Calle Comercio 567', 'Ambato', 987654321, 'martinlopez@empresa.com'),

(7, 'Patricia', 'Nathaly', 'Salazar', 'Vega', 'Av. Machala 123', 'Riobamba', 987123456, 'patriciasalazar@empresa.com'),

(8, 'Héctor', 'Luis', 'Castillo', 'Rivas', 'Calle Comercio 987', 'Tulcán', 988765432, 'hectorcastillo@empresa.com'),

(9, 'Marta', 'Isabel', 'Mendoza', 'Jiménez', 'Calle Sucre 234', 'Machala', 987654123, 'martamendoza@empresa.com'),

(10, 'Gonzalo', 'José', 'Ramírez', 'Sánchez', 'Calle 12 de Octubre 456', 'Ibarra', 987654789, 'gonzaloramirez@empresa.com'),

(6, 'Julia', 'Paola', 'Martínez', 'López', 'Calle 24 de Mayo 654', 'Ambato', 987123654, 'juliamartinez@empresa.com'),

(7, 'Ricardo', 'David', 'Rodríguez', 'Serrano', 'Calle Bolívar 321', 'Riobamba', 987654987, 'ricardorodriguez@empresa.com'),

(8, 'Carlos', 'Antonio', 'Pérez', 'Fernández', 'Calle 12 de Octubre 789', 'Tulcán', 988765123, 'carlosperez@empresa.com'),

(9, 'Lorena', 'Esther', 'González', 'Hernández', 'Av. 25 de Junio 789', 'Machala', 987654321, 'lorenagonzalez@empresa.com'),

(10, 'Victor', 'Héctor', 'Jiménez', 'Díaz', 'Calle 15 de Septiembre 987', 'Ibarra', 987654123, 'victorjimenez@empresa.com'),

(1, 'José', 'Luis', 'Vega', 'González', 'Calle Los Pinos 123', 'Quito', 987654321, 'joseluisvega@empresa.com'),

(2, 'Santiago', 'Alfredo', 'Torres', 'Martínez', 'Av. 9 de Octubre 456', 'Guayaquil', 987123456, 'santiagotorres@empresa.com'),

(3, 'Catalina', 'Isabel', 'Romero', 'Flores', 'Calle La Libertad 789', 'Cuenca', 988765432, 'catalinaromero@empresa.com'),

(4, 'Alfredo', 'Manuel', 'Ramírez', 'Serrano', 'Av. 25 de Julio 123', 'Manta', 987654123, 'alfredoramirez@empresa.com'),

(5, 'Patricia', 'María', 'Sánchez', 'Pérez', 'Calle Comercio 567', 'Loja', 987654789, 'patriciasanchez@empresa.com'),

(6, 'Carlos', 'Eduardo', 'Rodríguez', 'Guerra', 'Calle Los Andes 345', 'Ambato', 987123654, 'carlosrodriguez@empresa.com'),

(7, 'Alicia', 'Raquel', 'González', 'Sánchez', 'Av. Las Américas 789', 'Riobamba', 987654987, 'aliciagonzalez@empresa.com'),

(8, 'Antonio', 'Luis', 'Cruz', 'Suárez', 'Calle San José 234', 'Tulcán', 988765123, 'antoniocruz@empresa.com'),

(9, 'Bárbara', 'Natalia', 'Vargas', 'Jiménez', 'Av. Los Naranjos 678', 'Machala', 987654321, 'barbaravargas@empresa.com'),

(10, 'José', 'Antonio', 'Méndez', 'López', 'Calle La Paz 123', 'Ibarra', 987654654, 'joseantonio@empresa.com');

-- 5. Tabla `cliente` (Asegurándome que use el id como clave primaria)

INSERT INTO cliente (cedula, nombre1, nombre2, apellido1, apellido2, direccion, ciudad, celular, correo) VALUES

(1234567890, 'Carlos', 'Alberto', 'Moreno', 'González', 'Calle 1', 'Quito', 987654321, 'carlosmoreno@gmail.com'),

(1234567891, 'Sofía', 'Elena', 'Ramírez', 'Paredes', 'Calle 2', 'Guayaquil', 987123456, 'sofiaramirez@hotmail.com'),

(1234567892, 'Luis', 'Fernando', 'González', 'Lopez', 'Calle 3', 'Cuenca', 988765432, 'luisgonzalez@outlook.com'),

(1234567893, 'Ana', 'Lucía', 'Martínez', 'Díaz', 'Calle 4', 'Manta', 987654123, 'analucia@gmail.com'),

(1234567894, 'Javier', 'Antonio', 'Jiménez', 'Cordero', 'Calle 5', 'Loja', 987123654, 'javierjimenez@correo.com'),

(1234567895, 'Verónica', 'Fabiola', 'Pérez', 'Sánchez', 'Calle 6', 'Quito', 987654789, 'veronicaperez@gmail.com'),

(1234567896, 'Eduardo', 'José', 'Hernández', 'Serrano', 'Calle 7', 'Guayaquil', 987654321, 'eduardohernandez@outlook.com'),

(1234567897, 'Lucía', 'Fernanda', 'Mendoza', 'Quintero', 'Calle 8', 'Cuenca', 988765432, 'luciamendoza@gmail.com'),

(1234567898, 'Ricardo', 'Javier', 'Sánchez', 'Martínez', 'Calle 9', 'Manta', 987654123, 'ricardosanchez@correo.com'),

(1234567899, 'Patricia', 'Isabel', 'López', 'Mora', 'Calle 10', 'Loja', 987654987, 'patricialopez@empresa.com'),

(1234567900, 'Miguel', 'Ángel', 'Paredes', 'Sánchez', 'Calle 1', 'Ambato', 987654321, 'miguelparedes@correo.com'),

(1234567901, 'Lina', 'Paola', 'Castro', 'González', 'Calle 2', 'Riobamba', 987123456, 'linacastro@correo.com'),

(1234567902, 'Evelyn', 'Margarita', 'Romero', 'Vega', 'Calle 3', 'Tulcán', 988765432, 'evelynromero@correo.com'),

(1234567903, 'Carlos', 'Antonio', 'Sierra', 'Martínez', 'Calle 4', 'Machala', 987654123, 'carlossierra@correo.com'),

(1234567904, 'Isabel', 'Lucía', 'Ríos', 'Cordero', 'Calle 5', 'Ibarra', 987123654, 'isabelrios@correo.com'),

(1234567905, 'Laura', 'Elena', 'Gómez', 'Martínez', 'Calle 6', 'Ambato', 987654789, 'lauragomez@correo.com'),

(1234567906, 'Ernesto', 'José', 'Pérez', 'Vega', 'Calle 7', 'Riobamba', 987654321, 'ernestoperez@correo.com'),

(1234567907, 'Natalia', 'Fabiola', 'Ortega', 'Sánchez', 'Calle 8', 'Tulcán', 988765432, 'nataliaortega@correo.com'),

(1234567908, 'Manuel', 'David', 'Alvarado', 'Hernández', 'Calle 9', 'Machala', 987654123, 'manueldavid@correo.com'),

(1234567909, 'Paola', 'Esther', 'Castillo', 'Ríos', 'Calle 10', 'Ibarra', 987654987, 'paolacastillo@correo.com'),

(1234567910, 'Santiago', 'Ángel', 'Gómez', 'Reyes', 'Calle 1', 'Ambato', 987654321, 'santiago.gomez@correo.com'),

(1234567911, 'Carla', 'Patricia', 'Ruiz', 'Molina', 'Calle 2', 'Riobamba', 987123456, 'carla.ruiz@correo.com'),

(1234567912, 'Diana', 'Esther', 'Fernández', 'Torres', 'Calle 3', 'Cuenca', 988765432, 'diana.fernandez@correo.com'),

(1234567913, 'Miguel', 'Eduardo', 'Lopez', 'Cevallos', 'Calle 4', 'Manta', 987654123, 'miguel.lopez@correo.com'),

(1234567914, 'Vicky', 'Diana', 'González', 'Moreno', 'Calle 5', 'Loja', 987654789, 'vicky.gonzalez@correo.com'),

(1234567915, 'Jorge', 'Luis', 'Salazar', 'Mendoza', 'Calle 6', 'Ambato', 987123654, 'jorge.salazar@correo.com'),

