# AutoRental

**Andres Santiago Daza Daza**

**CampusLands**

**Floridablanca**

**Artemis**

**P1**

**2024**

# Tabla de contenidos

[**AutoRental 1**](#_ipvqubrgfus)

[**Tabla de contenidos 2**](#_5tz7y6jtefg8)

[**Introducción 3**](#_qb1kux6e1u61)

[**Caso de Estudio 3**](#_7oazya9mg7ek)

[**Instalación General 4**](#_iaxic0q6cyip)

[**Planificación 5**](#_whm2sf7msylo)

[Ejecución 5](#_1w7a45445684)

[**Construcción del Modelo Conceptual 6**](#_t4jklz1grquh)

[Descripción 6](#_igyl92jqdmuk)

[Gráfica 7](#_fygsljiaaml)

[Descripción Técnica 7](#_udkym2okwzng)

[**Construcción del modelo Lógico 7**](#_j58oi011ddrl)

[Descripción 7](#_ep5p7g8oa1i)

[Gráfica 8](#_57tcqi89ef4m)

[Descripción Técnica 8](#_6ebuxy7te47r)

[**Normalización del modelo Lógico 10**](#_s1w7qsfe4z3f)

[Descripción 10](#_5iq7g2gjuhr0)

[Gráfica 11](#_aff8l7v7jbft)

[Descripción Técnica 11](#_oe8w5tn9qc6v)

[**Construcción del modelo Físico 11**](#_9yaiheso5cs0)

[Descripción 11](#_hi67wdeqw2jn)

[Descripción Técnica 12](#_2bhzzjhscv2b)

[Código 12](#_xkc4uov6afdh)

[**Inserción de datos 14**](#_l1qrp2y1ltf2)

[Descripción 14](#_rf4roeweut8x)

[**Código 15**](#_lkodo6b2ncpi)

[**Funciones y Procedimientos 22**](#_2vpjgg6njuoj)

[Procedimientos 22](#_vutvy2fnxkum)

[Procedimiento para registrar un cliente (registrar\_cliente) 22](#_4cmwqdkdjuwi)

[Procedimiento para actualizar información de un cliente (actualizar\_cliente) 23](#_rbjx7vp8wh47)

[Procedimiento para registrar empleados (registrar\_empleado) 23](#_4sxhxeg8vv3q)

[Procedimiento para actualizar información de un empleado (Actualizar\_empleado) 24](#_bzglipqg9ehz)

[Procedimiento para registrar vehículos (registrar\_vehiculo) 25](#_23nd18eow5i4)

[Procedimiento para actualizar la información de un vehículo (actualizar\_vehiculo) 26](#_30osbz39orn8)

[Procedimiento para insertar sucursales (insertar\_sucursal) 27](#_sjdl6g7o6ft3)

[Procedimiento para actualizar la información de sucursales ya registradas (actualizar\_sucursales) 27](#_tjd3pzawiyf)

[Procedimiento para consultar disponibilidad de vehículos para alquiler por tipo de vehiculo, rango de precios y fechas de disponibilidad (consultar\_vehiculos) 28](#_21hnk1wlpxkx)

[Procedimiento para hacer alquileres (registrar\_alquileres) 29](#_ujteggbzfb7i)

[Funciones 30](#_934fwh4e4728)

[Función para Consultar el historial de alquileres por cliente (histrorial\_alqquileres) 30](#_w531bclpq1xa)

[**creación de triggers y eventos 31**](#_khcccouz6am7)

[Triggers 32](#_lm8010c6ataj)

[after\_cliente\_insert 32](#_ujdaarkszret)

[● Este trigger, está creado con el objetivo de que en el momento en el que se inserte un cliente nuevo en la tabla cliente, se inserta ese nuevo cliente a la tabla inserciones\_nuevas\_clientes, añadiendo el tipo de acción y la fecha y hora en la que se realizó la inserción. 32](#_6i5r6idsydus)

[after\_cliente\_update 33](#_8fow1wyd0ew3)

[● Este trigger, está creado con el objetivo de que en el momento en el que se actualice la información de un cliente en la tabla cliente, se inserta esa actualizacion a la tabla inserciones\_nuevas\_clientes, añadiendo el tipo de acción y la fecha y hora en la que se realizó la inserción. 33](#_5yk4rqezvzzm)

[Eventos 33](#_3fc4iiizyohc)

[daily\_city\_backup 33](#_15bmr0j0k3ky)

[**Creación de usuarios y asignación de permisos 35**](#_n3m2fkzdhd1r)

[Apartado para la creación de usuarios 35](#_yvy77xelkaan)

[Apartado para la asignación de permisos 35](#_frhnyjw35kn)

[Apartado para verificar los permisos de cada usuario 36](#_9u3e158ifxe2)

[**Tabla de usuarios 36**](#_j4wg0gkc6kvo)

# Introducción

Se solicita la creación de una base de datos para la empresa Auto Rental, de manera que se creen 5 tablas para cumplir con este proceso y se desarrollaron los debidos procedimientos y funciones para cumplir con todos estos requerimientos de manera que se mantienen todas las funcionalidades de manera perfecta.

Se dieron los permisos correspondientes a los 3 distintos usuarios, para mantener la debida seguridad

Aquí se muestran las cinco tablas solicitadas para la base de datos

* sucursales
* cliente
* empleado
* vehículos
* alquileres

# 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 Auto Rental, y usted ha sido designado para diseñar una base de datos para ese sistema de información.

Auto Rental 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 Auto Rental podrán alquilar un vehículo en una sucursal y entregarlo en otra sucursal.

Auto Rental 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 cliente 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%.

# 

# 

# 

# Instalación General

Para acceder a la base de datos y a todas sus funcionalidades, lo primero que necesitamos es tener instalado en nuestro equipo MySQL WorkBench o cualquier controlador de base de datos, Se recomienda el uso de WorkBench ya que permite el uso y asegura el funcionamiento, ya que todo su uso ya está probado.

Ahora para acceder a la base de datos, se tiene que seleccionar un usuario y una contraseña, se han creado 3 usuarios, de manera que cada uno de ellos tendrán sus propios permisos totalmente personalizados con anticipación, de manera que ninguno de ellos pueda acceder a permisos que van más allá de su cargo.

# Planificación

## Ejecución

Lo primero y más esencial que se busca, es desarrollar el debido análisis de los requerimientos, de manera que se puedan realizar todas las debidas funcionalidades de la manera más eficiente y eficaz.

Luego del análisis, se desarrolla el modelo conceptual, mediante el cual se presentará un tipo de pre-visualización al cliente de lo que se concluye después del análisis.

Luego del modelo conceptual y con algunas mejoras, se desarrolla el modelo lógico, acompañado de todas las especificaciones de cada uno de los parámetros que se encuentran en las tablas propuestas.

Luego se realizan 3 formas normales, de manera que se forme el modelo más acertado para empezar a construir la base de datos con la menor cantidad de contratiempos posible.

Teniendo como base el resultado de la tercera forma normal, pasamos a la creación del modelo físico, de manera que siguiendo los modelos anteriores, este sea creado de manera satisfactoria.

Una vez desarrollado el modelo físico, se realiza el diagrama E-R, con el objetivo de verificar que todo esté debidamente creado.

Luego, realizamos las inserciones de datos a las tablas, para empezar a desarrollar las funciones y procedimientos.

se crean las funciones y procedimientos requeridos para la base de datos

.

se crean los triggers y eventos necesarios para la base de datos.

Para finalizar se crean los usuarios, a los cuales se les darán los permisos requeridos para que puedan acceder de manera correcta a las funciones que tienen asignadas.

