

EJERCICIO 5

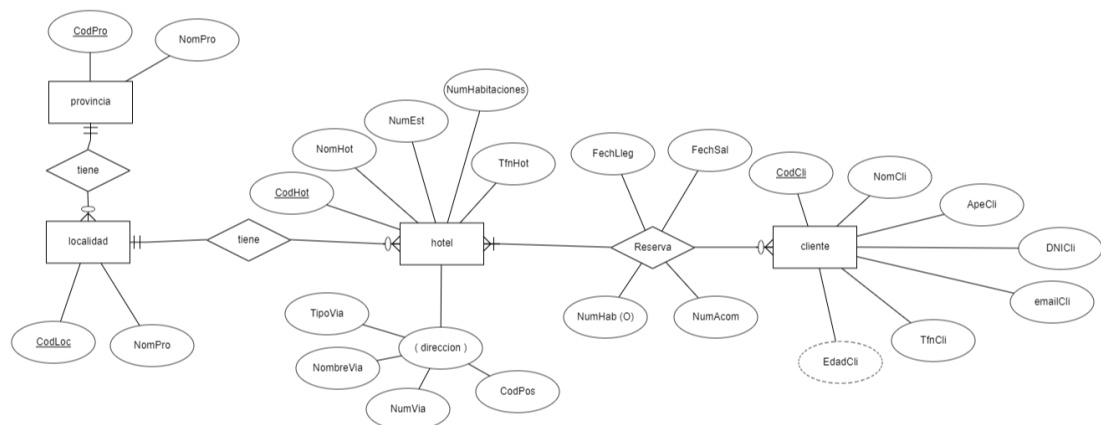
Objetivo del ejercicio:

1. Repasar el tipo de correspondencia muchos a muchos (m a n). En este caso la clave primaria no podrá ser compuesta.
2. Repasar los conceptos de atributo compuesto, atributo derivado y atributo opcional.
3. Repasar las cardinalidades mínimas.
4. Aprender que las relaciones de un tipo de correspondencia m a n pueden llevar atributos.
5. Aprender a usar funciones como expresión en el **SELECT** y como expresión en un predicado básico.

Se quiere diseñar una base de datos para almacenar información sobre hoteles de una misma cadena, y de los clientes que se alojan en los mismos. De los hoteles debemos conocer, su código, nombre, número de habitaciones, número de estrellas, dirección completa y el teléfono. De los clientes, su código, nombre, apellidos, DNI, email, teléfono y edad. Se desea también conocer la fecha de llegada del cliente al hotel, el número de habitación en el que se alojó, número de acompañantes, así como la fecha de salida del hotel. Tener en cuenta, que un cliente puede alojarse en un mismo hotel todas las veces que quiera en diferentes fechas. Pueden existir hoteles de reciente incorporación al sistema que aún no se hayan alojado clientes en los mismos. Cuando a un cliente se le da de alta debe indicar en qué hotel desea alojarse.

Se pide:

1. Modelar la base de datos. Para ello haremos:
 - a. Diseño Conceptual de Datos utilizando un Diagrama o Modelo Entidad-Relación. Lo hacemos en papel y lo pasamos a la Herramienta CASE ERD Plus.

