

Una empresa proveedora de Internet necesita una base de datos para almacenar cada uno de sus clientes junto con el plan/pack que tiene contratado.

Mediante un análisis previo se conoce que se tiene que almacenar la siguiente información:

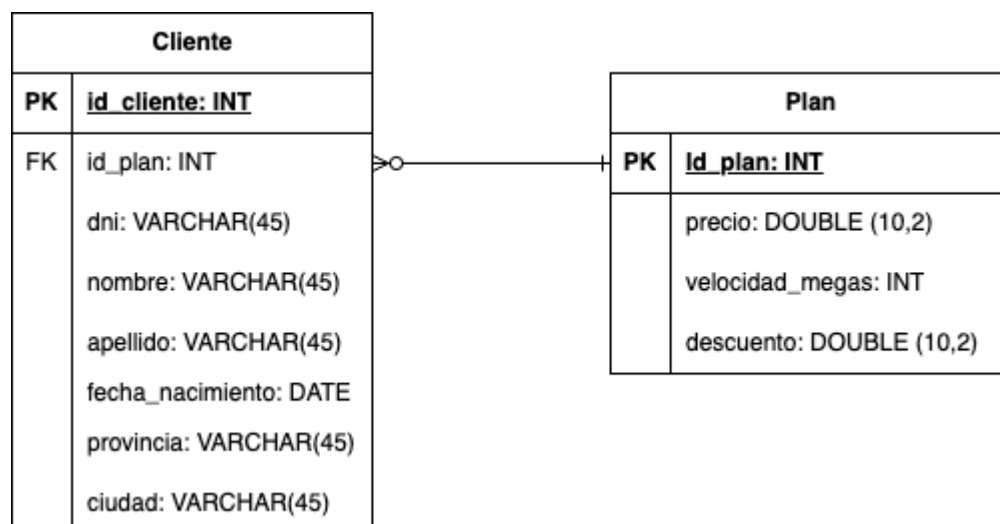
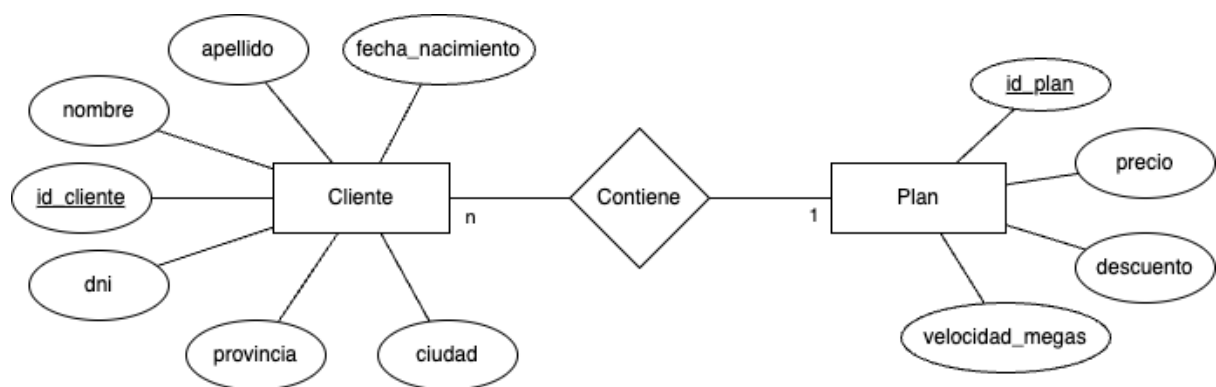
De los clientes se debe registrar: dni, nombre, apellido, fecha de nacimiento, provincia, ciudad.

En cuanto a los planes de internet: identificación del plan, velocidad ofrecida en megas, precio, descuento.



Ejercicio 1

Luego del planteo de los requerimientos de la empresa, se solicita modelar los mismos mediante un DER (Diagrama Entidad-Relación).





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

R/ La primary key de la tabla clientes es el "id" ya que es el identificador único de cada cliente.

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

R/ La primary key de la tabla planes es el "id" ya que es el identificador único de cada plan.

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.