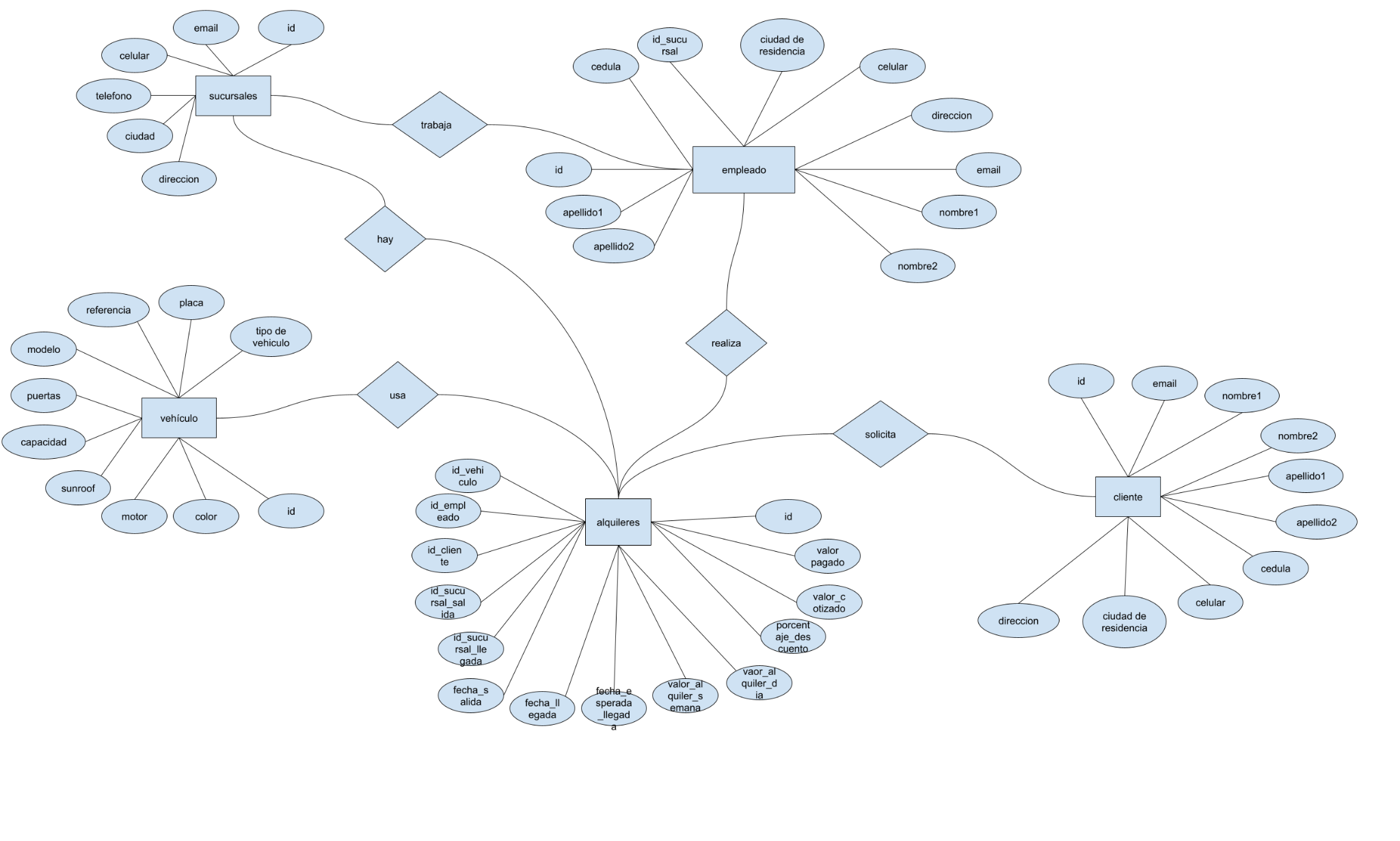
# Construcción del Modelo Conceptual

Para empezar, era necesario crear el modelo conceptual, este fue creado para una visión inicial de la organización que debía tener la base de datos.

## Descripción

Se desarrolló el modelo conceptual mediante dibujos de Google [Conceptual](https://docs.google.com/drawings/d/1KrIpS9VhC5nUV7rXurmWhs3lNo0HQY3mQCjfvSmpiKA/edit?usp=sharing), de manera que se buscaba cumplir con los requerimientos iniciales de las tablas solicitadas y sus relaciones iniciales.

## Gráfica



## Descripción Técnica

Se crearon un total de cinco tablas, representadas por rectángulos, las relaciones están representadas por rombos y los parámetros de cada tablas representan como óvalos.

# Construcción del modelo Lógico

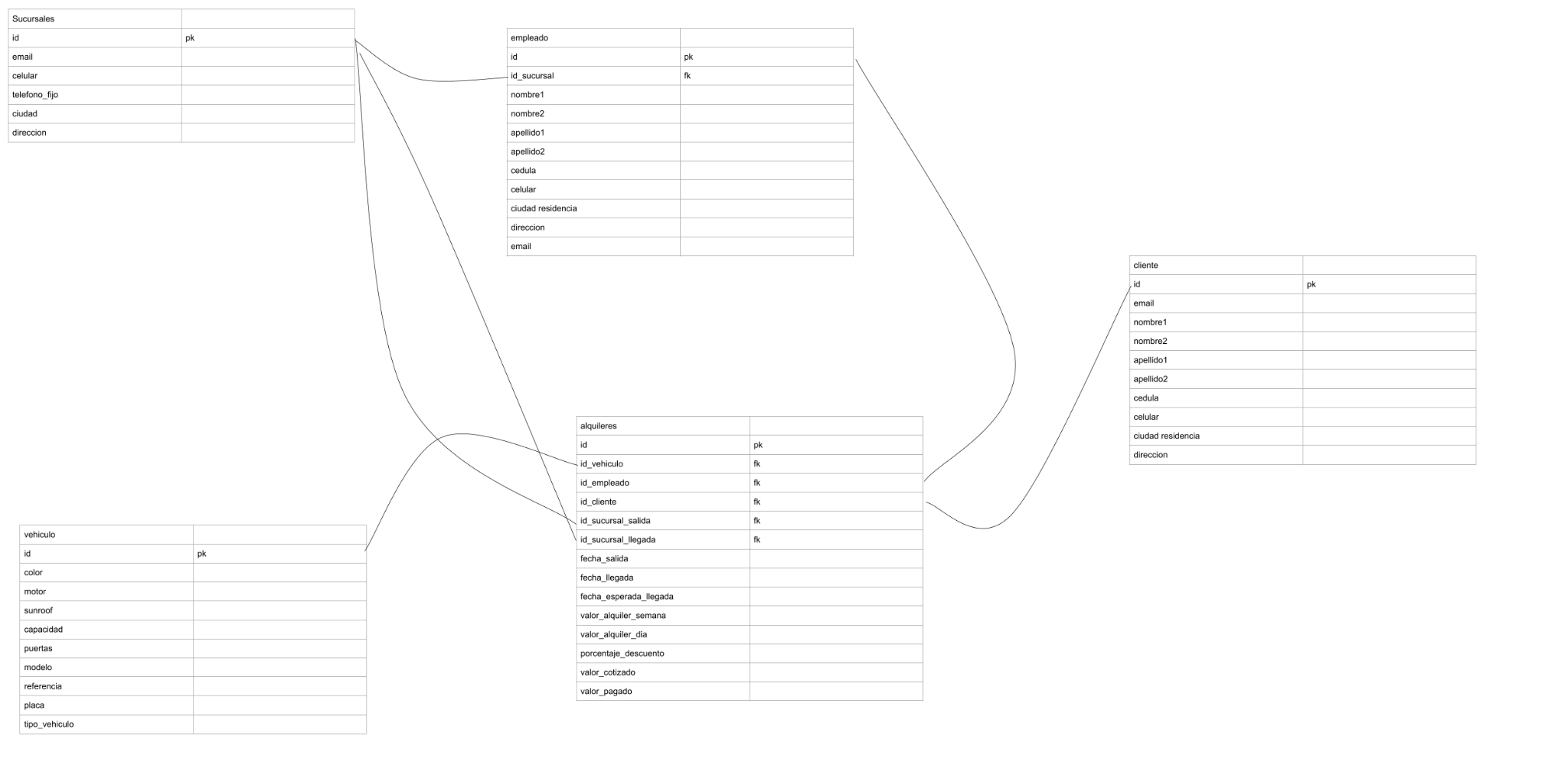
Se realizó la creación del modelo lógico, para dar una mayor base más seguros del funcionamiento deseado para la base de datos.

## Descripción

Se desarrolló el diagrama para el modelo Lógico mediante dibujos de google: [Lógico](https://docs.google.com/drawings/d/1XeKbWXPeej9sMFGR55KWRUv2ax_WI7JVpGb7FSPTGMY/edit?usp=sharing)

buscando que se visualicen de manera más eficiente las tablas con sus parámetros.

