

Seguridad de base de datos

Andres Pedraza Peña

Campuslands

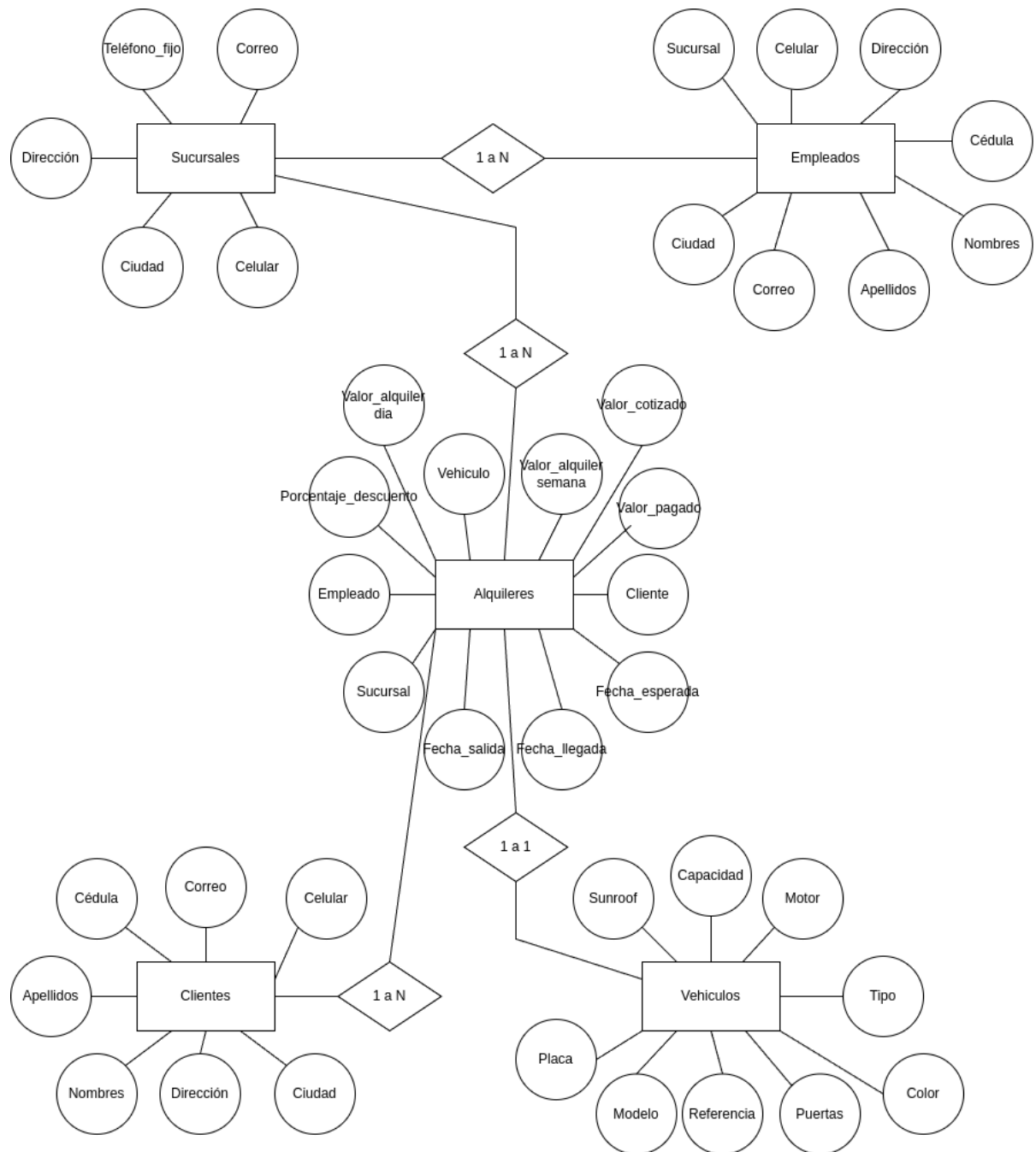
2024

Índice

1. Modelo conceptual
2. Modelo lógico
3. Normalización hasta la 3FN
4. Modelo físico
5. Inserciones
6. Procedimientos y funciones
7. Seguridad y permisos
8. Triggers y eventos