R/

- La relación entre las tablas es de uno a muchos, dado que cada cliente puede comprar un plan, pero cada plan puede ser comprado por muchos clientes.
- En la tabla de Cliente.
- Al campo "id_plan" de la tabla Plan.



Ejercicio 3

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**".

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.

SQL:

```
CREATE DATABASE IF NOT EXISTS empresa_internet;

create table Plan
(
    id_plan INT NOT NULL AUTO_INCREMENT,
    velocidad_megas INT NOT NULL,
    precio DOUBLE(10,2) NOT NULL,
    descuento DOUBLE(10,2),
    PRIMARY KEY (id_plan)
```

```

);

create table Cliente (
    id_cliente INT NOT NULL AUTO_INCREMENT,
    dni INT NOT NULL,
    nombre varchar(45) NOT NULL,
    apellido varchar(45) NOT NULL,
    fecha_nacimiento date NOT NULL,
    provincia varchar(45) NOT NULL,
    ciudad varchar(45) NOT NULL,
    id_plan INT NOT NULL,
    PRIMARY KEY (id_cliente),
    Foreign Key (id_plan) REFERENCES Plan(id_plan)
);

-- Insertar 5 planes
INSERT INTO Plan (velocidad_megas, precio, descuento)
VALUES
    (100, 50.00, 0.00),
    (200, 75.00, 0.10),
    (300, 100.00, 0.15),
    (400, 125.00, 0.20),
    (500, 150.00, 0.25);

-- Insertar 10 clientes
INSERT INTO Cliente (dni, nombre, apellido, fecha_nacimiento,
provincia, ciudad, id_plan)
VALUES
    (123456789, 'John', 'Doe', '1990-01-01', 'California', 'Los
Angeles', 1),
    (987654321, 'Jane', 'Smith', '1995-02-15', 'New York', 'New York
City', 2),
    (456789123, 'Michael', 'Johnson', '1985-05-10', 'Texas',
'Houston', 3),
    (789123456, 'Emily', 'Williams', '1992-09-20', 'Florida',
'Miami', 4),
    (321654987, 'David', 'Brown', '1988-03-05', 'Illinois',
'Chicago', 5),
    (654987321, 'Sarah', 'Davis', '1998-07-12', 'Arizona',
'Phoenix', 1),
    (987321654, 'Daniel', 'Miller', '1993-11-25', 'Colorado',
'Denver', 2),
    (741852963, 'Olivia', 'Wilson', '1991-04-30', 'Washington',
'Seattle', 3),
    (369258147, 'James', 'Anderson', '1987-08-15', 'Massachusetts',

```

```
'Boston', 4),  
    (852963741, 'Sophia', 'Taylor', '1996-12-10', 'Nevada', 'Las  
Vegas', 5);
```



Ejercicio 4

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

#Consultas

-- 1. Obtener clientes por plan

```
select nombre, apellido from Cliente where id_plan = 1;
```

-- 2. seleccionar los clientes cuyo apellido contengan la letra "S"

```
SELECT id_cliente, apellido FROM Cliente WHERE apellido LIKE '%S%';
```

-- 3. Calcular promedio de precios}

```
SELECT AVG(precio) AS promedio_precios FROM Plan;
```

-- 4. Obtener planes con planes con descuento

```
SELECT * FROM Plan WHERE descuento > 0;
```

-- 5. Obtener clientes por provincia

```
SELECT * FROM Cliente WHERE provincia = 'California';
```

-- 6. Obtener clientes por ciudad

```
SELECT * FROM Cliente WHERE ciudad = 'Los Angeles';
```

-- 7. Obtener clientes por rango de fechas de nacimiento

```
SELECT * FROM Cliente WHERE fecha_nacimiento BETWEEN '1990-01-01' AND '1995-02-15';
```

-- 8. Obtener planes por rango de precios

```
SELECT * FROM Plan WHERE precio BETWEEN 50.00 AND 100.00;
```

-- 9. Obtener planes por rango de velocidades

```
SELECT * FROM Plan WHERE velocidad_megas BETWEEN 100 AND 300;
```

-- 10. Obtener planes por rango de descuentos

```
SELECT * FROM Plan WHERE descuento BETWEEN 0.10 AND 0.20;
```