(1234567916, 'Mariana', 'Lucía', 'Paredes', 'Díaz', 'Calle 7', 'Riobamba', 987654987, 'mariana.paredes@correo.com'),

(1234567917, 'Pedro', 'José', 'Cordero', 'García', 'Calle 8', 'Tulcán', 988765123, 'pedro.cordero@correo.com'),

(1234567918, 'Valentina', 'Elena', 'Castillo', 'Ramírez', 'Calle 9', 'Machala', 987654321, 'valentina.castillo@correo.com'),

(1234567919, 'Raúl', 'Ángel', 'Hernández', 'Paredes', 'Calle 10', 'Ibarra', 987654654, 'raul.hernandez@correo.com'),

(1234567920, 'José', 'Manuel', 'Suárez', 'Reyes', 'Calle 11', 'Guayaquil', 987654321, 'josemanuel@correo.com'),

(1234567921, 'Verónica', 'Elena', 'Zambrano', 'Paredes', 'Calle 12', 'Quito', 987654987, 'veronicazambrano@correo.com'),

(1234567922, 'Ricardo', 'Alfonso', 'Soto', 'Vega', 'Calle 13', 'Cuenca', 988765432, 'ricardosoto@correo.com'),

(1234567923, 'Sofía', 'Alejandra', 'Cordero', 'Sánchez', 'Calle 14', 'Manta', 987654123, 'sofia.cordero@correo.com'),

(1234567924, 'Andrés', 'David', 'Pérez', 'González', 'Calle 15', 'Loja', 987123654, 'andres.perez@correo.com'),

(101910, 'Carlos', 'Eduardo', 'Gomez', 'Martinez', 'Calle Falsa 123', 'Quito', 987654321, 'carlos.gomez@mail.com'),

(102020, 'Ana', 'Maria', 'Perez', 'Lopez', 'Av. Amazonas 456', 'Guayaquil', 998877665, 'ana.perez@mail.com'),

(103030, 'Luis', 'Fernando', 'Rodriguez', 'Garcia', 'Calle Olmedo 789', 'Cuenca', 961234567, 'luis.rodriguez@mail.com'),

(104040, 'Marta', 'Elena', 'Lozano', 'Sanchez', 'Calle 10 de Agosto 101', 'Quito', 912345678, 'marta.lozano@mail.com'),

(105050, 'Juan', 'Carlos', 'Fernandez', 'Vega', 'Av. del Parque 202', 'Guayaquil', 963852741, 'juan.fernandez@mail.com'),

(106060, 'Elena', 'Isabel', 'Gonzalez', 'Serrano', 'Calle Pichincha 303', 'Ambato', 978654321, 'elena.gonzalez@mail.com'),

(107070, 'Roberto', 'Manuel', 'Diaz', 'Rodriguez', 'Calle Morona 404', 'Cuenca', 964785302, 'roberto.diaz@mail.com'),

(108080, 'Sofia', 'Mercedes', 'Garcia', 'Ruiz', 'Calle Amazonas 505', 'Loja', 911223344, 'sofia.garcia@mail.com'),

(109090, 'Ricardo', 'Alfonso', 'Mendez', 'Chavez', 'Calle Bolívar 606', 'Machala', 924678290, 'ricardo.mendez@mail.com'),

(110100, 'Patricia', 'Elena', 'Salazar', 'Rodriguez', 'Av. San Francisco 707', 'Quito', 960123478, 'patricia.salazar@mail.com'),

(111110, 'Fernando', 'Manuel', 'Martinez', 'Ortega', 'Calle del Sol 808', 'Cuenca', 953626474, 'fernando.martinez@mail.com'),

(112120, 'Julia', 'Victoria', 'Castro', 'Moreno', 'Calle de la Luz 909', 'Guayaquil', 931233455, 'julia.castro@mail.com'),

(113130, 'Pedro', 'Antonio', 'Lopez', 'Salazar', 'Calle El Ávila 1010', 'Quito', 943567890, 'pedro.lopez@mail.com'),

(114140, 'Maria', 'Josefa', 'Vargas', 'Zambrano', 'Av. Ricardo Paredes 1111', 'Ambato', 968501234, 'maria.vargas@mail.com'),

(115150, 'Javier', 'Luis', 'Paredes', 'Salcedo', 'Calle Córdova 1212', 'Loja', 977654321, 'javier.paredes@mail.com'),

(116160, 'Isabel', 'Ana', 'Morales', 'Perez', 'Calle Sucre 1313', 'Machala', 931235678, 'isabel.morales@mail.com'),

(117170, 'Oscar', 'Raul', 'Jimenez', 'Torres', 'Calle Quito 1414', 'Cuenca', 968412345, 'oscar.jimenez@mail.com'),

(118180, 'Carmen', 'Laura', 'Ruiz', 'Lopez', 'Av. 9 de Octubre 1515', 'Guayaquil', 912345987, 'carmen.ruiz@mail.com'),

(119190, 'Santiago', 'Felipe', 'Ruiz', 'Gonzalez', 'Calle Real 1616', 'Quito', 982345678, 'santiago.ruiz@mail.com'),

(120200, 'Adriana', 'Sofia', 'Martinez', 'Rodriguez', 'Calle Montúfar 1717', 'Guayaquil', 964829301, 'adriana.martinez@mail.com'),

(121210, 'Carlos', 'Alfonso', 'Hernandez', 'Lopez', 'Av. 6 de Diciembre 1818', 'Ambato', 975620304, 'carlos.hernandez@mail.com'),

(122220, 'Miriam', 'Cecilia', 'Vargas', 'Chavez', 'Calle El Rosal 1919', 'Quito', 912345678, 'miriam.vargas@mail.com'),

(123230, 'Miguel', 'Angel', 'Paredes', 'Salcedo', 'Av. de los Shyris 2020', 'Cuenca', 958733490, 'miguel.paredes@mail.com'),

(124240, 'Isabel', 'Diana', 'Castro', 'Alvarez', 'Calle Manabí 2121', 'Guayaquil', 987654321, 'isabel.castro@mail.com'),

(125250, 'Javier', 'Alfredo', 'Jimenez', 'Fernandez', 'Calle Napo 2222', 'Ambato', 981234567, 'javier.jimenez@mail.com'),

(126260, 'Claudia', 'Elena', 'Salazar', 'Rivas', 'Av. Baca Ortiz 2323', 'Loja', 969875432, 'claudia.salazar@mail.com'),

(127270, 'Fernando', 'Emilio', 'Ruiz', 'Serrano', 'Calle Puyo 2424', 'Quito', 962345678, 'fernando.ruiz@mail.com'),

(128280, 'Patricia', 'Marina', 'Ortega', 'Chavez', 'Calle El Cajas 2525', 'Cuenca', 947362180, 'patricia.ortega@mail.com'),

(129290, 'Julian', 'Pablo', 'Lopez', 'Rodriguez', 'Calle 12 de Octubre 2626', 'Ambato', 931234567, 'julian.lopez@mail.com'),

(130300, 'Monica', 'Angelica', 'Ramos', 'Lopez', 'Av. La Prensa 2727', 'Guayaquil', 961245736, 'monica.ramos@mail.com'),

(131310, 'Carlos', 'Eduardo', 'Martinez', 'Perez', 'Calle de la Paz 2828', 'Loja', 963412589, 'carlos.martinez@mail.com'),

(132320, 'Esteban', 'Felipe', 'Serrano', 'Vega', 'Calle Norte 2929', 'Quito', 984563210, 'esteban.serrano@mail.com'),

(133330, 'Elena', 'Valentina', 'Ramirez', 'Mena', 'Av. Pichincha 3030', 'Guayaquil', 998754321, 'elena.ramirez@mail.com'),

(134340, 'Lucia', 'Teresa', 'Gonzalez', 'Martinez', 'Calle Sucre 3131', 'Cuenca', 912763408, 'lucia.gonzalez@mail.com'),

(135350, 'Sofia', 'Paola', 'Vega', 'Cordero', 'Calle Quito 3232', 'Ambato', 957843210, 'sofia.vega@mail.com'),