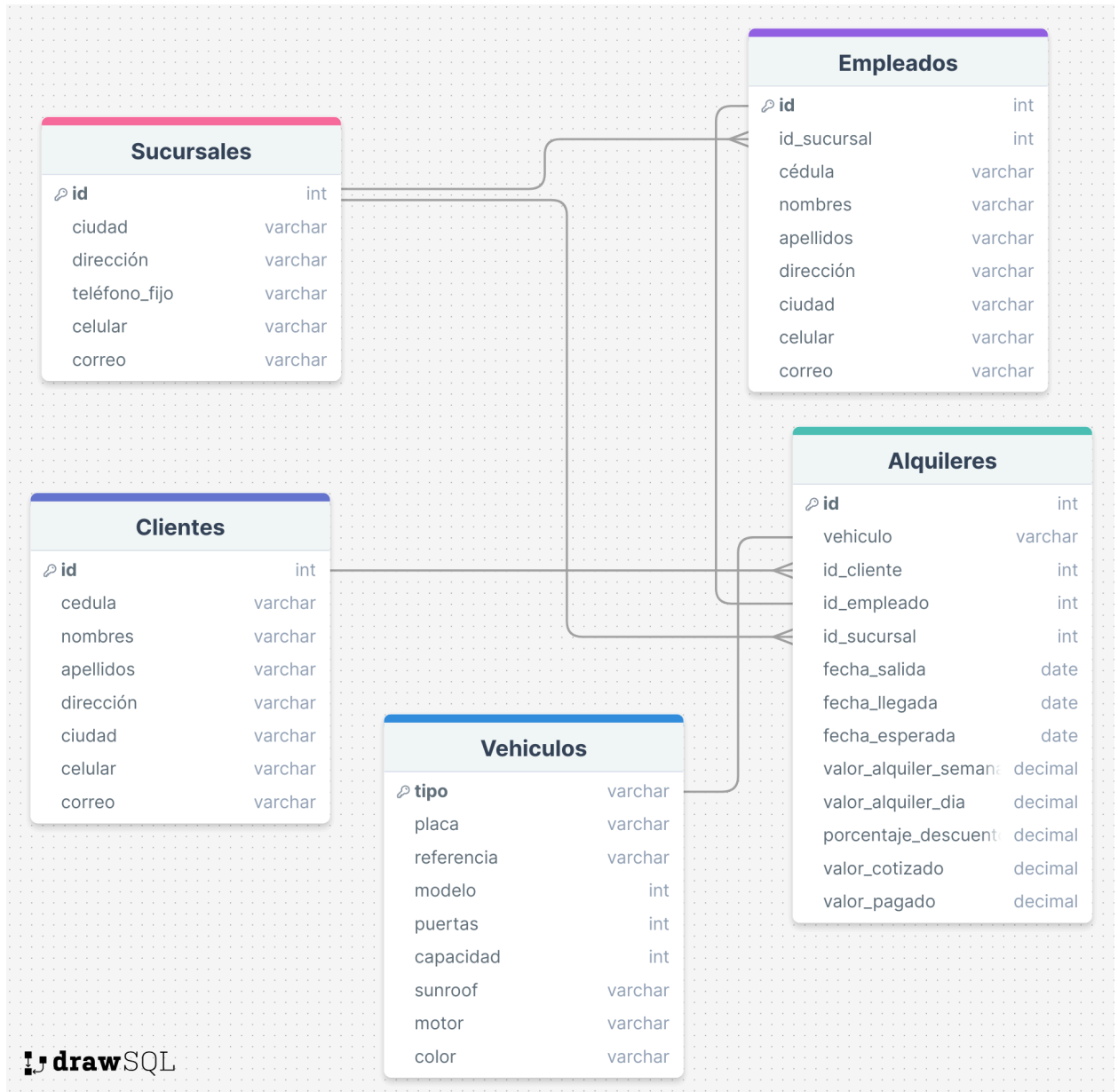
1. Modelo conceptual

El modelo conceptual incluye cinco componentes principales: sucursales, empleados, alquileres, clientes, vehículos; con sus respectivos atributos para comenzar un proceso de transformación a bases de datos.



2. Modelo lógico

De el modelo conceptual pasamos al lógico, en el cual desglosamos el tipo de dato que guardaremos en cada posible tabla como también referenciar sus llaves y su identificación de relaciones.



3. Normalización hasta la 3FN

Normalización es el proceso de organizar los datos en una base de datos para reducir la redundancia y mejorar la integridad de los datos, en este caso se realizará hasta la 3FN. Se descomponen las tablas en tablas más pequeñas y se definen relaciones entre ellas para eliminar la duplicación de datos y asegurar que las dependencias de los datos sean lógicas.

1. Sucursales:

- PK: id
- Atributos: ciudad, dirección, teléfono_fijo, celular, email

2. Empleados:

- PK: id
- FK: id_sucursal
- Atributos: cedula, nombres, apellidos, dirección, ciudad_residencia, celular, email

3. Clientes:

- PK: id
- Atributos: cedula, nombres, apellidos, dirección, ciudad_residencia, celular, email

4. Vehiculos:

- PK: id
- Atributos: tipo_vehículo, referencia, modelo, puertas, capacidad, sunroof, motor, color

5. Alquileres:

- PK: id
- FK: tipo_vehiculo, id_cliente, id_empleado, id_sucursal

- Atributos: fecha_salida, fecha_llegada, fecha_esperada, valor_semana, valor_día, descuento, valor_cotizado, valor_pagado

4. Modelo físico

En esta parte ya pasamos a concretar toda la información en la base de datos comenzando con la construcción de las tablas que en este caso es el modelo físico, esto con el fin de hacer funcional este sistema.