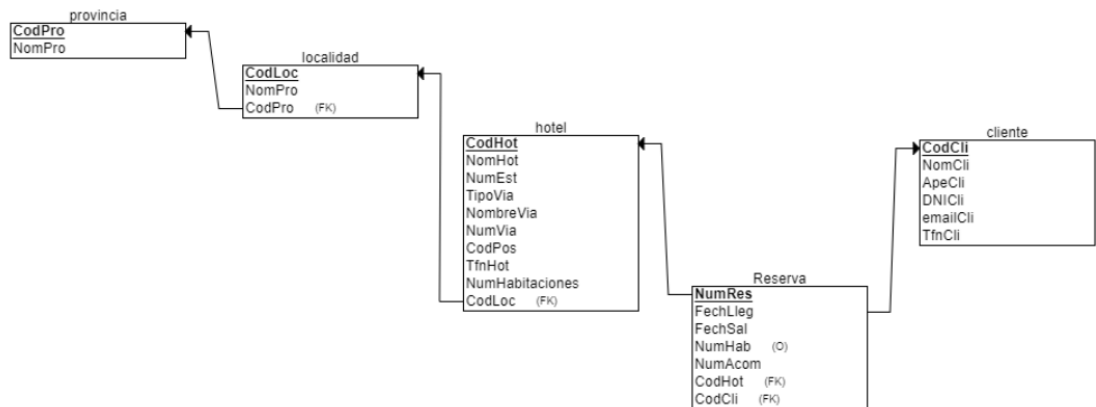


EJERCICIO 5

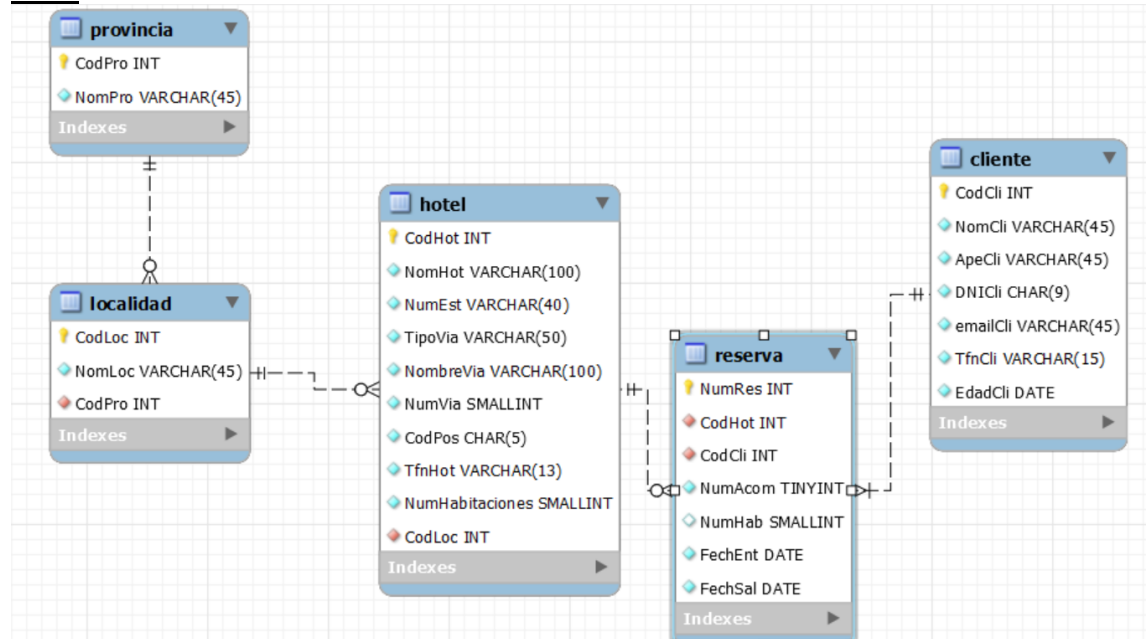
- b. Diseño Lógico de Datos utilizando un Diagrama de Estructura de datos (DED). Lo hacemos en papel y lo pasamos a la Herramienta CASE MySql Workbench. En este apartado también vamos a poner el Diagrama Referencial que genera ERD Plus a partir del Modelo Entidad-Relación.

Recuerda que el Diseño Lógico de Datos es hacer el modelo relacional y para ello podemos hacer un DED o un Diagrama Referencial.