(136360, 'Roberto', 'Santiago', 'Mendoza', 'Lopez', 'Av. Central 3333', 'Loja', 943245789, 'roberto.mendoza@mail.com'),

(137370, 'David', 'Antonio', 'Salazar', 'Chavez', 'Calle El Bosque 3434', 'Guayaquil', 924578901, 'david.salazar@mail.com'),

(138380, 'Margarita', 'Esther', 'Martinez', 'Gomez', 'Calle Parques 3535', 'Quito', 936540271, 'margarita.martinez@mail.com'),

(139390, 'Ricardo', 'Fabián', 'Navarro', 'Romero', 'Av. Granados 3636', 'Quito', 993456789, 'ricardo.navarro@mail.com'),

(140400, 'Eva', 'Lucia', 'Moreno', 'Gonzalez', 'Calle del Oro 3737', 'Guayaquil', 932423567, 'eva.moreno@mail.com'),

(141410, 'Daniel', 'Felipe', 'Lopez', 'Ramirez', 'Calle Naranja 3838', 'Ambato', 948563421, 'daniel.lopez@mail.com'),

(142420, 'Lorena', 'Isabel', 'Paredes', 'Serrano', 'Av. Calzado 3939', 'Quito', 959573216, 'lorena.paredes@mail.com'),

(143430, 'Juan', 'Carlos', 'Vega', 'Guzmán', 'Calle 24 de Mayo 4040', 'Cuenca', 913489653, 'juan.vega@mail.com'),

(144440, 'Mercedes', 'Natalia', 'Sanchez', 'Chavez', 'Calle Los Cerezos 4141', 'Guayaquil', 923849601, 'mercedes.sanchez@mail.com'),

(145450, 'Alejandro', 'Gabriel', 'Garcia', 'Lopez', 'Calle Los Andes 4242', 'Ambato', 937234850, 'alejandro.garcia@mail.com'),

(146460, 'Carlos', 'Eduardo', 'Martinez', 'Mena', 'Calle Luis Cordero 4343', 'Loja', 964823547, 'carlos.martinez@mail.com'),

(147470, 'Rosa', 'Elena', 'Vargas', 'Rodriguez', 'Av. Pichincha 4444', 'Quito', 962734598, 'rosa.vargas@mail.com'),

(148480, 'Fernando', 'Adrián', 'Gomez', 'Moreno', 'Calle 3 de Noviembre 4545', 'Cuenca', 973847650, 'fernando.gomez@mail.com'),

(149490, 'Silvia', 'Cecilia', 'Martinez', 'Paredes', 'Calle Amazonas 4646', 'Guayaquil', 981234509, 'silvia.martinez@mail.com'),

(150500, 'Guillermo', 'Antonio', 'Hernandez', 'Martinez', 'Calle La Prensa 4747', 'Ambato', 938374560, 'guillermo.hernandez@mail.com'),

(151510, 'Marta', 'Carmen', 'Rodriguez', 'Vega', 'Av. del Parque 4848', 'Loja', 953820463, 'marta.rodriguez@mail.com'),

(152520, 'José', 'Manuel', 'Lopez', 'Castro', 'Calle Santa Lucia 4949', 'Quito', 949203845, 'jose.lopez@mail.com'),

(153530, 'Patricia', 'Sofía', 'Martínez', 'Gómez', 'Calle Orellana 5050', 'Cuenca', 977134250, 'patricia.martinez@mail.com'),

(154540, 'Luis', 'Raul', 'Salazar', 'Gomez', 'Calle Los Vergeles 5151', 'Guayaquil', 951234678, 'luis.salazar@mail.com'),

(155550, 'Veronica', 'Andrea', 'Gonzalez', 'Serrano', 'Av. Los Cedros 5252', 'Ambato', 927843516, 'veronica.gonzalez@mail.com'),

(156560, 'Ricardo', 'Manuel', 'Perez', 'Vargas', 'Calle de la Alegría 5353', 'Loja', 951876432, 'ricardo.perez@mail.com'),

(157570, 'Sandra', 'Elena', 'Castro', 'Mena', 'Calle del Sol 5454', 'Quito', 986543210, 'sandra.castro@mail.com'),

(158580, 'Javier', 'Esteban', 'Fernandez', 'Paredes', 'Av. de los Shyris 5555', 'Guayaquil', 939876542, 'javier.fernandez@mail.com');

-- 6. Tabla `vehiculo`

INSERT INTO vehiculo (id\_tipo, placa, referencia, modelo, puertas, capacidad, sunroof, motor, color) VALUES

(1, 'ABC123', 'Toyota Corolla', '2020', 4, 5, false, '1.8L', 'Blanco'),

(2, 'XYZ456', 'Hyundai Tucson', '2021', 4, 5, true, '2.0L', 'Negro'),

(3, 'LMN789', 'Ford Ranger', '2019', 4, 5, false, '3.2L', 'Rojo'),

(4, 'DEF123', 'Chevrolet Camaro', '2022', 2, 4, true, '6.2L', 'Amarillo'),

(5, 'GHI456', 'Nissan X-Trail', '2020', 5, 7, true, '2.5L', 'Plata'),

(6, 'JKL789', 'Honda Civic', '2021', 4, 5, false, '2.0L', 'Azul'),

(7, 'MNO123', 'Mazda CX-5', '2021', 5, 5, true, '2.5L', 'Blanco'),

(8, 'PQR456', 'BMW M3', '2020', 4, 4, true, '3.0L', 'Gris'),

(9, 'STU789', 'Ford Fiesta', '2019', 4, 5, false, '1.6L', 'Verde'),

(10, 'VWX123', 'Jeep Grand Cherokee', '2022', 5, 7, true, '3.6L', 'Negro'),

(11, 'ZZZ111', 'Volkswagen Polo', '2021', 4, 5, false, '1.6L', 'Rojo'),

(12, 'YYY222', 'Chevrolet Spin', '2020', 5, 7, false, '1.8L', 'Blanco'),

(13, 'XXX333', 'Kia Sportage', '2021', 5, 5, true, '2.4L', 'Negro'),

(14, 'WWW444', 'Mitsubishi Outlander', '2020', 5, 7, true, '2.0L', 'Gris'),

(15, 'VVV555', 'Renault Duster', '2021', 5, 5, false, '1.6L', 'Azul'),

(16, 'UUU666', 'Jeep Cherokee', '2021', 5, 7, true, '3.2L', 'Verde'),

(17, 'TTT777', 'Ford EcoSport', '2022', 5, 5, false, '2.0L', 'Plata'),

(18, 'SSS888', 'Toyota Hilux', '2022', 4, 5, false, '3.0L', 'Negro'),

(19, 'RRR999', 'Honda CR-V', '2021', 5, 5, true, '2.4L', 'Blanco'),

(20, 'PPP000', 'Mazda 3', '2020', 4, 5, false, '2.0L', 'Rojo'),

(1, 'AAA111', 'Toyota Corolla', '2019', 4, 5, false, '1.6L', 'Blanco'),

(2, 'BBB222', 'Honda CR-V', '2020', 5, 5, true, '2.5L', 'Azul'),

(3, 'CCC333', 'Chevrolet Silverado', '2021', 4, 5, false, '5.3L', 'Negro'),

(4, 'DDD444', 'BMW X5', '2020', 5, 7, true, '3.0L', 'Gris'),

(5, 'EEE555', 'Audi Q7', '2021', 5, 7, true, '3.0L', 'Plata'),

(6, 'FFF666', 'Ford Escape', '2021', 5, 5, false, '2.5L', 'Rojo'),

(7, 'GGG777', 'Mazda CX-9', '2020', 5, 7, true, '2.5L', 'Verde'),

(8, 'HHH888', 'Volkswagen Tiguan', '2020', 5, 5, false, '2.0L', 'Blanco'),

(9, 'III999', 'Jeep Wrangler', '2022', 5, 5, true, '3.6L', 'Negro'),

(10, 'JJJ000', 'Subaru Outback', '2021', 5, 7, true, '2.5L', 'Azul');

-- 7. Tabla `alquileres` (Ahora usa el id de cliente como referencia, no la cedula)

INSERT INTO alquileres (id\_vehiculo, id\_cliente, id\_empleado, id\_sucursal\_salida, fecha\_salida, id\_sucursal\_llegada, fecha\_esperada\_llegada, fecha\_llegada, valor\_cotizado, valor\_pagado) VALUES