## Gráfica



## Descripción Técnica

Se crearon las tablas y las líneas que señalan la procedencia de las llaves foráneas y primarias, a continuación se muestra cada tabla con una información más detallada de cada parámetro:

* cliente {

id int primary key

nombre1 varchar(25)

nombre2 varchar(25)

apellido1 varchar(25)

apellido2 varchar(25)

email varchar(25)

cedula int(15)

celular varchar(25)

ciudad\_resistencia varchar(25)

direccion varchar(25)

}

* sucursales {

id int primary key

email varchar(25)

celular varchar(25)

telefono varchar(25)

ciudad varchar(25)

direccion varchar(25)

}

* empleado {

id int primary key

nombre1 varchar(25)

nombre2 varchar(25)

apellido1 varchar(25)

apellido2 varchar(25)

ciudad\_residencia varchar(25)

celular varchar(25)

direccion varchar(25)

email varchar(25)

cedula int(15)

id\_sucursal int foreign key

}

* vehículo {

id int primary key

color varchar(25)

motor varchar(25)

sunroof varchar(25)

capacidad varchar(25)

puertas int(3)

modelo varchar(25)

referencia varchar(25)

placa varchar(25)

tipo\_vehiculo varchar(25)

}

* alquileres {

id int primary key

id\_vehiculo int foreign key

id\_empleado int foreign key

id\_sucursal\_salida int foreign key

id\_sucursal\_llegada int foreign key

id\_cliente int foreign key

fecha\_salida date

fecha\_llegada date

fecha\_esperada\_llegada date

valor\_alquiler\_semana int(25)

valor\_alquiler\_dia int(25)

porcentaje\_descuento decimal(2,2)

valor\_cotizado int(40)

valor\_pagado int(40)

}

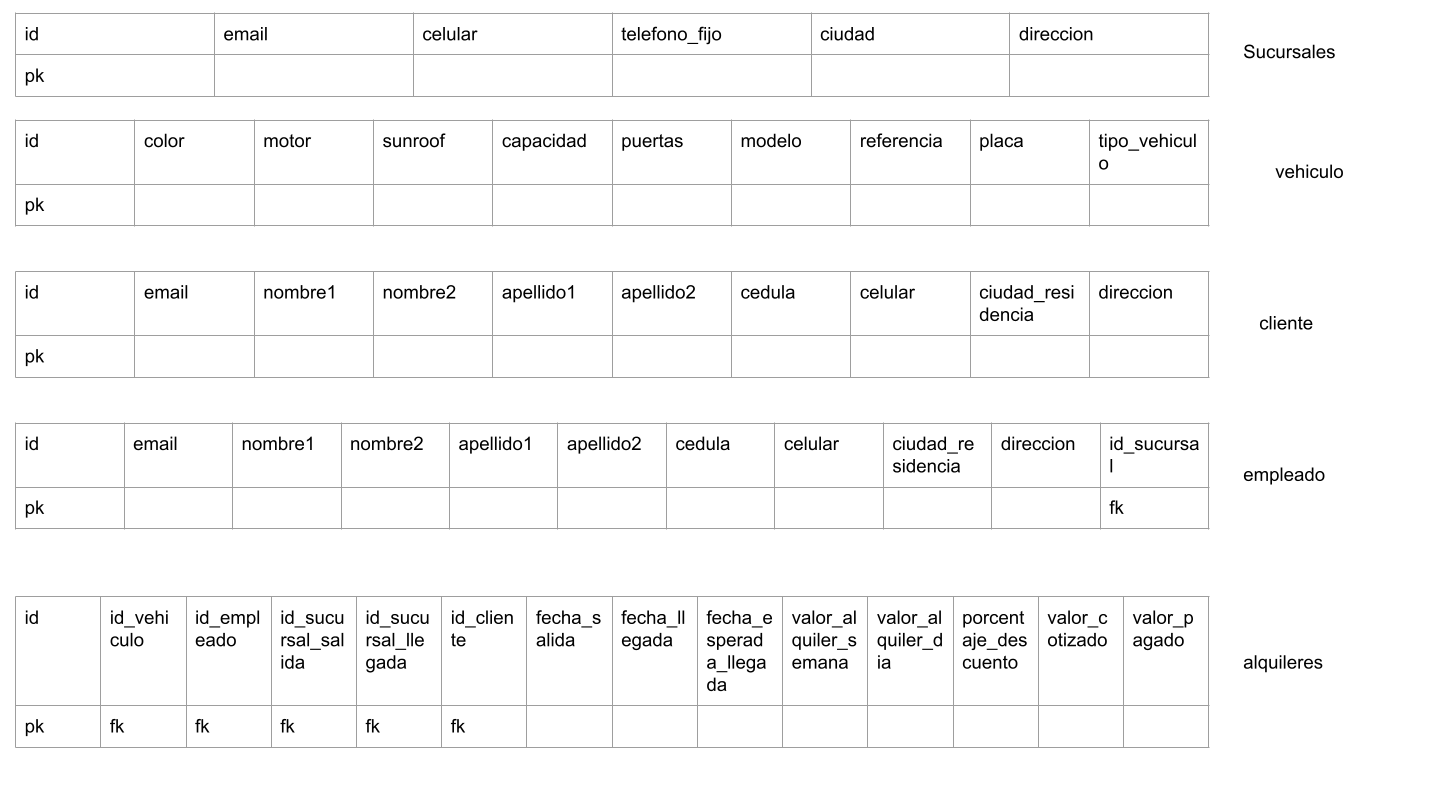
# Normalización del modelo Lógico

Se realizó la normalización hasta su tercera forma, según lo solicitado por el cliente.

## Descripción

Se desarrolló la normalización mediante dibujos de google: [Normalización](https://docs.google.com/drawings/d/1plaLlebTUZknFSs70WY19-vAXL2Koxspv4mFjjHfWhc/edit?usp=sharing) buscando pulir el modelo lógico y permitir leer más fácilmente las tablas y sus parámetros.

## Gráfica



## Descripción Técnica

Se crearon las tablas de manera ordenada y concisa, buscando que las tablas no sean redundantes y que la información cumpla con lo solicitado por el cliente, el objetivo es pasar de estas formas normales al modelo físico sin preocupaciones.

# Construcción del modelo Físico

El objetivo con el modelo físico es crearlo y que no se deban realizar muchas modificaciones en el futuro.

## Descripción

El modelo físico se desarrolló en MySQL, utilizando la creación de tablas y un mínimo de 100 inserts por tabla

## Descripción Técnica

Se crearon un total de 5 tablas, las cuales serán parte de la base de datos “Auto Rental”, estas tablas tienen sus propios parámetros o valores, los cuales las hacen únicas

## Código

create table cliente (

id int primary key,

nombre1 varchar(25) not null,

nombre2 varchar(25),

apellido1 varchar(25) not null,

apellido2 varchar(25),

email varchar(50) not null,

cedula varchar(25) not null,

celular varchar(25) not null,

ciudad\_residencia varchar(25) not null,

direccion varchar(25) not null

);

create table sucursales(

id int primary key,

email varchar(25) not null,

celular varchar(25) not null,

telefono varchar(25) not null,

ciudad varchar(25) not null,

direccion varchar(25) not null

);

create table empleado(

id int primary key,

nombre1 varchar(25) not null,

nombre2 varchar(25),

apellido1 varchar(25) not null,

apellido2 varchar(25),

email varchar(50) not null,

cedula varchar(25) not null,

celular varchar(25) not null,

ciudad\_residencia varchar(25) not null,

direccion varchar(25) not null,

id\_sucursal int,

foreign key (id\_sucursal) references sucursales(id)

);

create table vehiculo(

id int primary key,

color varchar(25) not null,

motor varchar(25) not null,

sunroof varchar(25) not null,

capacidad varchar(25) not null,

puertas int(3) not null,

modelo varchar(25) not null,

referencia varchar(25) not null,

placa varchar(25) not null,

tipo\_vehiculo varchar(25) not null

);

create table alquileres(

id int primary key,

fecha\_salida date not null,

fecha\_llegada date not null,

fecha\_esperada\_llegada date not null,

valor\_alquiler\_semana int(25) not null,

valor\_alquiler\_dia int(25) not null,

porcentaje\_descuento decimal(2,2) not null,

valor\_cotizado int(40) not null,

valor\_pagado int(40) not null,

id\_vehiculo int,

id\_empleado int,

id\_cliente int,

id\_sucursal\_salida int,

id\_sucursal\_llegada int,

foreign key (id\_vehiculo) references vehiculo(id),

foreign key (id\_empleado) references empleado(id),

foreign key (id\_cliente) references cliente(id),

foreign key (id\_sucursal\_salida) references sucursales(id),

foreign key (id\_sucursal\_llegada) references sucursales(id)

);

# Inserción de datos

Se desarrollaron inserts iniciales teniendo en cuenta los parámetros que solicitan cada tabla y sus debidas relaciones

## Descripción

Se realizaron los inserts independientes para cada tabla, con un mínimo de 100 inserciones.

