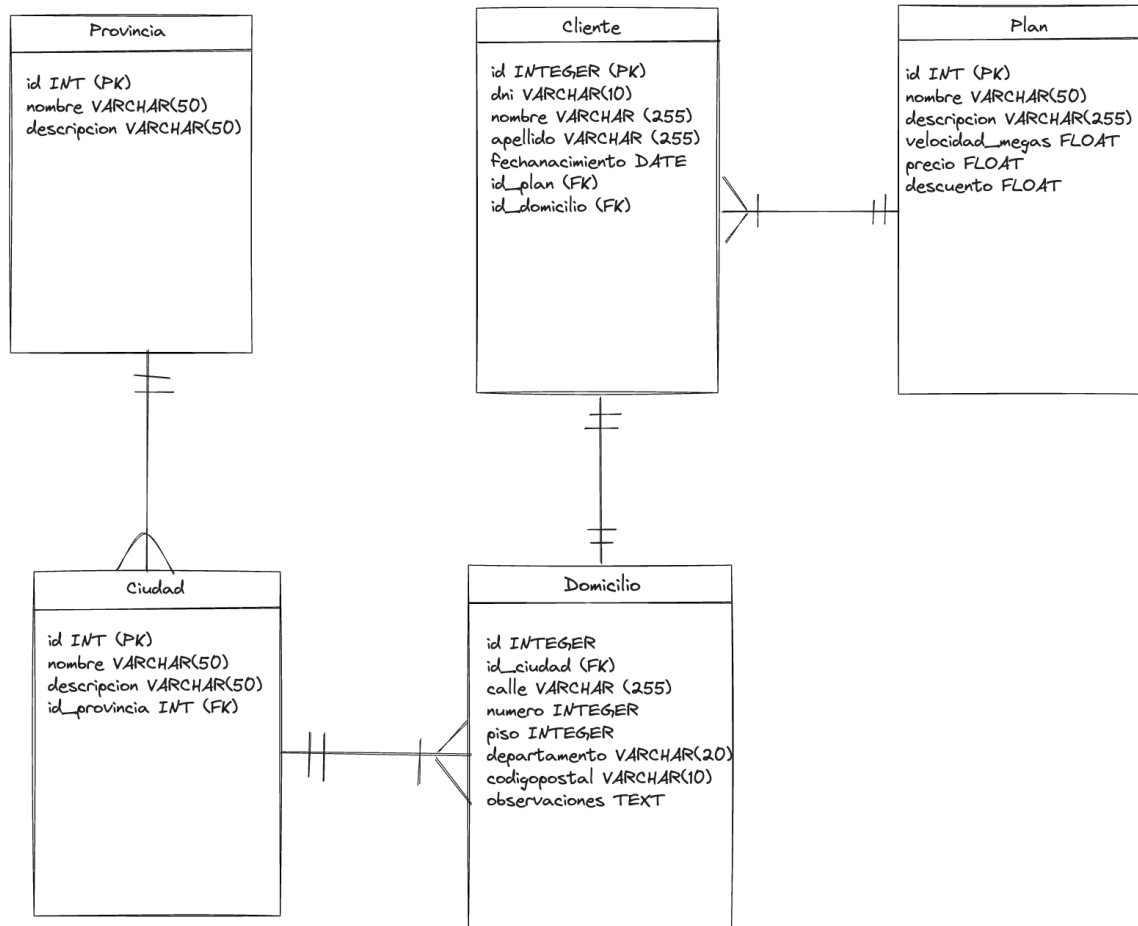


Práctica grupal DER y SQL

Ejercicio Learning-Hub: DB relacionales - Parte 2 VIVO

Luciano Rodríguez, Martín Perren, John Rios, Johan Sebastian - Mercado Libre

Ejercicio 1



Ejercicio 2

Una vez modelada y planteada la base de datos, responder a las siguientes preguntas:

a. ¿Cuál es la primary key para la tabla de clientes? Justificar respuesta

La PK de la tabla clientes la definimos como ID, ya que no podemos considerar el DNI como una clave única, debemos definir esta clave subrogada.

b. ¿Cuál es la primary key para la tabla de planes de internet? Justificar respuesta.