(1, 1, 1, 1, '2024-01-01', 2, '2024-01-07', '2024-01-10', 350, 350),

(2, 2, 2, 2, '2024-01-05', 3, '2024-01-12', '2024-01-11', 500, 500),

(3, 3, 3, 3, '2024-01-10', 4, '2024-01-15', '2024-01-14', 600, 600),

(4, 4, 4, 4, '2024-02-01', 5, '2024-02-05', null, 1000, 1000),

(5, 5, 5, 5, '2024-02-10', 1, '2024-02-15', '2024-02-14', 550, 550),

(6, 6, 6, 1, '2024-03-01', 2, '2024-03-07', '2024-03-06', 600, 600),

(7, 7, 7, 2, '2024-03-02', 3, '2024-03-09', '2024-03-08', 800, 800),

(8, 8, 8, 3, '2024-03-03', 4, '2024-03-10', '2024-03-09', 1200, 1200),

(9, 9, 9, 4, '2024-03-04', 5, '2024-03-11', '2024-03-10', 550, 550),

(10, 10, 10, 5, '2024-03-05', 1, '2024-03-12', '2024-03-11', 400, 400),

(11, 1, 1, 1, '2024-04-01', 2, '2024-04-07', '2024-04-06', 350, 350),

(12, 2, 2, 2, '2024-04-05', 3, '2024-04-12', '2024-04-11', 500, 500),

(13, 3, 3, 3, '2024-04-10', 4, '2024-04-15', '2024-04-14', 600, 600),

(14, 4, 4, 4, '2024-05-01', 5, '2024-05-05', '2024-05-04', 1000, 1000),

(15, 5, 5, 5, '2024-05-10', 1, '2024-05-15', '2024-05-14', 550, 550),

(16, 6, 6, 1, '2024-06-01', 2, '2024-06-07', '2024-06-06', 600, 600),

(17, 7, 7, 2, '2024-06-02', 3, '2024-06-09', '2024-06-08', 800, 800),

(18, 8, 8, 3, '2024-06-03', 4, '2024-06-10', '2024-06-09', 1200, 1200),

(19, 9, 9, 4, '2024-06-04', 5, '2024-06-11', '2024-06-10', 550, 550),

(20, 10, 10, 5, '2024-06-05', 1, '2024-06-12', '2024-06-11', 400, 400),

(1, 1, 1, 1, '2024-07-01', 2, '2024-07-07', '2024-07-06', 350, 350),

(2, 2, 2, 2, '2024-07-05', 3, '2024-07-12', null, 500, 500),

(3, 3, 3, 3, '2024-07-10', 4, '2024-07-15', '2024-07-14', 600, 600),

(4, 4, 4, 4, '2024-08-01', 5, '2024-08-05', '2024-08-04', 1000, 1000),

(5, 5, 5, 5, '2024-08-10', 1, '2024-08-15', '2024-08-14', 550, 550),

(6, 6, 6, 1, '2024-09-01', 2, '2024-09-07', '2024-09-06', 600, 600),

(7, 7, 7, 2, '2024-09-02', 3, '2024-09-09', '2024-09-08', 800, 800),

(8, 8, 8, 3, '2024-09-03', 4, '2024-09-10', '2024-09-09', 1200, 1200),

(9, 9, 9, 4, '2024-09-04', 5, '2024-09-11', '2024-09-10', 550, 550),

(10, 10, 10, 5, '2024-09-05', 1, '2024-09-12', '2024-09-11', 400, 400),

(1, 1, 1, 1, '2024-01-01', 2, '2024-01-07', '2024-01-06', 350, 350),

(2, 2, 2, 2, '2024-01-02', 3, '2024-01-09', '2024-01-08', 500, 500),

(3, 3, 3, 3, '2024-01-03', 4, '2024-01-10', '2024-01-09', 600, 600),

(4, 4, 4, 4, '2024-01-04', 5, '2024-01-11', '2024-01-10', 1000, 1000),

(5, 5, 5, 5, '2024-01-05', 1, '2024-01-12', '2024-01-11', 550, 550),

(6, 6, 6, 1, '2024-02-01', 2, '2024-02-07', '2024-02-06', 600, 600),

(7, 7, 7, 2, '2024-02-02', 3, '2024-02-09', '2024-02-08', 800, 800),

(8, 8, 8, 3, '2024-02-03', 4, '2024-02-10', '2024-02-09', 1200, 1200),

(9, 9, 9, 4, '2024-02-04', 5, '2024-02-11', '2024-02-10', 550, 550),

(10, 10, 10, 5, '2024-02-05', 1, '2024-02-12', '2024-02-11', 400, 400),

(11, 11, 1, 1, '2024-03-01', 2, '2024-03-07', '2024-03-06', 350, 350),

(12, 12, 2, 2, '2024-03-02', 3, '2024-03-09', '2024-03-08', 500, 500),

(13, 13, 3, 3, '2024-03-03', 4, '2024-03-10', '2024-03-09', 600, 600),

(14, 14, 4, 4, '2024-03-04', 5, '2024-03-11', '2024-03-10', 1000, 1000),

(15, 15, 5, 5, '2024-03-05', 1, '2024-03-12', '2024-03-11', 550, 550),

(16, 16, 6, 1, '2024-04-01', 2, '2024-04-07', '2024-04-06', 600, 600),

(17, 17, 7, 2, '2024-04-02', 3, '2024-04-09', '2024-04-08', 800, 800),

(18, 18, 8, 3, '2024-04-03', 4, '2024-04-10', '2024-04-09', 1200, 1200),

(5, 19, 1, 1, '2024-07-01', 2, '2024-07-07', '2024-07-06', 350, 350),

(6, 20, 2, 2, '2024-07-02', 3, '2024-07-09', '2024-07-08', 500, 500),

(7, 21, 3, 3, '2024-07-03', 4, '2024-07-10', '2024-07-09', 600, 600),

(8, 22, 4, 4, '2024-07-04', 5, '2024-07-11', '2024-07-10', 1000, 1000),

(9, 23, 5, 5, '2024-07-05', 1, '2024-07-12', '2024-07-11', 550, 550),

(10, 24, 6, 1, '2024-08-01', 2, '2024-08-07', '2024-08-06', 600, 600),

(11, 25, 7, 2, '2024-08-02', 3, '2024-08-09', '2024-08-08', 800, 800),