# Código

Aquí, un ejemplo de cómo se insertaron datos en las tablas:

INSERT INTO cliente (id, nombre1, nombre2, apellido1, apellido2, email, cedula, celular, ciudad\_residencia, direccion) VALUES

(1, 'Juan', 'Carlos', 'González', 'Pérez', 'juancarlos1@example.com', 1234567890, '3001234567', 'Bogotá', 'Calle 1 #23-45'),

(2, 'María', 'Elena', 'Ramírez', 'López', 'mariaelena2@example.com', 2345678901, '3002345678', 'Medellín', 'Carrera 2 #34-56'),

(3, 'Pedro', 'Andrés', 'Martínez', 'García', 'pedroandres3@example.com', 3456789012, '3003456789', 'Cali', 'Avenida 3 #45-67'),

(4, 'Ana', 'Isabel', 'Rodríguez', 'Hernández', 'anaisabel4@example.com', 4567890123, '3004567890', 'Barranquilla', 'Diagonal 4 #56-78'),

(5, 'Luis', 'Miguel', 'Hernández', 'Sánchez', 'luismiguel5@example.com', 5678901234, '3005678901', 'Cartagena', 'Transversal 5 #67-89'),

(6, 'Laura', 'Patricia', 'López', 'Gómez', 'laurapatricia6@example.com', 6789012345, '3006789012', 'Bucaramanga', 'Calle 6 #78-90'),

(7, 'Jorge', 'Luis', 'Pérez', 'Torres', 'jorgeluis7@example.com', 7890123456, '3007890123', 'Pereira', 'Carrera 7 #89-01'),

(8, 'Carolina', 'Andrea', 'Gómez', 'Ramírez', 'carolinaandrea8@example.com', 8901234567, '3008901234', 'Manizales', 'Avenida 8 #90-12'),

(9, 'Ricardo', 'José', 'Torres', 'Martínez', 'ricardojose9@example.com', 9012345678, '3009012345', 'Cúcuta', 'Diagonal 9 #01-23'),

(10, 'Catalina', 'Sofía', 'García', 'Rodríguez', 'catalinasofia10@example.com', 1234567891, '3001234568', 'Santa Marta', 'Transversal 10 #12-34'),

(11, 'Andrés', 'Felipe', 'Moreno', 'Rojas', 'andresfelipe11@example.com', 1122334455, '3101234567', 'Ibagué', 'Calle 11 #23-45'),

(12, 'Valentina', 'María', 'Ortiz', 'Guzmán', 'valentinamaria12@example.com', 2233445566, '3102345678', 'Neiva', 'Carrera 12 #34-56'),

(13, 'Santiago', 'David', 'Jiménez', 'Peña', 'santiagodavid13@example.com', 3344556677, '3103456789', 'Popayán', 'Avenida 13 #45-67'),

(14, 'Camila', 'Fernanda', 'Castro', 'Mendoza', 'camilafernanda14@example.com', 4455667788, '3104567890', 'Pasto', 'Diagonal 14 #56-78'),

(15, 'Gabriel', 'Alejandro', 'Suárez', 'Ortiz', 'gabrielalejandro15@example.com', 5566778899, '3105678901', 'Armenia', 'Transversal 15 #67-89'),

(16, 'Natalia', 'Juliana', 'Soto', 'Romero', 'nataliajuliana16@example.com', 6677889900, '3106789012', 'Villavicencio', 'Calle 16 #78-90'),

(17, 'Sebastián', 'José', 'Mejía', 'Ramos', 'sebastianjose17@example.com', 7788990011, '3107890123', 'Montería', 'Carrera 17 #89-01'),

(18, 'Daniela', 'Paola', 'Mora', 'Santos', 'danielapaola18@example.com', 8899001122, '3108901234', 'Tunja', 'Avenida 18 #90-12'),

(19, 'Manuel', 'Fernando', 'Vargas', 'Nieto', 'manuelfernando19@example.com', 9900112233, '3109012345', 'Sincelejo', 'Diagonal 19 #01-23'),

(20, 'Sara', 'Luz', 'Rojas', 'Díaz', 'saraluz20@example.com', 1011122334, '3201234567', 'Florencia', 'Transversal 20 #12-34'),

(21, 'Álvaro', 'Eugenio', 'Pérez', 'Gómez', 'alvaroeugenio21@example.com', 2122233344, '3202345678', 'Yopal', 'Calle 21 #23-45'),

(22, 'Mónica', 'Andrea', 'Martínez', 'Ramírez', 'monicaandrea22@example.com', 3233344455, '3203456789', 'Valledupar', 'Carrera 22 #34-56'),

(23, 'David', 'Enrique', 'Ríos', 'López', 'davidenrique23@example.com', 4344455566, '3204567890', 'Riohacha', 'Avenida 23 #45-67'),

(24, 'Paula', 'Milena', 'Gutiérrez', 'Molina', 'paulamilena24@example.com', 5455566677, '3205678901', 'Quibdó', 'Diagonal 24 #56-78'),

(25, 'Alejandro', 'Antonio', 'Figueroa', 'Reyes', 'alejandroantonio25@example.com', 6566677788, '3206789012', 'San Andrés', 'Transversal 25 #67-89'),

(26, 'Carolina', 'Marcela', 'Ruiz', 'Córdoba', 'carolinamarcela26@example.com', 7677788899, '3207890123', 'Leticia', 'Calle 26 #78-90'),

(27, 'Miguel', 'Ángel', 'Lara', 'Cruz', 'miguelangel27@example.com', 8788899900, '3208901234', 'Mitú', 'Carrera 27 #89-01'),

(28, 'Lucía', 'Cristina', 'Campos', 'Álvarez', 'luciacristina28@example.com', 9899000011, '3209012345', 'Mocoa', 'Avenida 28 #90-12'),

(29, 'Sofía', 'Mariana', 'Rincón', 'Paredes', 'sofiamariana29@example.com', 1001122334, '3301234567', 'Inírida', 'Diagonal 29 #01-23'),

(30, 'Tomás', 'Eduardo', 'Londoño', 'Morales', 'tomaseduardo30@example.com', 1102233445, '3302345678', 'San José del Guaviare', 'Transversal 30 #12-34'),

(31, 'Emilia', 'Valeria', 'Peña', 'Herrera', 'emiliavaleria31@example.com', 1203344556, '3303456789', 'Puerto Carreño', 'Calle 31 #23-45'),

(32, 'Martín', 'Vicente', 'Castaño', 'Ríos', 'martinvicente32@example.com', 1304455667, '3304567890', 'Arauca', 'Carrera 32 #34-56'),

(33, 'Clara', 'Verónica', 'Paredes', 'Medina', 'claraveronica33@example.com', 1405566778, '3305678901', 'Puerto Inírida', 'Avenida 33 #45-67'),

(34, 'Rodrigo', 'Iván', 'Serrano', 'Méndez', 'rodrigoivan34@example.com', 1506677889, '3306789012', 'San José del Guaviare', 'Diagonal 34 #56-78'),

(35, 'Isabel', 'Cristina', 'Reyes', 'García', 'isabelcristina35@example.com', 1607788990, '3307890123', 'Mitú', 'Transversal 35 #67-89'),

(36, 'Samuel', 'Alberto', 'León', 'Lara', 'samuelalberto36@example.com', 1708899901, '3308901234', 'Leticia', 'Calle 36 #78-90'),

(37, 'María', 'José', 'Patiño', 'López', 'mariajose37@example.com', 1809900012, '3309012345', 'San Andrés', 'Carrera 37 #89-01'),

(38, 'Esteban', 'Fernando', 'González', 'Rodríguez', 'estebanfernando38@example.com', 1901122333, '3401234567', 'Quibdó', 'Avenida 38 #90-12'),

(39, 'Marta', 'Patricia', 'Hernández', 'Rojas', 'martapatricia39@example.com', 2002233444, '3402345678', 'Riohacha', 'Diagonal 39 #01-23'),

(40, 'Francisco', 'Antonio', 'Suárez', 'Gómez', 'franciscoantonio40@example.com', 2103344555, '3403456789', 'Valledupar', 'Transversal 40 #12-34'),

(41, 'Laura', 'Catalina', 'Gutiérrez', 'Martínez', 'lauracatalina41@example.com', 2204455666, '3404567890', 'Yopal', 'Calle 41 #23-45'),

(42, 'Carlos', 'David', 'Ortiz', 'Peña', 'carlosdavid42@example.com', 2305566777, '3405678901', 'Florencia', 'Carrera 42 #34-56'),

(43, 'Liliana', 'María', 'Rojas', 'García', 'lilianamaria43@example.com', 2406677888, '3406789012', 'Sincelejo', 'Avenida 43 #45-67'),

(44, 'Germán', 'Andrés', 'Pérez', 'Ramírez', 'germanandres44@example.com', 2507788999, '3407890123', 'Tunja', 'Diagonal 44 #56-78'),

(45, 'Vanessa', 'Paola', 'López', 'Santos', 'vanessapaola45@example.com', 2608899900, '3408901234', 'Montería', 'Transversal 45 #67-89'),

(46, 'Rafael', 'Javier', 'Torres', 'Ramos', 'rafaeljavier46@example.com', 2709900011, '3409012345', 'Villavicencio', 'Calle 46 #78-90'),

(47, 'Marcela', 'Andrea', 'Soto', 'Ortiz', 'marcelaandrea47@example.com', 2801122334, '3501234567', 'Armenia', 'Carrera 47 #89-01'),

(48, 'Felipe', 'Alejandro', 'Vargas', 'Molina', 'felipealejandro48@example.com', 2902233445, '3502345678', 'Pasto', 'Avenida 48 #90-12'),