-- Crear tabla sucursales

```
create table sucursales(
    id int primary key,
    ciudad varchar(50),
    direccion varchar(100),
    telefono_fijo varchar(15),
    celular varchar(15),
    correo varchar(50)
);
```

-- Crear tabla empleados

```
create table empleados(
    id int primary key,
    id_sucursal int,
    foreign key(id_sucursal) references sucursales(id),
```

```
cedula varchar(20),  
nombres varchar(50),  
apellidos varchar(50),  
direccion varchar(100),  
ciudad varchar(50),  
celular varchar(15),  
correo varchar(50)  
);
```

-- Crear tabla clientes

```
create table clientes(  
    id int primary key,  
    cedula varchar(20),  
    nombres varchar(50),  
    apellidos varchar(50),  
    direccion varchar(100),  
    ciudad varchar(50),  
    celular varchar(15),  
    correo varchar(50)  
);
```

-- Crear tabla vehiculos

```
create table vehiculos(  
    id int primary key,
```

```

        tipo varchar(20) primary key,
placa varchar(10),
referencia varchar(50),
modelo int,
puertas int,
capacidad int,
sunroof varchar(5),
motor varchar(20),
color varchar(20)
);

-- Crear tabla alquileres
create table alquileres(
        id int primary key,
        tipo_vehiculo varchar(20),
        foreign key(tipo_vehiculo) references vehiculos(tipo),
        id_cliente int,
        foreign key(id_cliente) references clientes(id),
        id_empleado int,
        foreign key(id_empleado) references empleados(id),
        id_sucursal int,
        foreign key(id_sucursal) references sucursales(id),
        fecha_salida date,

```



```

fecha_llegada date,

fecha_esperada date,

valor_alquiler_semana decimal(10, 2),

valor_alquiler_dia decimal(10, 2),

porcentaje_descuento decimal(5, 2),

valor_cotizado decimal(10, 2),

valor_pagado decimal(10, 2)

);

```

5. Inserciones

En este apartado se comienza la inserción de datos en las tablas previamente creadas con el fin de gestionar y almacenar información, se crearon 100 inserciones por cada tabla que se encuentran en el documento sql, a continuación una pequeña parte de cada tabla.

```

INSERT INTO sucursales VALUES
(1, 'Bogotá', 'Calle 1 #10-20', '5712345611', '3201234561', 'sucursal1@example.com'),
(2, 'Bogotá', 'Calle 2 #10-20', '5712345612', '3201234562', 'sucursal2@example.com'),
(3, 'Bogotá', 'Calle 3 #10-20', '5712345613', '3201234563', 'sucursal3@example.com'),
(4, 'Bogotá', 'Calle 4 #10-20', '5712345614', '3201234564', 'sucursal4@example.com'),
(5, 'Bogotá', 'Calle 5 #10-20', '5712345615', '3201234565', 'sucursal5@example.com'),

```

```

INSERT INTO empleados VALUES
(1, 1, '1012345671', 'Juan', 'Pérez', 'Calle 1 #10-20', 'Bogotá', '3001234571', 'juan.perez@example.com'),
(2, 2, '1012345672', 'Carlos', 'González', 'Calle 2 #10-20', 'Bogotá', '3001234572', 'carlos.gonzalez@example.com'),
(3, 3, '1012345673', 'Luis', 'Martínez', 'Calle 3 #10-20', 'Bogotá', '3001234573', 'luis.martinez@example.com'),
(4, 4, '1012345674', 'Ana', 'Rodríguez', 'Calle 4 #10-20', 'Bogotá', '3001234574', 'ana.rodriguez@example.com'),
(5, 5, '1012345675', 'María', 'López', 'Calle 5 #10-20', 'Bogotá', '3001234575', 'maria.lopez@example.com'),

```

```

INSERT INTO clientes VALUES
(1, 1234567890, 'Juan Carlos', 'González Pérez', 'Calle 1 #23-45', 'Bogotá', '3001234567', 'juancarlos1@example.com'),
(2, 2345678901, 'María Alejandra', 'López Martínez', 'Carrera 2 #34-56', 'Medellín', '3102345678', 'mariaalejandra2@example.com'),
(3, 3456789012, 'Luis Felipe', 'Ramírez Gómez', 'Avenida 3 #45-67', 'Cali', '3203456789', 'luisfelipe3@example.com'),
(4, 4567890123, 'Ana Isabel', 'Hernández García', 'Calle 4 #56-78', 'Barranquilla', '3304567890', 'anaisabel4@example.com'),
(5, 5678901234, 'Javier Andrés', 'Suárez Pérez', 'Calle 5 #67-89', 'Cartagena', '3405678901', 'javierandres5@example.com'),