(12, 26, 8, 3, '2024-08-03', 4, '2024-08-10', '2024-08-09', 1200, 1200),

(13, 27, 9, 4, '2024-08-04', 5, '2024-08-11', '2024-08-10', 550, 550),

(14, 28, 10, 5, '2024-08-05', 1, '2024-08-12', '2024-08-11', 400, 400),

(15, 29, 1, 1, '2024-09-01', 2, '2024-09-07', '2024-09-06', 350, 350),

(16, 30, 2, 2, '2024-09-02', 3, '2024-09-09', '2024-09-08', 500, 500),

(17, 31, 3, 3, '2024-09-03', 4, '2024-09-10', '2024-09-09', 600, 600),

(18, 32, 4, 4, '2024-09-04', 5, '2024-09-11', '2024-09-10', 1000, 1000),

(19, 33, 5, 5, '2024-09-05', 1, '2024-09-12', '2024-09-11', 550, 550),

(20, 34, 6, 1, '2024-10-01', 2, '2024-10-07', '2024-10-06', 600, 600),

(21, 35, 7, 2, '2024-10-02', 3, '2024-10-09', '2024-10-08', 800, 800),

(22, 36, 8, 3, '2024-10-03', 4, '2024-10-10', '2024-10-09', 1200, 1200),

(23, 37, 9, 4, '2024-10-04', 5, '2024-10-11', '2024-10-10', 550, 550),

(24, 38, 10, 5, '2024-10-05', 1, '2024-10-12', '2024-10-11', 400, 400),

(11, 39, 1, 1, '2024-11-01', 2, '2024-11-07', '2024-11-06', 350, 350),

(12, 40, 2, 2, '2024-11-02', 3, '2024-11-09', '2024-11-08', 500, 500),

(13, 41, 3, 3, '2024-11-03', 4, '2024-11-10', '2024-11-09', 600, 600),

(14, 42, 4, 4, '2024-11-04', 5, '2024-11-11', '2024-11-10', 1000, 1000),

(15, 43, 5, 5, '2024-11-05', 1, '2024-11-20', '2024-11-20', 550, 550),

(16, 44, 6, 1, '2024-12-01', 2, '2024-12-07', '2024-12-06', 600, 600),

(17, 45, 7, 2, '2024-12-02', 3, '2024-12-09', '2024-12-08', 800, 800),

(18, 46, 8, 3, '2024-12-03', 4, '2024-12-10', '2024-12-09', 1200, 1200),

(19, 47, 9, 4, '2024-12-04', 5, '2024-12-11', '2024-12-10', 550, 550),

(20, 48, 10, 5, '2024-12-05', 1, '2024-12-12', '2024-12-11', 400, 400),

(21, 49, 1, 1, '2025-01-01', 2, '2025-01-07', '2025-01-06', 350, 350),

(22, 50, 2, 2, '2025-01-02', 3, '2025-01-09', '2025-01-08', 500, 500),

(23, 51, 3, 3, '2025-01-03', 4, '2025-01-10', '2025-01-09', 600, 600),

(24, 52, 4, 4, '2025-01-04', 5, '2025-01-11', '2025-01-10', 1000, 1000),

(25, 53, 5, 5, '2025-01-05', 1, '2025-01-12', '2025-01-11', 550, 550),

(26, 54, 6, 1, '2025-02-01', 2, '2025-02-07', '2025-02-06', 600, 600),

(27, 55, 7, 2, '2025-02-02', 3, '2025-02-09', '2025-02-08', 800, 800),

(28, 56, 8, 3, '2025-02-03', 4, '2025-02-10', '2025-02-09', 1200, 1200),

(29, 57, 9, 4, '2025-02-04', 5, '2025-02-11', '2025-02-10', 550, 550),

(30, 58, 10, 5, '2025-02-05', 1, '2025-02-12', '2025-02-11', 400, 400),

(1, 59, 1, 1, '2025-03-01', 2, '2025-03-07', '2025-03-06', 350, 350),

(2, 60, 2, 2, '2025-03-02', 3, '2025-03-09', '2025-03-08', 500, 500),

(3, 61, 3, 3, '2025-03-03', 4, '2025-03-10', '2025-03-09', 600, 600),

(4, 62, 4, 4, '2025-03-04', 5, '2025-03-11', '2025-03-10', 1000, 1000),

(5, 63, 5, 5, '2025-03-05', 1, '2025-03-12', '2025-03-11', 550, 550),

(6, 64, 6, 1, '2025-04-01', 2, '2025-04-07', '2025-04-06', 600, 600),

(7, 65, 7, 2, '2025-04-02', 3, '2025-04-09', '2025-04-08', 800, 800),

(8, 66, 8, 3, '2025-04-03', 4, '2025-04-10', '2025-04-09', 1200, 1200),

(9, 67, 9, 4, '2025-04-04', 5, '2025-04-11', '2025-04-10', 550, 550),

(10, 68, 10, 5, '2025-04-05', 1, '2025-04-12', '2025-04-11', 400, 400),

(11, 69, 1, 1, '2025-05-01', 2, '2025-05-07', '2025-05-06', 350, 350),

(12, 70, 2, 2, '2025-05-02', 3, '2025-05-09', '2025-05-08', 500, 500),

(13, 71, 3, 3, '2025-05-03', 4, '2025-05-10', '2025-05-09', 600, 600),

(14, 72, 4, 4, '2025-05-04', 5, '2025-05-11', '2025-05-10', 1000, 1000),

(15, 73, 5, 5, '2025-05-05', 1, '2025-05-12', '2025-05-11', 550, 550),

(16, 74, 6, 1, '2025-06-01', 2, '2025-06-07', '2025-06-06', 600, 600),

(17, 75, 7, 2, '2025-06-02', 3, '2025-06-09', '2025-06-08', 800, 800),

(18, 76, 8, 3, '2025-06-03', 4, '2025-06-10', '2025-06-09', 1200, 1200),

(19, 77, 9, 4, '2025-06-04', 5, '2025-06-11', '2025-06-10', 550, 550),

(20, 78, 10, 5, '2025-06-05', 1, '2025-06-12', '2025-06-11', 400, 400);