La clave primaria es id, ya que esta identifica unívocamente al plan. Esta es una clave única y es subrogada.

c. ¿Cómo serían las relaciones entre tablas? ¿En qué tabla debería haber foreign key? ¿A qué campo de qué tabla hace referencia dicha foreign key? Justificar respuesta.

En la tabla cliente se encuentra el id_plan que es una clave foránea para poder identificar y relacionarlo con el plan contratado por el mismo. Esta clave hace referencia al campo ID de la tabla plan, la cual es clave primaria.

Una vez realizado el planteo del diagrama y de haber respondido estas preguntas, utilizar **PHPMyAdmin o MySQL Workbench** para ejecutar lo siguiente:

Se solicita crear una nueva base de datos llamada “**empresa_internet**”.

```
CREATE DATABASE empresa_internet CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

```
CREATE TABLE Provincia (  
    id INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    descripcion VARCHAR(50)  
);
```

```
CREATE TABLE Ciudad (  
    id INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    descripcion VARCHAR(50),  
    id_provincia INT,  
    FOREIGN KEY (id_provincia) REFERENCES Provincia(id)  
);
```

```
CREATE TABLE Domicilio (  
    id INT PRIMARY KEY,  
    id_ciudad INT,  
    calle VARCHAR(255),  
    numero INT,  
    piso INT,
```

```
departamento VARCHAR(20),
codigo_postal VARCHAR(10),
observaciones VARCHAR(255),
FOREIGN KEY (id_ciudad) REFERENCES Ciudad(id)
);
```

```
CREATE TABLE Plan (
    id INT PRIMARY KEY,
    nombre VARCHAR(50),
    descripcion VARCHAR(255),
    velocidad_megas FLOAT,
    precio FLOAT,
    descuento FLOAT
);
```

```
CREATE TABLE Cliente (
    id INT PRIMARY KEY,
    apellido VARCHAR(50),
    descripcion VARCHAR(50),
    dni VARCHAR(50),
    fecha_nacimiento DATE,
    id_plan INT,
    id_domicilio INT,
    FOREIGN KEY (id_plan) REFERENCES Plan(id),
    FOREIGN KEY (id_domicilio) REFERENCES Domicilio(id)
);
```

Incorporar 10 registros en la tabla de clientes y 5 en la tabla de planes de internet.
Realizar las asociaciones/relaciones correspondientes entre estos registros.

```
-- Insertar provincias
INSERT INTO Provincia (id, nombre, descripcion) VALUES
(1, 'Buenos Aires', 'Provincia de Buenos Aires'),
(2, 'Córdoba', 'Provincia de Córdoba'),
(3, 'Santa Fe', 'Provincia de Santa Fe');

-- Insertar ciudades
```

```
INSERT INTO Ciudad (id, nombre, descripcion, id_provincia) VALUES
(1, 'La Plata', 'Capital de la Provincia de Buenos Aires', 1),
(2, 'Mar del Plata', 'Ciudad turística de la Provincia de Buenos Aires', 1),
(3, 'Córdoba', 'Capital de la Provincia de Córdoba', 2),
(4, 'Rosario', 'Ciudad más poblada de la Provincia de Santa Fe', 3);
```

-- Insertar planes de internet

```
INSERT INTO Plan (id, nombre, descripcion, velocidad_megas, precio, descuento)
VALUES
(1, 'Básico', 'Plan básico de internet', 10.0, 20.0, 0),
(2, 'Estándar', 'Plan estándar de internet', 50.0, 40.0, 0),
(3, 'Premium', 'Plan premium de internet', 100.0, 60.0, 0),
(4, 'Ultra', 'Plan ultra de internet', 500.0, 80.0, 0),
(5, 'MegaUltra', 'Plan mega ultra de internet', 1000.0, 100.0, 0);
```

-- Insertar domicilios