(49, 'Verónica', 'Isabel', 'Moreno', 'Peña', 'veronicaisabel49@example.com', 3003344556, '3503456789', 'Popayán', 'Diagonal 49 #01-23'),

(50, 'Oscar', 'Luis', 'Ríos', 'Guzmán', 'oscarluis50@example.com', 3104455667, '3504567890', 'Neiva', 'Transversal 50 #12-34'),

(51, 'Carmen', 'Julia', 'Castaño', 'Herrera', 'carmenjulia51@example.com', 3205566778, '3505678901', 'Ibagué', 'Calle 51 #23-45'),

(52, 'Hugo', 'Antonio', 'Guzmán', 'Rodríguez', 'hugoantonio52@example.com', 3306677889, '3506789012', 'Santa Marta', 'Carrera 52 #34-56'),

(53, 'Diana', 'Paola', 'Paredes', 'Martínez', 'dianapaola53@example.com', 3407788990, '3507890123', 'Cúcuta', 'Avenida 53 #45-67'),

(54, 'José', 'Miguel', 'Campos', 'Ramírez', 'josemiguel54@example.com', 3508899901, '3508901234', 'Manizales', 'Diagonal 54 #56-78'),

(55, 'Cecilia', 'María', 'Mora', 'Torres', 'ceciliamaria55@example.com', 3609900012, '3509012345', 'Pereira', 'Transversal 55 #67-89'),

(56, 'Fernando', 'Alberto', 'Vargas', 'Gómez', 'fernandoalberto56@example.com', 3701122334, '3601234567', 'Bucaramanga', 'Calle 56 #78-90'),

(57, 'Patricia', 'Carolina', 'Rojas', 'Sánchez', 'patriciacarolina57@example.com', 3802233445, '3602345678', 'Cartagena', 'Carrera 57 #89-01'),

(58, 'Luis', 'Javier', 'Gutiérrez', 'Hernández', 'luisjavier58@example.com', 3903344556, '3603456789', 'Barranquilla', 'Avenida 58 #90-12'),

(59, 'Estela', 'Beatriz', 'Patiño', 'García', 'estelabeatriz59@example.com', 4004455667, '3604567890', 'Cali', 'Diagonal 59 #01-23'),

(60, 'Mario', 'Eugenio', 'Serrano', 'López', 'marioeugenio60@example.com', 4105566778, '3605678901', 'Medellín', 'Transversal 60 #12-34'),

(61, 'Alicia', 'Patricia', 'Lara', 'Pérez', 'aliciapatricia61@example.com', 4206677889, '3606789012', 'Bogotá', 'Calle 61 #23-45'),

(62, 'Fabián', 'Andrés', 'Campos', 'Rojas', 'fabianandres62@example.com', 4307788990, '3607890123', 'Ibagué', 'Carrera 62 #34-56'),

(63, 'Paola', 'María', 'Ríos', 'López', 'paolamaria63@example.com', 4408899901, '3608901234', 'Neiva', 'Avenida 63 #45-67'),

(64, 'Julián', 'Antonio', 'González', 'Rodríguez', 'julianantonio64@example.com', 4509900012, '3609012345', 'Popayán', 'Diagonal 64 #56-78'),

(65, 'Liliana', 'Patricia', 'Soto', 'Guzmán', 'lilianapatricia65@example.com', 4601122334, '3701234567', 'Pasto', 'Transversal 65 #67-89'),

(66, 'Ramiro', 'Alberto', 'Vargas', 'Herrera', 'ramiroalberto66@example.com', 4702233445, '3702345678', 'Armenia', 'Calle 66 #78-90'),

(67, 'Adriana', 'María', 'Gómez', 'Peña', 'adrianamaria67@example.com', 4803344556, '3703456789', 'Villavicencio', 'Carrera 67 #89-01'),

(68, 'Edwin', 'Andrés', 'Reyes', 'Ortiz', 'edwinandres68@example.com', 4904455667, '3704567890', 'Montería', 'Avenida 68 #90-12'),

(69, 'María', 'Paula', 'Moreno', 'Ramírez', 'mariapaula69@example.com', 5005566778, '3705678901', 'Tunja', 'Diagonal 69 #01-23'),

(70, 'Gonzalo', 'David', 'Rojas', 'Santos', 'gonzalodavid70@example.com', 5106677889, '3706789012', 'Sincelejo', 'Transversal 70 #12-34'),

(71, 'Rosalía', 'Elena', 'Paredes', 'Ramos', 'rosaliaelena71@example.com', 5207788990, '3707890123', 'Florencia', 'Calle 71 #23-45'),

(72, 'Nelson', 'Enrique', 'Gutiérrez', 'Gómez', 'nelsonenrique72@example.com', 5308899901, '3708901234', 'Yopal', 'Carrera 72 #34-56'),

(73, 'Claudia', 'Andrea', 'Campos', 'García', 'claudiaandrea73@example.com', 5409900012, '3709012345', 'Valledupar', 'Avenida 73 #45-67'),

(74, 'Hernán', 'Alfonso', 'Mora', 'Ramírez', 'hernanalfonso74@example.com', 5511122334, '3801234567', 'Riohacha', 'Diagonal 74 #56-78'),

(75, 'Olga', 'Patricia', 'Ríos', 'Guzmán', 'olgarpatricia75@example.com', 5612233445, '3802345678', 'Quibdó', 'Transversal 75 #67-89'),

(76, 'Raúl', 'Antonio', 'González', 'Rodríguez', 'raulantonio76@example.com', 5713344556, '3803456789', 'San Andrés', 'Calle 76 #78-90'),

(77, 'Lucía', 'María', 'Soto', 'Peña', 'luciamaria77@example.com', 5814455667, '3804567890', 'Mitú', 'Carrera 77 #89-01'),

(78, 'Eduardo', 'Javier', 'Vargas', 'García', 'eduardojavier78@example.com', 5915566778, '3805678901', 'Leticia', 'Avenida 78 #90-12'),

(79, 'Rosa', 'Andrea', 'Pérez', 'Ramírez', 'rosaandrea79@example.com', 6016677889, '3806789012', 'San José del Guaviare', 'Diagonal 79 #01-23'),

(80, 'Jorge', 'Luis', 'Castaño', 'López', 'jorgeluis80@example.com', 6117788990, '3807890123', 'Puerto Carreño', 'Transversal 80 #12-34'),

(81, 'Liliana', 'Paola', 'Gutiérrez', 'Pérez', 'lilianapaola81@example.com', 6218899901, '3808901234', 'Arauca', 'Calle 81 #23-45'),

(82, 'Jaime', 'Antonio', 'Ortiz', 'Santos', 'jaimeantonio82@example.com', 6319900012, '3809012345', 'Puerto Inírida', 'Carrera 82 #34-56'),

(83, 'María', 'Fernanda', 'Ríos', 'Molina', 'mariafernanda83@example.com', 6421122334, '3901234567', 'San José del Guaviare', 'Avenida 83 #45-67'),

(84, 'Francisco', 'José', 'Hernández', 'Peña', 'franciscojose84@example.com', 6522233445, '3902345678', 'Mitú', 'Diagonal 84 #56-78'),

(85, 'Gloria', 'Patricia', 'Gómez', 'Córdoba', 'gloriapatricia85@example.com', 6623344556, '3903456789', 'Leticia', 'Transversal 85 #67-89'),

(86, 'Carlos', 'Alberto', 'López', 'Mora', 'carlosalberto86@example.com', 6724455667, '3904567890', 'San Andrés', 'Calle 86 #78-90'),

(87, 'Ana', 'María', 'Campos', 'Rojas', 'anamaría87@example.com', 6825566778, '3905678901', 'Quibdó', 'Carrera 87 #89-01'),

(88, 'Gabriel', 'José', 'Paredes', 'Rodríguez', 'gabrieljose88@example.com', 6926677889, '3906789012', 'Riohacha', 'Avenida 88 #90-12'),

(89, 'Esteban', 'Luis', 'Rojas', 'Ramírez', 'estebanluis89@example.com', 7027788990, '3907890123', 'Valledupar', 'Diagonal 89 #01-23'),

(90, 'Mónica', 'Andrea', 'Suárez', 'Gómez', 'monicaandrea90@example.com', 7128899901, '3908901234', 'Yopal', 'Transversal 90 #12-34'),

(91, 'Luis', 'Miguel', 'González', 'López', 'luismiguel91@example.com', 7229900012, '3909012345', 'Florencia', 'Calle 91 #23-45'),

(92, 'Diana', 'Paola', 'Hernández', 'Peña', 'dianapaola92@example.com', 7331122334, '4001234567', 'Sincelejo', 'Carrera 92 #34-56'),

(93, 'Oscar', 'Eduardo', 'Ríos', 'Guzmán', 'oscareduardo93@example.com', 7432233445, '4002345678', 'Tunja', 'Avenida 93 #45-67'),