**consultas**

| -- 1. Obtener todos los empleados que pertenecen a una sucursal específica.  select \* from empleado where id\_sucursal = 5;  -- 2. Obtener cliente con el correo eduardohernandez@outlook.com  select \* from cliente where correo = 'eduardohernandez@outlook.com';  -- 3. Listar todos los vehículos de un tipo específico.  select v.\* from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where t.tipo = 'Minivan';  -- 4. Obtener el número de clientes por cada sucursal.  select s.ciudad , count(distinct a.id\_cliente) from sucursal s inner join alquileres a on s.id = a.id\_sucursal\_salida group by s.id;  -- 5. Ver todos los alquileres que se han realizado entre dos fechas específicas.  select \* from alquileres where fecha\_salida between '2024-08-01' and '2024-10-05';  -- 6. Listar todos los vehículos que están en uso  select distinct v.\* from alquileres a inner join vehiculo v on a.id\_vehiculo = v.id where current\_date() between a.fecha\_salida and fecha\_esperada\_llegada;  -- 7. Obtener el total de dinero pagado por todos los alquileres  select sum(valor\_pagado) as total\_dinero from alquileres;  -- 8. Ver los detalles de un alquiler específico dado el ID del alquiler.  select \* from alquileres where id = 25;  -- 9. Obtener el número de vehículos por modelo.  select count(\*) as cantidad\_de\_vehiculos , modelo from vehiculo group by modelo;  -- 10. numero de empleados por ciudad.  select count(\*) as cantidad\_empleados , ciudad from empleado group by 2;  -- 11. Obtener el valor total cotizado por cada vehículo  select sum(valor\_cotizado) as total\_valor\_cotizado , id\_vehiculo from alquileres group by id\_vehiculo;  -- 12. Ver los descuentos activos.  select \* from descuento where current\_date() between fecha\_inicio and fecha\_fin;  -- 13. Listar los vehículos que tienen sunroof.  select \* from vehiculo where sunroof = true;  -- 14. Ver los vehículos que tienen más de 4 puertas.  select \* from vehiculo where puertas > 4;  -- 15. Obtener la información de los clientes que han alquilado vehículos en una sucursal específica.  select distinct c.\* , s.\* from alquileres a inner join cliente c on c.id = a.id\_cliente inner join sucursal s on a.id\_sucursal\_salida = s.id where s.id = 4;  -- 16. Obtener los 3 empleados que han registrado más alquileres.  select distinct e.\* , count(a.id) from alquileres a inner join empleado e on e.id = a.id\_empleado group by 1 order by 11 desc limit 3;  -- 17. Listar los vehículos que han tenido más de 3 alquileres.  select v.\* , count(a.id) as cantidad\_de\_alquileres from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by 1 having count(a.id) > 3;  -- 18. entregas con retraso  select \* from alquileres where fecha\_llegada > fecha\_esperada\_llegada ;  -- 19. Ver los clientes que han alquilado un vehículo más de una vez.  select c.\* , count(\*) from alquileres a inner join cliente c on c.id = a.id\_cliente group by 1 having count(\*) > 1;  -- 20. Listar todos los vehículos que han sido alquilados por un cliente específico.  select distinct v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo inner join cliente c on c.id = a.id\_cliente where c.nombre1 = 'Luis';  -- 21. Obtener el tipo de vehiculo con el mayor descuento.  select t.\* , d.porcentaje\_descuento from tipo\_vehiculo t inner join descuento d on t.id = d.id\_tipo order by 5 desc limit 1;  -- 22. Listar los alquileres que no tienen fecha de llegada registrada.  select \* from alquileres where fecha\_llegada is null;  -- 23 cuantos alquileres ha realizado el cliente con identificador 5  select count(a.id\_cliente) as cantidad\_de\_alquileres , c.\* from alquileres a inner join cliente c on a.id\_cliente = c.id where a.id\_cliente = '5' group by 2;  -- 24 cuantas veses ha sido alquilado un vehiculo  select count(v.id) as cantidad\_veses\_alquilado , v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by v.id;  -- 25 nombre del cliente y del empleado que realizaron un alquiler  select a.id as id\_alquiler , c.nombre1 as nombre\_cliente , e.nombre1 as nombre\_empleado from alquileres a inner join cliente c on a.id\_cliente = c.id inner join empleado e on a.id\_empleado = e.id;  -- funcion para ver el valor cotizado real  drop function if exists valor\_cotizado;  delimiter //  create function valor\_cotizado (id\_vehiculo int , fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias float;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  return semanas + dias;  end //  delimiter ;  select id , valor\_cotizado (id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada) as valor\_cotizado\_real from alquileres;  -- dias y semanas que tuvo el vehiculo  drop function if exists dias\_semanas;  delimiter //  create function dias\_semanas (fecha1 date, fecha2 date)  returns varchar(45) deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  return concat('semanas ' , semanas , ' dias ' , dias);  end //  delimiter ;  select id , dias\_semanas(fecha\_salida , fecha\_esperada\_llegada) as semanas\_dias from alquileres;  -- calcular el valor pagado real  drop function if exists valor\_pagado;  delimiter //  create function valor\_pagado (id\_vehiculo int , fecha1 date, fecha2 date , fecha3 date)  returns float deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  declare precio\_retraso float;  declare precio\_descuento float;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set precio\_descuento = (semanas + dias) - (((semanas + dias) \* (select d.porcentaje\_descuento from vehiculo v inner join descuento d on v.id\_tipo = d.id\_tipo where v.id = id\_vehiculo and fecha1 between d.fecha\_inicio and d.fecha\_fin)/100));  if precio\_descuento is null then set precio\_descuento = semanas + dias;  end if;  set precio\_retraso = (fecha3 - fecha2)\*(select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo and fecha3 > fecha2);  if precio\_retraso is null then set precio\_retraso = 0;  else set precio\_retraso = precio\_retraso + (precio\_retraso \* (8/100));  end if;  return precio\_descuento + precio\_retraso;  end //  delimiter ;  select id , valor\_pagado(id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada , fecha\_llegada) as valor\_pagado\_real from alquileres;  -- funcion para ver la cantidad de dias de retraso  drop function if exists dias\_retraso;  delimiter //  create function dias\_retraso (fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias int;  set total\_dias = fecha2 - fecha1;  if total\_dias <= 0 or total\_dias is null then set total\_dias = 0 ;  end if ;  return total\_dias;  end //  delimiter ;  select id , dias\_retraso (fecha\_esperada\_llegada , fecha\_llegada) as dias\_de\_retraso from alquileres;  -- funcion para ver nombres completos  drop function if exists nombre\_completo;  delimiter //  create function nombre\_completo (nombre1 varchar(45) , nombre2 varchar(45) , apellido1 varchar(45) , apellido2 varchar(45))  returns varchar(200) deterministic  begin  return concat(nombre1,' ',nombre2,' ',apellido1,' ',apellido2);  end //  delimiter ;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from cliente;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from empleado; |
| --- |
| -- 1. Obtener todos los empleados que pertenecen a una sucursal específica.  select \* from empleado where id\_sucursal = 5;  -- 2. Obtener cliente con el correo eduardohernandez@outlook.com  select \* from cliente where correo = 'eduardohernandez@outlook.com';  -- 3. Listar todos los vehículos de un tipo específico.  select v.\* from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where t.tipo = 'Minivan';  -- 4. Obtener el número de clientes por cada sucursal.  select s.ciudad , count(distinct a.id\_cliente) from sucursal s inner join alquileres a on s.id = a.id\_sucursal\_salida group by s.id;  -- 5. Ver todos los alquileres que se han realizado entre dos fechas específicas.  select \* from alquileres where fecha\_salida between '2024-08-01' and '2024-10-05';  -- 6. Listar todos los vehículos que están en uso  select distinct v.\* from alquileres a inner join vehiculo v on a.id\_vehiculo = v.id where current\_date() between a.fecha\_salida and fecha\_esperada\_llegada;  -- 7. Obtener el total de dinero pagado por todos los alquileres  select sum(valor\_pagado) as total\_dinero from alquileres;  -- 8. Ver los detalles de un alquiler específico dado el ID del alquiler.  select \* from alquileres where id = 25;  -- 9. Obtener el número de vehículos por modelo.  select count(\*) as cantidad\_de\_vehiculos , modelo from vehiculo group by modelo;  -- 10. numero de empleados por ciudad.  select count(\*) as cantidad\_empleados , ciudad from empleado group by 2;  -- 11. Obtener el valor total cotizado por cada vehículo  select sum(valor\_cotizado) as total\_valor\_cotizado , id\_vehiculo from alquileres group by id\_vehiculo;  -- 12. Ver los descuentos activos.  select \* from descuento where current\_date() between fecha\_inicio and fecha\_fin;  -- 13. Listar los vehículos que tienen sunroof.  select \* from vehiculo where sunroof = true;  -- 14. Ver los vehículos que tienen más de 4 puertas.  select \* from vehiculo where puertas > 4;  -- 15. Obtener la información de los clientes que han alquilado vehículos en una sucursal específica.  select distinct c.\* , s.\* from alquileres a inner join cliente c on c.id = a.id\_cliente inner join sucursal s on a.id\_sucursal\_salida = s.id where s.id = 4;  -- 16. Obtener los 3 empleados que han registrado más alquileres.  select distinct e.\* , count(a.id) from alquileres a inner join empleado e on e.id = a.id\_empleado group by 1 order by 11 desc limit 3;  -- 17. Listar los vehículos que han tenido más de 3 alquileres.  