```

```
INSERT INTO vehiculos VALUES
(1, 'Sedán', 'DEF456', 'SUV', '2023', 4, 5, 'si', 'Gasolina', 'Blanco'),
(2, 'Compacto', 'GHI789', 'Hatchback', '2023', 4, 5, 'no', 'Gasolina', 'Rojo'),
(3, 'Camioneta', 'JKL012', 'Pickup', '2023', 2, 3, 'no', 'Diésel', 'Negro'),
(4, 'SUV', 'MNO345', 'SUV', '2023', 4, 5, 'si', 'Gasolina', 'Azul'),
(5, 'Sedán', 'PQR678', 'Sedán', '2023', 4, 5, 'no', 'Gasolina', 'Plata'),
```

```
INSERT INTO alquileres VALUES
(1, 1, 1, 1, 1, '2024-06-01', '2024-06-10', '2024-06-11', 500000, 75000, 0.10, 450000, 450000),
(2, 2, 2, 2, 2, '2024-06-05', '2024-06-15', '2024-06-16', 400000, 60000, 0.05, 380000, 380000),
(3, 3, 3, 3, 3, '2024-06-10', '2024-06-20', '2024-06-21', 300000, 50000, 0.07, 280000, 280000),
(4, 4, 4, 4, 4, '2024-06-15', '2024-06-25', '2024-06-26', 450000, 70000, 0.12, 396000, 396000),
(5, 5, 5, 5, 5, '2024-06-20', '2024-06-30', '2024-07-01', 500000, 75000, 0.15, 425000, 425000),
```

6. Procedimientos y funciones

En esta parte se crean los procedimientos y funciones en la base de datos que nos permiten ejecutar operaciones complejas de manera eficiente y reutilizable, con respecto a las tablas y las inserciones realizadas, se encuentran en el documento sql, a continuación una pequeña parte

```
-- Insertar sucursal
delimiter //
create procedure insertar_sucursal(sucursal_id int, sucursal_email varchar(25), sucursal_celular varchar(25), sucursal_telefono int, 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 ;

-- Actualizar sucursal
delimiter //
create procedure actualizar_sucursal(sucursal_id int, n_sucursal_id int, sucursal_email varchar(25), sucursal_celular varchar(25), sucursal_telefono int, 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 ;
```

7. Seguridad y permisos

En esta parte es donde se maneja la parte de seguridad y se le dan los permisos necesarios para que el usuario interactue con estos datos, pudiendolos modificar, actualizar, eliminar. Se encuentran en el documento sql, a continuación una pequeña parte

```
-- Creación de usuarios
create user 'empleado'@'%' identified by 'empleado_publico';
create user 'cliente'@'%' identified by 'cliente_publico';

-- Asignar permisos a empleados para que accedan a la tabla sucursales, vehiculos, empleados y base de datos
grant execute on procedure mysql2_dia4.insertar_sucursal to 'empleado'@'%';
grant execute on procedure mysql2_dia4.actualizar_sucursal to 'empleado'@'%';
grant execute on procedure mysql2_dia4.registrar_empleado to 'empleado'@'%';
grant execute on procedure mysql2_dia4.actualizar_empleado to 'empleado'@'%';
grant execute on procedure mysql2_dia4.registrar_vehiculo to 'empleado'@'%';
grant execute on procedure mysql2_dia4.actualizar_vehiculo to 'empleado'@'%';
```

8. Triggers y eventos

Por último realizamos triggers que son los que permiten la ejecución automática de código en respuesta a ciertos cambios o a intervalos de tiempo, estos nos ayudan a mantener la integridad de los datos y ejecutar procesos de manera eficiente. Se encuentran en el documento sql, a continuación una pequeña parte

```
create table inserciones_actualizaciones_empleados (
  id int primary key,
  nombres varchar(50),
  apellidos varchar(50),
  direccion varchar(100),
  ciudad varchar(50),
  celular varchar(15),
  correo varchar(50),
  action varchar(25),
  hora_actualizacion timestamp default current_timestamp
);

delimiter //
create trigger after_empleado_insert
after insert on empleados
for each row
begin
  insert into inserciones_actualizaciones_empleados (id, nombres, apellidos, direccion, ciudad, celular, correo, action)
  values (new.id, new.nombres, new.apellidos, new.direccion, new.ciudad, new.celular, new.correo, 'insert');
end //
delimiter ;
```