Diagrama Referencial



DED



c. Diseño Físico de Datos. Creamos la base de datos y las tablas en SQL.

```
CREATE DATABASE ejercicio5;
```

```
CREATE TABLE cliente
(
  CodCli INT NOT NULL,
  NomCli VARCHAR(50) NOT NULL,
  ApeCli VARCHAR(50) NOT NULL,
  DNICli CHAR(9) NOT NULL,
  emailCli VARCHAR(100) NOT NULL,
  TfnCli VARCHAR(15) NOT NULL,
  EdadCli DATE NOT NULL,
  PRIMARY KEY (CodCli)
);
```

```
CREATE TABLE provincia
(
  CodPro INT NOT NULL,
  NomPro VARCHAR(50) NOT NULL,
  PRIMARY KEY (CodPro)
);
```

```
CREATE TABLE localidad
(
  CodLoc INT NOT NULL,
  NomLoc VARCHAR(50) NOT NULL,
  CodPro INT NOT NULL,
  PRIMARY KEY (CodLoc),
  FOREIGN KEY (CodPro) REFERENCES provincia(CodPro)
);
```

```
CREATE TABLE hotel
(
  CodHot INT NOT NULL,
  NomHot VARCHAR(100) NOT NULL,
  NumEst VARCHAR(40) NOT NULL,
  TipoVia VARCHAR(50) NOT NULL,
  NombreVia VARCHAR(100) NOT NULL,
  NumVia SMALLINT NOT NULL,
  CodPos CHAR(5) NOT NULL,
  TfnHot VARCHAR(13) NOT NULL,
  NumHabitaciones SMALLINT NOT NULL,
  CodLoc INT NOT NULL,
  PRIMARY KEY (CodHot),
  FOREIGN KEY (CodLoc) REFERENCES localidad(CodLoc)
);
```

```
CREATE TABLE reserva
(
    NumRes INT NOT NULL,
    FechLleg DATE NOT NULL,
    FechSal DATE NOT NULL,
    NumHab SMALLINT,
    NumAcom TINYINT NOT NULL,
    CodHot INT NOT NULL,
    CodCli INT NOT NULL,
    PRIMARY KEY (NumRes),
    FOREIGN KEY (CodHot) REFERENCES hotel(CodHot),
    FOREIGN KEY (CodCli) REFERENCES cliente(CodCli)
);
```

2. Insertar datos desde phpmyadmin.

```
INSERT INTO cliente (CodCli, NomCli, ApeCli, DNICli, emailCli, TfnCli,
EdadCli ) VALUES (1, 'Cliente1', 'Apellidos1', '12345678A',
'email1@gmail.com', '600123456', '1990-12-20');
```

```
INSERT INTO cliente (CodCli, NomCli, ApeCli, DNICli, emailCli, TfnCli,
EdadCli )
VALUES
(2, 'Cliente2', 'Apellidos2', '12345678B', 'email2@gmail.com', '601123456',
'1980-10-04'),
(3, 'Cliente3', 'Apellidos3', '12345678C', 'email3@gmail.com', '602123456',
'1979-11-10'),
(4, 'Cliente4', 'Apellidos4', '12345678D', 'email4@gmail.com', '603123456',
'1972-05-24');
```

```
INSERT INTO provincia (CodPro, NomPro) VALUES (1, 'Málaga'), (2,
'Sevilla');
```

```
INSERT INTO localidad (CodLoc, NomLoc, CodPro) VALUES (1,
'Torremolinos', 1), (2, 'Marbella', 1);
```

```
INSERT INTO hotel (CodHot, NomHot, NumEst, TipoVia, NombreVia,
NumVia, CodPos, TfnHot, NumHabitaciones, CodLoc)
VALUES
(1, 'Hotel1', '3 Estrellas', 'Avenida', 'Sol', 20, '29620', '952123456', 50, 1),
(2, 'Hotel2', '4 Estrellas', 'Calle', 'Agua', 40, '29600', '952222222', 100, 2),
(3, 'Hotel3', '5 Estrellas', 'Avenida', 'Mar', 10, '29600', '952333333', 50, 2);
```

```
INSERT INTO reserva (NumRes, FechLleg, FechSal, NumHab, NumAcom,
CodHot, CodCli)
```

```
VALUES (1000, '2021-09-05', '2021-09-16', 10, 3, 1, 1), (1001, '2021-04-05', '2021-04-18', 20, 1, 2, 2);
```

```
INSERT INTO reserva (NumRes, FechLleg, FechSal, NumHab, NumAcom, CodHot, CodCli)
```

```
VALUES (1002, '2022-01-05', '2022-01-16', 10, 3, 1, 1), (1003, '2022-03-05', '2023-03-18', 20, 1, 2, 2);
```