select v.\* , count(a.id) as cantidad\_de\_alquileres from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by 1 having count(a.id) > 3;  -- 18. entregas con retraso  select \* from alquileres where fecha\_llegada > fecha\_esperada\_llegada ;  -- 19. Ver los clientes que han alquilado un vehículo más de una vez.  select c.\* , count(\*) from alquileres a inner join cliente c on c.id = a.id\_cliente group by 1 having count(\*) > 1;  -- 20. Listar todos los vehículos que han sido alquilados por un cliente específico.  select distinct v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo inner join cliente c on c.id = a.id\_cliente where c.nombre1 = 'Luis';  -- 21. Obtener el tipo de vehiculo con el mayor descuento.  select t.\* , d.porcentaje\_descuento from tipo\_vehiculo t inner join descuento d on t.id = d.id\_tipo order by 5 desc limit 1;  -- 22. Listar los alquileres que no tienen fecha de llegada registrada.  select \* from alquileres where fecha\_llegada is null;  -- 23 cuantos alquileres ha realizado el cliente con identificador 5  select count(a.id\_cliente) as cantidad\_de\_alquileres , c.\* from alquileres a inner join cliente c on a.id\_cliente = c.id where a.id\_cliente = '5' group by 2;  -- 24 cuantas veses ha sido alquilado un vehiculo  select count(v.id) as cantidad\_veses\_alquilado , v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by v.id;  -- 25 nombre del cliente y del empleado que realizaron un alquiler  select a.id as id\_alquiler , c.nombre1 as nombre\_cliente , e.nombre1 as nombre\_empleado from alquileres a inner join cliente c on a.id\_cliente = c.id inner join empleado e on a.id\_empleado = e.id;  -- funcion para ver el valor cotizado real  drop function if exists valor\_cotizado;  delimiter //  create function valor\_cotizado (id\_vehiculo int , fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias float;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  return semanas + dias;  end //  delimiter ;  select id , valor\_cotizado (id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada) as valor\_cotizado\_real from alquileres;  -- dias y semanas que tuvo el vehiculo  drop function if exists dias\_semanas;  delimiter //  create function dias\_semanas (fecha1 date, fecha2 date)  returns varchar(45) deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  return concat('semanas ' , semanas , ' dias ' , dias);  end //  delimiter ;  select id , dias\_semanas(fecha\_salida , fecha\_esperada\_llegada) as semanas\_dias from alquileres;  -- calcular el valor pagado real  drop function if exists valor\_pagado;  delimiter //  create function valor\_pagado (id\_vehiculo int , fecha1 date, fecha2 date , fecha3 date)  returns float deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  declare precio\_retraso float;  declare precio\_descuento float;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set precio\_descuento = (semanas + dias) - (((semanas + dias) \* (select d.porcentaje\_descuento from vehiculo v inner join descuento d on v.id\_tipo = d.id\_tipo where v.id = id\_vehiculo and fecha1 between d.fecha\_inicio and d.fecha\_fin)/100));  if precio\_descuento is null then set precio\_descuento = semanas + dias;  end if;  set precio\_retraso = (fecha3 - fecha2)\*(select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo and fecha3 > fecha2);  if precio\_retraso is null then set precio\_retraso = 0;  else set precio\_retraso = precio\_retraso + (precio\_retraso \* (8/100));  end if;  return precio\_descuento + precio\_retraso;  end //  delimiter ;  select id , valor\_pagado(id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada , fecha\_llegada) as valor\_pagado\_real from alquileres;  -- funcion para ver la cantidad de dias de retraso  drop function if exists dias\_retraso;  delimiter //  create function dias\_retraso (fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias int;  set total\_dias = fecha2 - fecha1;  if total\_dias <= 0 or total\_dias is null then set total\_dias = 0 ;  end if ;  return total\_dias;  end //  delimiter ;  select id , dias\_retraso (fecha\_esperada\_llegada , fecha\_llegada) as dias\_de\_retraso from alquileres;  -- funcion para ver nombres completos  drop function if exists nombre\_completo;  delimiter //  create function nombre\_completo (nombre1 varchar(45) , nombre2 varchar(45) , apellido1 varchar(45) , apellido2 varchar(45))  returns varchar(200) deterministic  begin  return concat(nombre1,' ',nombre2,' ',apellido1,' ',apellido2);  end //  delimiter ;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from cliente;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from empleado; |
| -- 1. Obtener todos los empleados que pertenecen a una sucursal específica.  select \* from empleado where id\_sucursal = 5;  -- 2. Obtener cliente con el correo eduardohernandez@outlook.com  select \* from cliente where correo = 'eduardohernandez@outlook.com';  -- 3. Listar todos los vehículos de un tipo específico.  select v.\* from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where t.tipo = 'Minivan';  -- 4. Obtener el número de clientes por cada sucursal.  select s.ciudad , count(distinct a.id\_cliente) from sucursal s inner join alquileres a on s.id = a.id\_sucursal\_salida group by s.id;  -- 5. Ver todos los alquileres que se han realizado entre dos fechas específicas.  select \* from alquileres where fecha\_salida between '2024-08-01' and '2024-10-05';  -- 6. Listar todos los vehículos que están en uso  select distinct v.\* from alquileres a inner join vehiculo v on a.id\_vehiculo = v.id where current\_date() between a.fecha\_salida and fecha\_esperada\_llegada;  -- 7. Obtener el total de dinero pagado por todos los alquileres  select sum(valor\_pagado) as total\_dinero from alquileres;  -- 8. Ver los detalles de un alquiler específico dado el ID del alquiler.  select \* from alquileres where id = 25;  -- 9. Obtener el número de vehículos por modelo.  select count(\*) as cantidad\_de\_vehiculos , modelo from vehiculo group by modelo;  -- 10. numero de empleados por ciudad.  select count(\*) as cantidad\_empleados , ciudad from empleado group by 2;  -- 11. Obtener el valor total cotizado por cada vehículo  select sum(valor\_cotizado) as total\_valor\_cotizado , id\_vehiculo from alquileres group by id\_vehiculo;  -- 12. Ver los descuentos activos.  select \* from descuento where current\_date() between fecha\_inicio and fecha\_fin;  -- 13. Listar los vehículos que tienen sunroof.  select \* from vehiculo where sunroof = true;  -- 14. Ver los vehículos que tienen más de 4 puertas.  select \* from vehiculo where puertas > 4;  -- 15. Obtener la información de los clientes que han alquilado vehículos en una sucursal específica.  select distinct c.\* , s.\* from alquileres a inner join cliente c on c.id = a.id\_cliente inner join sucursal s on a.id\_sucursal\_salida = s.id where s.id = 4;  -- 16. Obtener los 3 empleados que han registrado más alquileres.  select distinct e.\* , count(a.id) from alquileres a inner join empleado e on e.id = a.id\_empleado group by 1 order by 11 desc limit 3;  -- 17. Listar los vehículos que han tenido más de 3 alquileres.  select v.\* , count(a.id) as cantidad\_de\_alquileres from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by 1 having count(a.id) > 3;  -- 18. entregas con retraso  select \* from alquileres where fecha\_llegada > fecha\_esperada\_llegada ;  -- 19. Ver los clientes que han alquilado un vehículo más de una vez.  select c.\* , count(\*) from alquileres a inner join cliente c on c.id = a.id\_cliente group by 1 having count(\*) > 1;  -- 20. Listar todos los vehículos que han sido alquilados por un cliente específico.  select distinct v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo inner join cliente c on c.id = a.id\_cliente where c.nombre1 = 'Luis';  -- 21. Obtener el tipo de vehiculo con el mayor descuento.  select t.\* , d.porcentaje\_descuento from tipo\_vehiculo t inner join descuento d on t.id = d.id\_tipo order by 5 desc limit 1;  -- 22. Listar los alquileres que no tienen fecha de llegada registrada.  select \* from alquileres where fecha\_llegada is null;  -- 23 cuantos alquileres ha realizado el cliente con identificador 5  select count(a.id\_cliente) as cantidad\_de\_alquileres , c.