```
INSERT INTO Domicilio (id, id_ciudad, calle, numero, piso, departamento,
codigo_postal, observaciones) VALUES
(1, 1, 'Calle Falsa', 123, NULL, NULL, '1900', 'Observaciones 1'),
(2, 2, 'Avenida Siempre Viva', 456, 3, 'B', '2000', 'Observaciones 2'),
(3, 3, 'Calle Principal', 789, NULL, NULL, '3000', 'Observaciones 3'),
(4, 4, 'Avenida Libertador', 1011, 10, 'A', '4000', 'Observaciones 4');
```

-- Insertar clientes

```
INSERT INTO Cliente (id, apellido, descripcion, dni, fecha_nacimiento, id_plan,
id_domicilio) VALUES
(1, 'González', 'Cliente premium', '12345678', '1990-01-01', 3, 1),
(2, 'Rodríguez', 'Cliente estándar', '23456789', '1985-02-02', 2, 2),
(3, 'López', 'Cliente básico', '34567890', '1980-03-03', 1, 3),
(4, 'Martínez', 'Cliente ultra', '45678901', '1975-04-04', 4, 4),
(5, 'Sánchez', 'Cliente mega ultra', '56789012', '1970-05-05', 5, 1),
(6, 'Pérez', 'Cliente básico', '67890123', '1965-06-06', 1, 2),
(7, 'Gómez', 'Cliente premium', '78901234', '1960-07-07', 3, 3),
(8, 'Díaz', 'Cliente ultra', '89012345', '1955-08-08', 4, 4),
(9, 'Vázquez', 'Cliente mega ultra', '90123456', '1950-09-09', 5, 1),
(10, 'Fernández', 'Cliente estándar', '01234567', '1945-10-10', 2, 2);
```

Plantear 10 consultas SQL que se podrían realizar a la base de datos. Expresar las sentencias.

1- Obtener todos los clientes junto con sus planes de internet asociados:

```
SELECT c.id, c.apellido, p.nombre AS plan
FROM Cliente c
JOIN Plan p ON c.id_plan = p.id;
```

2- Encontrar los clientes que tienen un plan de internet premium:

```
SELECT id, apellido
FROM Cliente
WHERE id_plan = (SELECT id FROM Plan WHERE nombre = 'Premium');
```

3- Mostrar todos los planes de internet con una velocidad de conexión superior a 100 Mbps.

```
SELECT *
FROM Plan
WHERE velocidad_megas > 100;
```

4- Mostrar la descripción de todos los planes de internet ordenados por precio de forma descendente:

```
SELECT nombre, descripcion, precio
FROM Plan
ORDER BY precio DESC;
```

5- Obtener los clientes que nacieron después de 1980.

```
SELECT id, apellido, fecha_nacimiento
FROM Cliente
```

```
WHERE YEAR(fecha_nacimiento) > 1980;
```

6- Mostrar la cantidad de clientes por plan de internet:

```
SELECT p.nombre AS plan, COUNT(c.id) AS cantidad_clientes  
FROM Cliente c  
JOIN Plan p ON c.id_plan = p.id  
GROUP BY p.nombre;
```

7- Encontrar los planes de internet con un precio inferior a \$50:

```
SELECT *  
FROM Plan  
WHERE precio < 50.0;
```

8- Mostrar los domicilios con código postal '1900':

```
SELECT *  
FROM Domicilio  
WHERE codigo_postal = '1900';
```

9- Encontrar los clientes que tienen más de 60 años de edad:

```
SELECT *  
FROM Cliente  
WHERE YEAR(CURDATE()) - YEAR(fecha_nacimiento) > 60;
```

10- Mostrar los nombres y velocidades de conexión de todos los planes de internet:

```
SELECT nombre, velocidad_megas  
FROM Plan;
```