3. Realizar las siguientes consultas en SQL:

- Muestra todas las filas y todos los campos de las tablas. Ordenar el resultado de la consulta.
- Muestra algunos campos de las tablas.
- Mostrar el nombre de todos los clientes junto el código del hotel donde tiene reserva y la fecha de entrada y de salida de la misma. Uso de función escalar DATE_FORMAT como expresión del SELECT)

```
SELECT NomCli, ApeCli, CodHot, DATE_FORMAT(FechLleg, '%d - %m - %Y') AS 'Fecha Llegada', DATE_FORMAT(FechSal, '%d - %m - %Y') AS 'Fecha Salida'
```

```
FROM cliente C JOIN reserva R ON (C.CodCli=R.CodCli);
```

- Mostrar el nombre de todos los hoteles junto con el código del cliente que reserva. Si el hotel aún no tiene ninguna reserva, el nombre del hotel debe salir igualmente y en código del cliente debe salir el mensaje 'Sin Reservas'. (Uso de funciones escalares IFNULL e IF como expresión del SELECT)

```
SELECT NomHot, IFNULL(CodCli,'Sin Reservas')
```

```
FROM hotel H LEFT JOIN reserva R ON (H.CodHot=R.CodHot);
```

Obtenemos el mismo resultado si usamos la función IF.

```
SELECT NomHot, IF(CodCli IS NULL,'Sin Reservas', CodCli)
```

```
FROM hotel H LEFT JOIN reserva R ON (H.CodHot=R.CodHot);
```

Si queremos sacar también el nombre de los clientes haremos lo siguiente:

```
SELECT NomHot, IF(C.CodCli IS NULL,'Sin Reservas', C.CodCli), IFNULL (NomCli,'Sin Datos'), IFNULL(ApeCli,'Sin Datos')
```

EJERCICIO 5

```
FROM (hotel H LEFT JOIN reserva R ON (H.CodHot=R.CodHot)) LEFT  
JOIN cliente C ON (c.CodCli=R.CodCli);
```

- Mostrar el nombre de todos los hoteles en los que han reservado los clientes el año pasado. (Uso de funciones escalares YEAR y SYSDATE como expresión de un predicado básico)

```
SELECT NomHot
```

```
FROM hotel H JOIN reserva R ON (H.CodHot=R.CodHot)
```

```
WHERE YEAR(SYSDATE())-1=YEAR(FechLleg);
```

- Mostrar el nombre de todos los hoteles, junto con el nombre de los clientes que han reservado alguna vez en los mismos en el presente año. (Uso de funciones escalares YEAR y SYSDATE como expresión de un predicado básico)

```
SELECT NomHot, NomCli, ApeCli
```

```
FROM (cliente C JOIN reserva R ON (C.CodCli=R.CodCli)) JOIN hotel H  
ON (H.CodHot=R.CodHot)
```

```
WHERE YEAR(SYSDATE())=YEAR(FechLleg);
```

- Mostrar cuantas reservas se han realizado en el año actual. (Uso de función de agregado COUNT como expresión del SELECT. Uso de funciones escalares YEAR y SYSDATE como expresión de un predicado básico)

```
SELECT COUNT(*) AS 'Reservas realizadas en el año actual'
```

```
FROM reserva
```

```
WHERE YEAR(SYSDATE())=YEAR(FechLleg);
```