\* from alquileres a inner join cliente c on a.id\_cliente = c.id where a.id\_cliente = '5' group by 2;  -- 24 cuantas veses ha sido alquilado un vehiculo  select count(v.id) as cantidad\_veses\_alquilado , v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by v.id;  -- 25 nombre del cliente y del empleado que realizaron un alquiler  select a.id as id\_alquiler , c.nombre1 as nombre\_cliente , e.nombre1 as nombre\_empleado from alquileres a inner join cliente c on a.id\_cliente = c.id inner join empleado e on a.id\_empleado = e.id;  -- funcion para ver el valor cotizado real  drop function if exists valor\_cotizado;  delimiter //  create function valor\_cotizado (id\_vehiculo int , fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias float;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  return semanas + dias;  end //  delimiter ;  select id , valor\_cotizado (id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada) as valor\_cotizado\_real from alquileres;  -- dias y semanas que tuvo el vehiculo  drop function if exists dias\_semanas;  delimiter //  create function dias\_semanas (fecha1 date, fecha2 date)  returns varchar(45) deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  return concat('semanas ' , semanas , ' dias ' , dias);  end //  delimiter ;  select id , dias\_semanas(fecha\_salida , fecha\_esperada\_llegada) as semanas\_dias from alquileres;  -- calcular el valor pagado real  drop function if exists valor\_pagado;  delimiter //  create function valor\_pagado (id\_vehiculo int , fecha1 date, fecha2 date , fecha3 date)  returns float deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  declare precio\_retraso float;  declare precio\_descuento float;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set precio\_descuento = (semanas + dias) - (((semanas + dias) \* (select d.porcentaje\_descuento from vehiculo v inner join descuento d on v.id\_tipo = d.id\_tipo where v.id = id\_vehiculo and fecha1 between d.fecha\_inicio and d.fecha\_fin)/100));  if precio\_descuento is null then set precio\_descuento = semanas + dias;  end if;  set precio\_retraso = (fecha3 - fecha2)\*(select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo and fecha3 > fecha2);  if precio\_retraso is null then set precio\_retraso = 0;  else set precio\_retraso = precio\_retraso + (precio\_retraso \* (8/100));  end if;  return precio\_descuento + precio\_retraso;  end //  delimiter ;  select id , valor\_pagado(id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada , fecha\_llegada) as valor\_pagado\_real from alquileres;  -- funcion para ver la cantidad de dias de retraso  drop function if exists dias\_retraso;  delimiter //  create function dias\_retraso (fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias int;  set total\_dias = fecha2 - fecha1;  if total\_dias <= 0 or total\_dias is null then set total\_dias = 0 ;  end if ;  return total\_dias;  end //  delimiter ;  select id , dias\_retraso (fecha\_esperada\_llegada , fecha\_llegada) as dias\_de\_retraso from alquileres;  -- funcion para ver nombres completos  drop function if exists nombre\_completo;  delimiter //  create function nombre\_completo (nombre1 varchar(45) , nombre2 varchar(45) , apellido1 varchar(45) , apellido2 varchar(45))  returns varchar(200) deterministic  begin  return concat(nombre1,' ',nombre2,' ',apellido1,' ',apellido2);  end //  delimiter ;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from cliente;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from empleado; |
| -- 1. Obtener todos los empleados que pertenecen a una sucursal específica.  select \* from empleado where id\_sucursal = 5;  -- 2. Obtener cliente con el correo eduardohernandez@outlook.com  select \* from cliente where correo = 'eduardohernandez@outlook.com';  -- 3. Listar todos los vehículos de un tipo específico.  select v.\* from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where t.tipo = 'Minivan';  -- 4. Obtener el número de clientes por cada sucursal.  select s.ciudad , count(distinct a.id\_cliente) from sucursal s inner join alquileres a on s.id = a.id\_sucursal\_salida group by s.id;  -- 5. Ver todos los alquileres que se han realizado entre dos fechas específicas.  select \* from alquileres where fecha\_salida between '2024-08-01' and '2024-10-05';  -- 6. Listar todos los vehículos que están en uso  select distinct v.\* from alquileres a inner join vehiculo v on a.id\_vehiculo = v.id where current\_date() between a.fecha\_salida and fecha\_esperada\_llegada;  -- 7. Obtener el total de dinero pagado por todos los alquileres  select sum(valor\_pagado) as total\_dinero from alquileres;  -- 8. Ver los detalles de un alquiler específico dado el ID del alquiler.  select \* from alquileres where id = 25;  -- 9. Obtener el número de vehículos por modelo.  select count(\*) as cantidad\_de\_vehiculos , modelo from vehiculo group by modelo;  -- 10. numero de empleados por ciudad.  select count(\*) as cantidad\_empleados , ciudad from empleado group by 2;  -- 11. Obtener el valor total cotizado por cada vehículo  select sum(valor\_cotizado) as total\_valor\_cotizado , id\_vehiculo from alquileres group by id\_vehiculo;  -- 12. Ver los descuentos activos.  select \* from descuento where current\_date() between fecha\_inicio and fecha\_fin;  -- 13. Listar los vehículos que tienen sunroof.  select \* from vehiculo where sunroof = true;  -- 14. Ver los vehículos que tienen más de 4 puertas.  select \* from vehiculo where puertas > 4;  -- 15. Obtener la información de los clientes que han alquilado vehículos en una sucursal específica.  select distinct c.\* , s.\* from alquileres a inner join cliente c on c.id = a.id\_cliente inner join sucursal s on a.id\_sucursal\_salida = s.id where s.id = 4;  -- 16. Obtener los 3 empleados que han registrado más alquileres.  select distinct e.\* , count(a.id) from alquileres a inner join empleado e on e.id = a.id\_empleado group by 1 order by 11 desc limit 3;  -- 17. Listar los vehículos que han tenido más de 3 alquileres.  select v.\* , count(a.id) as cantidad\_de\_alquileres from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by 1 having count(a.id) > 3;  -- 18. entregas con retraso  select \* from alquileres where fecha\_llegada > fecha\_esperada\_llegada ;  -- 19. Ver los clientes que han alquilado un vehículo más de una vez.  select c.\* , count(\*) from alquileres a inner join cliente c on c.id = a.id\_cliente group by 1 having count(\*) > 1;  -- 20. Listar todos los vehículos que han sido alquilados por un cliente específico.  select distinct v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo inner join cliente c on c.id = a.id\_cliente where c.nombre1 = 'Luis';  -- 21. Obtener el tipo de vehiculo con el mayor descuento.  select t.\* , d.porcentaje\_descuento from tipo\_vehiculo t inner join descuento d on t.id = d.id\_tipo order by 5 desc limit 1;  -- 22. Listar los alquileres que no tienen fecha de llegada registrada.  select \* from alquileres where fecha\_llegada is null;  -- 23 cuantos alquileres ha realizado el cliente con identificador 5  select count(a.id\_cliente) as cantidad\_de\_alquileres , c.\* from alquileres a inner join cliente c on a.id\_cliente = c.id where a.id\_cliente = '5' group by 2;  -- 24 cuantas veses ha sido alquilado un vehiculo  select count(v.id) as cantidad\_veses\_alquilado , v.\* from alquileres a inner join vehiculo v on v.id = a.id\_vehiculo group by v.id;  -- 25 nombre del cliente y del empleado que realizaron un alquiler  select a.id as id\_alquiler , c.nombre1 as nombre\_cliente , e.nombre1 as nombre\_empleado from alquileres a inner join cliente c on a.id\_cliente = c.id inner join empleado e on a.id\_empleado = e.id;  -- funcion para ver el valor cotizado real  drop function if exists valor\_cotizado;  delimiter //  create function valor\_cotizado (id\_vehiculo int , fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias float;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  return semanas + dias;  end //  delimiter ;  select id , valor\_cotizado (id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada) as valor\_cotizado\_real from alquileres;  -- dias y semanas que tuvo el vehiculo  drop function if exists dias\_semanas;  delimiter //  create function dias\_semanas (fecha1 date, fecha2 date)  returns varchar(45) deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  return concat('semanas ' , semanas , ' dias ' , dias);  end //  delimiter ;  select id , dias\_semanas(fecha\_salida , fecha\_esperada\_llegada) as semanas\_dias from alquileres;  -- calcular el valor pagado real  drop function if exists valor\_pagado;  delimiter //  create function valor\_pagado (id\_vehiculo int , fecha1 date, fecha2 date , fecha3 date)  returns float deterministic  begin  declare total\_dias int;  declare semanas int;  declare dias int;  declare precio\_retraso float;  declare precio\_descuento float;  set total\_dias = fecha2 - fecha1;  set semanas = floor(total\_dias / 7);  set dias = total\_dias % 7;  set semanas = semanas \* (select t.valor\_semana from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set dias = dias \* (select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo);  set precio\_descuento = (semanas + dias) - (((semanas + dias) \* (select d.porcentaje\_descuento from vehiculo v inner join descuento d on v.id\_tipo = d.id\_tipo where v.id = id\_vehiculo and fecha1 between d.fecha\_inicio and d.fecha\_fin)/100));  if precio\_descuento is null then set precio\_descuento = semanas + dias;  end if;  set precio\_retraso = (fecha3 - fecha2)\*(select t.valor\_dia from vehiculo v inner join tipo\_vehiculo t on v.id\_tipo = t.id where v.id = id\_vehiculo and fecha3 > fecha2);  if precio\_retraso is null then set precio\_retraso = 0;  else set precio\_retraso = precio\_retraso + (precio\_retraso \* (8/100));  end if;  return precio\_descuento + precio\_retraso;  end //  delimiter ;  select id , valor\_pagado(id\_vehiculo , fecha\_salida , fecha\_esperada\_llegada , fecha\_llegada) as valor\_pagado\_real from alquileres;  -- funcion para ver la cantidad de dias de retraso  drop function if exists dias\_retraso;  delimiter //  create function dias\_retraso (fecha1 date, fecha2 date)  returns int deterministic  begin  declare total\_dias int;  set total\_dias = fecha2 - fecha1;  if total\_dias <= 0 or total\_dias is null then set total\_dias = 0 ;  end if ;  return total\_dias;  end //  delimiter ;  select id , dias\_retraso (fecha\_esperada\_llegada , fecha\_llegada) as dias\_de\_retraso from alquileres;  -- funcion para ver nombres completos  drop function if exists nombre\_completo;  delimiter //  create function nombre\_completo (nombre1 varchar(45) , nombre2 varchar(45) , apellido1 varchar(45) , apellido2 varchar(45))  returns varchar(200) deterministic  begin  return concat(nombre1,' ',nombre2,' ',apellido1,' ',apellido2);  end //  delimiter ;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from cliente;  select id , nombre\_completo (nombre1 , nombre2 , apellido1 , apellido2) as nombre\_completo from empleado; |

# Referencias