(94, 'Beatriz', 'Patricia', 'Campos', 'Ramírez', 'beatrizpatricia94@example.com', 7533344556, '4003456789', 'Montería', 'Diagonal 94 #56-78'),

(95, 'Ramón', 'Antonio', 'Pérez', 'Santos', 'ramonantonio95@example.com', 7634455667, '4004567890', 'Villavicencio', 'Transversal 95 #67-89'),

(96, 'Paula', 'Andrea', 'Gutiérrez', 'García', 'paulaandrea96@example.com', 7735566778, '4005678901', 'Armenia', 'Calle 96 #78-90'),

(97, 'Mario', 'David', 'Ortiz', 'Peña', 'mariodavid97@example.com', 7836677889, '4006789012', 'Pasto', 'Carrera 97 #89-01'),

(98, 'Lucía', 'Isabel', 'Ríos', 'Rodríguez', 'luciaisabel98@example.com', 7937788990, '4007890123', 'Popayán', 'Avenida 98 #90-12'),

(99, 'Raúl', 'Enrique', 'González', 'Guzmán', 'raulenrique99@example.com', 8038899901, '4008901234', 'Neiva', 'Diagonal 99 #01-23'),

(100, 'Elena', 'Patricia', 'Pérez', 'López', 'elenapatricia100@example.com', 8139900012, '4009012345', 'Ibagué', 'Transversal 100 #12-34');

# Funciones y Procedimientos

Se desarrollaron distintos procedimientos y funciones, teniendo como objetivo asignarlas a distintos usuarios para su implementación.

## Procedimientos

### **Procedimiento para registrar un cliente (**registrar\_cliente**)**

* El primer procedimiento creado en el archivo MySQL\_Dia4\_Pro\_Fun.sql, sirve para insertar un cliente a la tabla de clientes, este procedimiento fue creado para permitirle al futuro usuario “empleado” el registro de nuevos clientes, este procedimiento recibe como parámetros, todos los datos del cliente que se desea registrar.

delimiter //

create procedure registrar\_cliente(cliente\_id int, cliente\_nombre1 varchar(25), cliente\_nombre2 varchar(25), cliente\_apellido1 varchar(25), cliente\_apellido2 varchar(25), cliente\_email varchar(50), cliente\_cedula varchar(25), cliente\_celular varchar(25), cliente\_ciudad\_residencia varchar(25), cliente\_direccion varchar(25))

begin

insert into cliente(id, nombre1, nombre2, apellido1, apellido2, email, cedula, celular, ciudad\_residencia, direccion)

values (cliente\_id, cliente\_nombre1, cliente\_nombre2, cliente\_apellido1, cliente\_apellido2, cliente\_email, cliente\_cedula, cliente\_celular, cliente\_ciudad\_residencia, cliente\_direccion);

end//

delimiter ;

* Ejemplo de uso

CALL registrar\_cliente('101', 'Alberto', 'Nicol', 'instantaneo', 'Apesta', 'AlbertEinstein@gmail.com', '1234567890', '123-123-123', 'Bucaramanga', 'Rincon de Giron 100#123-123');

### **Procedimiento para actualizar información de un cliente (**actualizar\_cliente**)**

* Este procedimiento, sirve para actualizar la información de un cliente ya registrado en la tabla clientes, este procedimiento fue creado para el futuro usuario “empleado”, de manera que este pueda actualizar la información, ingresando como primer parámetro el id del cliente que desea actualizar, para después añadir los demás datos.

**delimiter //**

**create procedure actualizar\_cliente(id\_cliente int, n\_id\_cliente int, n\_cliente\_nombre1 varchar(25), n\_cliente\_nombre2 varchar(25), n\_cliente\_apellido1 varchar(25), n\_cliente\_apellido2 varchar(25), n\_cliente\_email varchar(50), n\_cliente\_cedula varchar(25), n\_cliente\_celular varchar(25), n\_cliente\_ciudad\_residencia varchar(25), n\_cliente\_direccion varchar(25))**

**begin**

**update cliente set id = n\_id\_cliente, nombre1 = n\_cliente\_nombre1, nombre2 = n\_cliente\_nombre2, apellido1 = n\_cliente\_apellido1, apellido2 = n\_cliente\_apellido2, email = n\_cliente\_email, cedula = n\_cliente\_cedula, celular = n\_cliente\_celular, ciudad\_residencia = n\_cliente\_ciudad\_residencia, direccion = n\_cliente\_direccion**

**where id = id\_cliente;**

**end//**

**delimiter ;**

* ejemplo de uso

call actualizar\_cliente(12345, 102, 'Juancho','Felipe','Casillas','Vida','JuanchoVida@gmail.com','6333333-3', '633-33-33', 'Piedecuesta', 'barrio 232#232-232');

### **Procedimiento para registrar empleados (**registrar\_empleado**)**

* este procedimiento, sirve para permitirle al futuro usuario “Director”, añadir nuevos empleados a la tabla de empleados, este usuario está pensado para que sea algo así como un “superior” de los demás empleados, por ende tiene los permisos para agregar empleados nuevos, este procedimiento, recibe como parámetros, toda la información del nuevo empleado, incluida la sucursal en la que estará asignado, se pensó, que se podría llegar a equivocar al momento de asignar la sucursal, así que se maneja el caso de que se coloque un id de sucursal inexistente, mostrando un mensaje de error.

**delimiter //**

**create procedure registrar\_empleado(empleado\_id int, empleado\_nombre1 varchar(25), empleado\_nombre2 varchar(25), empleado\_apellido1 varchar(25), empleado\_apellido2 varchar(25), empleado\_email varchar(25), empleado\_cedula varchar(25), empleado\_celular varchar(25), empleado\_ciudad\_residencia varchar(25), empleado\_direccion varchar(25), empleado\_sucursal int)**

**begin**

**declare n\_sucursales int;**

**select count(\*) into n\_sucursales from sucursales where sucursales.id = empleado\_sucursal;**

**if n\_sucursales > 0 then**

**insert into empleado(id, nombre1, nombre2, apellido1, apellido2, email, cedula, celular, ciudad\_residencia, direccion, id\_sucursal) values(empleado\_id, empleado\_nombre1, empleado\_nombre2, empleado\_apellido1, empleado\_apellido2, empleado\_email, empleado\_cedula, empleado\_celular, empleado\_ciudad\_residencia, empleado\_direccion, empleado\_sucursal);**

**else**

**select 'error: la sucursal no existe bro...' as mensaje;**

**end if;**

**end//**

**delimiter ;**

* ejemplo de uso

call registrar\_empleado(102, 'Cristiano', 'nose', 'Ronaldo', 'nose', 'elBichoSiuu@gmail.com', '7777777777', '123432177', 'niidea', 'sisas', 100);

### **Procedimiento para actualizar información de un empleado (**Actualizar\_empleado**)**

* Este procedimiento sirve para actualizar la información de un empleado ya existente, recibiendo como primer parámetro el id del empleado que se busca actualizar, de manera que los demás parámetros, son el resto de la información del empleado en cuestión, este procedimiento está pensado para ser utilizado por el futuro usuario “director”.

**delimiter //**

**create procedure actualizar\_empleado(empleado\_id int, n\_empleado\_id int, empleado\_nombre1 varchar(25), empleado\_nombre2 varchar(25), empleado\_apellido1 varchar(25), empleado\_apellido2 varchar(25), empleado\_email varchar(25), empleado\_cedula varchar(25), empleado\_celular varchar(25), empleado\_ciudad\_residencia varchar(25), empleado\_direccion varchar(25), empleado\_sucursal int)**

**begin**

**update empleado set id = n\_empleado\_id, nombre1 = empleado\_nombre1, nombre2 = empleado\_nombre2, apellido1 = empleado\_apellido1, apellido2 = empleado\_apellido2, email = empleado\_email, cedula = empleado\_cedula, celular = empleado\_celular, ciudad\_residencia = empleado\_ciudad\_residencia, direccion = empleado\_direccion, id\_sucursal = empleado\_sucursal**

**where id = empleado\_id;**

**end//**

**delimiter ;**

* ejemplo de uso

call actualizar\_empleado(103, 102, 'Cristiano', null,'Ronaldo', 'siuu', 'elBichoSiuu@gmail.com', '7777777777', '123432177', 'niidea', 'sisas', 99);

### **Procedimiento para registrar vehículos (**registrar\_vehiculo**)**

* Este procedimiento, sirve para registrar un nuevo vehículo en la tabla vehículo, recibe como parámetros la información del nuevo vehículo y está pensado para ser utilizado por el futuro usuario “empleado”, además, este procedimiento, tiene la funcionalidad para evitar que se registren vehículos los cuales su tipo de vehículo no haya sido ingresado nunca antes en la tabla de vehículos, y mostrará un mensaje de error en el caso de que se trate de ingresar un vehículo con el tipo de vehículo incorrecto.

**delimiter //**

**create procedure registrar\_vehiculo(vehiculo\_id int, vehiculo\_color varchar(25), vehiculo\_motor varchar(25), vehiculo\_sunroof varchar(25), vehiculo\_capacidad varchar(25), vehiculo\_puertas int, vehiculo\_modelo varchar(25), vehiculo\_referencia varchar(25), vehiculo\_placa varchar(25), vehiculo\_tipo\_vehiculo varchar(25))**

**begin**

**declare tipo\_vehiculo\_e int;**

**select count(\*) into tipo\_vehiculo\_e from vehiculo where tipo\_vehiculo = vehiculo\_tipo\_vehiculo;**

**if tipo\_vehiculo\_e > 0 then**

**insert into vehiculo(id, color, motor, sunroof, capacidad, puertas, modelo, referencia, placa, tipo\_vehiculo)**

**values(vehiculo\_id, vehiculo\_color, vehiculo\_motor, vehiculo\_sunroof, vehiculo\_capacidad, vehiculo\_puertas, vehiculo\_modelo, vehiculo\_referencia, vehiculo\_placa, vehiculo\_tipo\_vehiculo);**

**select 'Vehículo registrado exitosamente.' as mensaje;**

**else**

**select 'Error: El tipo de vehículo especificado no es válido.' as mensaje;**

**end if;**

**end//**

**delimiter ;**

* ejemplo de uso:

call registrar\_vehiculo(101, 'Rojo', 'Gasolina', 'No', '5 personas', 4, 'Sedán', '2023', 'ABC123', 'Automóvil');

### **Procedimiento para actualizar la información de un vehículo (**actualizar\_vehiculo**)**

* Este procedimiento, sirve para actualizar la información de los vehículos ya registrados en la tabla vehículos, de manera que recibe como primer parámetro el id del vehículo que se busca actualizar, este procedimiento está pensado para el futuro usuario “empleado”.

**delimiter //**

**create procedure actualizar\_vehiculo( vehiculo\_id int, nuevo\_vehiculo\_id int, vehiculo\_color varchar(25), vehiculo\_motor varchar(25), vehiculo\_sunroof varchar(25), vehiculo\_capacidad varchar(25), vehiculo\_puertas int, vehiculo\_modelo varchar(25), vehiculo\_referencia varchar(25), vehiculo\_placa varchar(25), vehiculo\_tipo\_vehiculo varchar(25))**

**begin**

**update vehiculo set id = nuevo\_vehiculo\_id, color = vehiculo\_color, motor = vehiculo\_motor, sunroof = vehiculo\_sunroof, capacidad = vehiculo\_capacidad, puertas = vehiculo\_puertas, modelo = vehiculo\_modelo, referencia = vehiculo\_referencia, placa = vehiculo\_placa, tipo\_vehiculo = vehiculo\_tipo\_vehiculo**

**where id = vehiculo\_id;**

**select 'Vehículo actualizado correctamente.' as mensaje;**

**end//**

**delimiter ;**

* ejemplo de uso

call actualizar\_vehiculo(101, 102, 'Rojo', 'Gasolina', 'No', '5 personas', 4, 'Sedán', '2023', 'ABC123', 'Automóvil');

### **Procedimiento para insertar sucursales (**insertar\_sucursal**)**

* Este procedimiento, sirve para permitir al futuro usuario “empleado”, insertar nuevas sucursales en la tabla de sucursales, este procedimiento, recibe como parámetros toda la información de la nueva sucursal.

**delimiter //**

**create procedure insertar\_sucursal(sucursal\_id int, sucursal\_email varchar(25), sucursal\_celular varchar(25), sucursal\_telefono varchar(25), sucursal\_ciudad varchar(25), sucursal\_direccion varchar(25))**

**begin**

**insert into sucursales(id, email, celular, telefono, ciudad, direccion) values(sucursal\_id, sucursal\_email, sucursal\_celular, sucursal\_telefono, sucursal\_ciudad, sucursal\_direccion);**

**end//**

**delimiter ;**

### **Procedimiento para actualizar la información de sucursales ya registradas (**actualizar\_sucursales**)**

* Este procedimiento, sirve para permitir al futuro usuario “empleado”, actualizar la información de las sucursales ya existentes, ingresando como primer parámetro el id de la sucursal en cuestión que se desea actualizar.

**delimiter //**

**create procedure actualizar\_sucursal( sucursal\_id int, n\_sucursal\_id int, sucursal\_email varchar(25), sucursal\_celular varchar(25), sucursal\_telefono varchar(25), sucursal\_ciudad varchar(25), sucursal\_direccion varchar(25))**

**begin**

**update sucursales set id = n\_sucursal\_id, email = sucursal\_email, celular = sucursal\_celular, telefono = sucursal\_telefono, ciudad = sucursal\_ciudad, direccion = sucursal\_direccion**

**where id = sucursal\_id;**

**end//**

**delimiter ;**

### **Procedimiento para consultar disponibilidad de vehículos para alquiler por tipo de vehiculo, rango de precios y fechas de disponibilidad (**consultar\_vehiculos**)**

* Este procedimiento sirve para consultar los vehículos que están disponibles para alquiler, teniendo en cuenta su tipo de vehículo, el rango de los precios y las fechas de disponibilidad. El procedimiento recibe como parámetros, El tipo de vehículo, el rango de las fechas que se solicita el vehículo, el rango de precio minimo y maximo por dia y el rango de precio minimo y maximo por semana.
* Al recibir estos parámetros, el procedimiento lo primero que busca es seleccionar los datos de los vehículos, para luego hacer uso de un left join con la tabla alquileres para ingresar el rango de precios y verificar que el vehículo no esté siendo ya utilizado en algún otro alquiler, luego verifica y filtra los resultados, por el tipo de vehículo solicitado, y por las fechas solicitadas.

**delimiter //**

**create procedure consultar\_vehiculos(con\_tipo\_vehiculo varchar(25), con\_fecha\_inicio date, con\_fecha\_fin date, con\_precio\_min\_dia int, con\_precio\_max\_dia int, con\_precio\_min\_semana int, con\_precio\_max\_semana int)**

**begin**

**select vehiculo.id as VehiculoID, vehiculo.color, vehiculo.motor, vehiculo.sunroof, vehiculo.capacidad, vehiculo.puertas, vehiculo.modelo, vehiculo.referencia, vehiculo.placa, vehiculo.tipo\_vehiculo, alquileres.valor\_alquiler\_semana, alquileres.valor\_alquiler\_dia**

**from vehiculo**

**left join alquileres on vehiculo.id = alquileres.id\_vehiculo**

**where vehiculo.tipo\_vehiculo = con\_tipo\_vehiculo and (alquileres.id is null or (alquileres.fecha\_salida > con\_fecha\_fin or alquileres.fecha\_llegada < con\_fecha\_inicio)) and ((ifnull(alquileres.valor\_alquiler\_dia, 0) between con\_precio\_min\_dia and con\_precio\_max\_dia) or (ifnull(alquileres.valor\_alquiler\_semana, 0) between con\_precio\_min\_semana and con\_precio\_max\_semana))**

**order by vehiculo.id;**

**end//**

**delimiter ;**

* ejemplo de uso

call consultar\_vehiculos('Automóvil', '2024-10-01', '2024-10-10', 0, 0, 500000, 550000);

### **Procedimiento para hacer alquileres (**registrar\_alquileres**)**

* Este procedimiento lo que busca, es permitirle al futuro usuario “cliente”, hacer un alquiler, recibiendo todos los datos del alquiler como parámetro, luego, verifica la existencia de los vehículos, de los empleados, del cliente, de la sucursal de salida y de la sucursal de llegada, de manera que en caso de que alguna de estas no esté presente, devolverá un mensaje de error, en caso de que si existan, realizará la inserción con total normalidad.

**delimiter //**

**create procedure registrar\_alquiler(alquiler\_id int, fecha\_salida date, fecha\_llegada date, fecha\_esperada\_llegada date, valor\_alquiler\_semana int, valor\_alquiler\_dia int, porcentaje\_descuento decimal(5,2), valor\_cotizado int, valor\_pagado int, vehiculo\_id int, empleado\_id int, cliente\_id int, sucursal\_salida\_id int, sucursal\_llegada\_id int)**

**begin**

**declare existencia\_vehiculo int;**

**declare existencia\_empleado int;**

**declare existencia\_cliente int;**

**declare existencia\_sucursal\_salida int;**

**declare existencia\_sucursal\_llegada int;**

**select count(\*) into existencia\_vehiculo from vehiculo where id = vehiculo\_id;**

**select count(\*) into existencia\_empleado from empleado where id = empleado\_id;**

**select count(\*) into existencia\_cliente from cliente where id = cliente\_id;**

**select count(\*) into existencia\_sucursal\_salida from sucursales where id = sucursal\_salida\_id;**

**select count(\*) into existencia\_sucursal\_llegada from sucursales where id = sucursal\_llegada\_id;**

**if existencia\_vehiculo > 0 and existencia\_empleado > 0 and existencia\_cliente > 0 and existencia\_sucursal\_salida > 0 and existencia\_sucursal\_llegada > 0 then**

**insert into alquileres(id, fecha\_salida, fecha\_llegada, fecha\_esperada\_llegada, valor\_alquiler\_semana, valor\_alquiler\_dia, porcentaje\_descuento, valor\_cotizado, valor\_pagado, id\_vehiculo, id\_empleado, id\_cliente, id\_sucursal\_salida, id\_sucursal\_llegada)**

**values(alquiler\_id, fecha\_salida, fecha\_llegada, fecha\_esperada\_llegada, valor\_alquiler\_semana, valor\_alquiler\_dia, porcentaje\_descuento, valor\_cotizado, valor\_pagado, vehiculo\_id, empleado\_id, cliente\_id, sucursal\_salida\_id, sucursal\_llegada\_id);**

**else**

**select 'Error: Verifique la existencia de los elementos requeridos (vehículo, empleado, cliente, sucursal de salida y llegada).' as mensaje;**

**end if;**

**end//**

**delimiter ;**

## Funciones

### **Función para Consultar el historial de alquileres por cliente (**histrorial\_alqquileres**)**

* Esta función, tiene como objetivo mostrarle al futuro usuario “ciente”, todos los alquileres que ha realizado los clientes con x id.
* lo que hace, es recibir el id del cliente deseado como parámetro, para mostrar todos los resultados asociados a dicho cliente.

delimiter //

create function historial\_alquileres(cliente\_id INT)

returns text deterministic

begin

declare resultado TEXT;

select

group\_concat(

concat('alquiler id: ', alq.id,

', fecha salida: ', alq.fecha\_salida,

', fecha llegada: ', alq.fecha\_llegada,

', vehículo: ', veh.placa,

', empleado: ', emp.nombre1,

', valor cotizado: ', alq.valor\_cotizado

) separator '\n') into resultado

from alquileres alq

join vehiculo veh on alq.id\_vehiculo = veh.id

join empleado emp on alq.id\_empleado = emp.id

where alq.id\_cliente = cliente\_id;

if resultado is null then

set resultado = 'no se encontraron alquileres para este cliente.';

end if;

return resultado;

end //

delimiter ;

* ejemplo de uso

select historial\_alquileres(1) as Historial;

# creación de triggers y eventos

Se crearon un trigger y un evento, after\_cliente\_insert y daily\_city\_backup

## Triggers

### **after\_cliente\_insert**

### Este trigger, está creado con el objetivo de que en el momento en el que se inserte un cliente nuevo en la tabla cliente, se inserta ese nuevo cliente a la tabla inserciones\_nuevas\_clientes, añadiendo el tipo de acción y la fecha y hora en la que se realizó la inserción.

delimiter //

create trigger after\_cliente\_insert

after insert on cliente

for each row

begin

insert into inserciones\_nuevas\_clientes(id, nombre1, nombre2, apellido1, apellido2, email, cedula, celular, ciudad\_residencia, direccion, action) values(new.id, new.nombre1, new.nombre2, new.apellido1, new.apellido2, new.email, new.cedula, new.celular, new.ciudad\_residencia, new.direccion, "INSERT");

end//

delimiter ;

* Se espera que en el momento en el que se realice un insert en la tabla cliente, se accione inmediatamente este trigger.

### after\_cliente\_update

### Este trigger, está creado con el objetivo de que en el momento en el que se actualice la información de un cliente en la tabla cliente, se inserta esa actualizacion a la tabla inserciones\_nuevas\_clientes, añadiendo el tipo de acción y la fecha y hora en la que se realizó la inserción.

delimiter //

create trigger after\_cliente\_update

after insert on cliente

for each row

begin

insert into inserciones\_nuevas\_clientes(id, nombre1, nombre2, apellido1, apellido2, email, cedula, celular, ciudad\_residencia, direccion, action) values(new.id, new.nombre1, new.nombre2, new.apellido1, new.apellido2, new.email, new.cedula, new.celular, new.ciudad\_residencia, new.direccion, "UPDATE");

end//

delimiter ;

* Se espera que en el momento en el que se realice un update en la tabla cliente, se accione inmediatamente este trigger.

## Eventos

### daily\_city\_backup

* Este evento busca crear una backup diaria de los clientes o de los datos registrados en la tabla de cliente, en la tabla backup\_clientes. de manera que diariamente, se realizará automáticamente esta back up.

create event daily\_city\_backup

on schedule every 1 day

do

begin

truncate table backup\_clientes;

insert into backup\_clientes(id, nombre1, nombre2, apellido1, apellido2, email, cedula, celular, ciudad\_residencia, direccion) select id, nombre1, nombre2, apellido1, apellido2, email, cedula, celular, ciudad\_residencia, direccion

from cliente;

end//

delimiter ;

* Se espera que diariamente se desarrollé la back up, eliminando la backup anterior, de manera que limite el consumo y este no sea tan extenso.

# Creación de usuarios y asignación de permisos

Se crearon un total de 3 usuarios, “cliente”, “empleado”, “director”; cada uno con sus permisos exclusivos sobre las funciones, correspondientes, estos permisos permiten mantener una seguridad aceptable a la base de datos, ya que ninguno de ellos tiene permisos para eliminar ningún dato de la base de datos.

## Apartado para la creación de usuarios

**create user 'director'@'localhost' identified by 'contraseña\_director';**

**create user 'cliente'@'localhost' identified by 'contraseña\_cliente';**

create user 'empleado'@'localhost' identified by 'contraseña\_empleado';

## Apartado para la asignación de permisos

grant execute on function AutoRental.historial\_alquileres to 'cliente'@'localhost';

grant execute on procedure AutoRental.registrar\_alquiler to 'cliente'@'localhost';

grant execute on procedure AutoRental.consultar\_vehiculos to 'cliente'@'localhost';

grant execute on procedure AutoRental.insertar\_sucursal to 'empleado'@'localhost';

grant execute on procedure AutoRental.actualizar\_sucursal to 'empleado'@'localhost';

grant execute on procedure AutoRental.registrar\_vehiculo to 'empleado'@'localhost';

grant execute on procedure AutoRental.actualizar\_vehiculo to 'empleado'@'localhost';

grant execute on procedure AutoRental.registrar\_cliente to 'empleado'@'localhost';

grant execute on procedure AutoRental.actualizar\_cliente to 'empleado'@'localhost';

grant execute on procedure AutoRental.registrar\_empleado to 'director'@'localhost';

grant execute on procedure AutoRental.actualizar\_empleado to 'director'@'localhost';

## Apartado para verificar los permisos de cada usuario

SHOW GRANTS FOR 'cliente'@'localhost';

SHOW GRANTS FOR 'empleado'@'localhost';

SHOW GRANTS FOR 'director'@'localhost';

# Tabla de usuarios

| **Usuario** | **Acceso** | **Descripción** | **Comandos** | **Funciones / Procedimientos** |
| --- | --- | --- | --- | --- |
| **Director** | **%** | **Usuario creado para el manejo de los empleados dentro de la empresa.** | **update, insert, call.** | **registrar\_empleado()**  **actualizar\_empleado()** |
| **Empleado** | **%** | **Usuario creado para que maneje distintas actividades dentro de la empresa y que tenga únicamente los permisos para ello.** | **update, insert, call.** | **insertar\_sucursal()**  **actualizar\_sucursal()**  **registrar\_vehiculo()**  **actualizar\_vehiculo()**  **registrar\_cliente()**  **actualizar\_cliente()** |
| **Cliente** | **%** | **Usuario creado para que los clientes puedan realizar alquileres, consultar, etc** | **select, insert, call.** | **historial\_alquileres()**  **registrar\_alquiler()**  **consultar\_vehiculos()** |

# Referencias

*MySQL :: MySQL 8.4 Reference Manual :: 15.1.17 CREATE PROCEDURE and CREATE FUNCTION Statements*. (n.d.). MySQL :: Developer Zone. Retrieved July 1, 2024, from <https://dev.mysql.com/doc/refman/8.4/en/create-procedure.html>

*MySQL :: MySQL 8.4 Reference Manual :: 15.7.1.3 CREATE USER Statement*. (n.d.). MySQL :: Developer Zone. Retrieved July 1, 2024, from <https://dev.mysql.com/doc/refman/8.4/en/create-user.html>

Rodríguez, F. (2023, December 22). *¿Qué es MySQL Workbench?* KeepCoding. Retrieved July 1, 2024, from <https://keepcoding.io/blog/que-es-mysql-workbench/>