

Entrega 2 IIC2413

Grupo 25

Catalina Garrido
Rodrigo Espinoza

Fecha entrega: 30 Abril 2019

1 Esquemas

1.1 Tablas

- Usuarios (id_usuario INT PRIMARY KEY, nombre VARCHAR(50), fecha_nacimiento date, correo VARCHAR(50), nacionalidad VARCHAR(50))
- Regiones (id_región INT PRIMARY KEY, nombre VARCHAR(50), reseña VARCHAR(1000))
- Hoteles_Región(id_región INT, id_hotel INT, PRIMARY KEY(id_hotel, id_región), FOREIGN KEY(id_región) REFERENCES Regiones(id_región))
- Hoteles (id_hotel INT PRIMARY KEY, nombre_hotel VARCHAR(100), dirección_hotel VARCHAR(200), teléfono_hotel VARCHAR(20), descripción_hotel VARCHAR(1000), estrellas INT, FOREIGN KEY(id_hotel) REFERENCES Hotel_Región(id_hotel))
- Habitaciones_Hotel(id_hotel, id_habitación, PRIMARY KEY(id_hotel, id_habitación), FOREIGN KEY(id_hotel) REFERENCES Hoteles(id_hotel))
- Habitaciones (id_habitación INT PRIMARY KEY, nombre_habitación VARCHAR(100), precio_habitación FLOAT, id_hotel INT, FOREIGN KEY(id_hotel) REFERENCES Hoteles(id_hotel), FOREIGN KEY(id_habitación) REFERENCES Habitaciones_Hotel(id_habitación))
- Reservas(id_reserva INT PRIMARY KEY, id_usuario INT, id_habitación INT, fecha_inicio DATE, fecha_fin DATE, FOREIGN KEY(id_usuario) REFERENCES Usuarios(id_usuario), FOREIGN KEY(id_habitación) REFERENCES Habitaciones(id_habitación))
- Restaurantes (id_restaurant INT PRIMARY KEY, nombre_restaurant VARCHAR(100), dirección_restaurant VARCHAR(200), teléfono_restaurant VARCHAR(20), descripción_restaurant VARCHAR(1000), FOREIGN KEY(id_restaurante) REFERENCES Restaurante_Región(id_restaurante))

- Restaurantes_Región(id_región INT, id_restaurant INT, PRIMARY KEY(id_restaurant, id_región), FOREIGN KEY(id_región) REFERENCES Regiones(id_región))
- Producto_Restaurante(id_restaurant INT, id_producto INT, PRIMARY KEY(id_restaurant, id_producto), FOREIGN KEY(id_restaurant) REFERENCES Restaurantes(id_restaurant))
- Productos (id_producto INT PRIMARY KEY, nombre_producto VARCHAR(100), descripción_producto VARCHAR(1000), precio_producto FLOAT, FOREIGN KEY(id_producto) REFERENCES Producto_Restaurante(id_producto))
- Agencias_turismo (id_agencia INT PRIMARY KEY, nombre_agencia VARCHAR(100), dirección_agencia VARCHAR(100), teléfono_agencia VARCHAR(15), FOREIGN KEY(id_agencia) REFERENCES Agencia_Región(id_agencia))
- Agencia_Región(id_región INT, id_hotel INT, PRIMARY KEY(id_hotel, id_región), FOREIGN KEY(id_región) REFERENCES Regiones(id_región))
- Agencia_Tour(id_agencia INT, id_tour INT, PRIMARY KEY(id_agencia, id_tour), FOREIGN KEY(id_agencia) REFERENCES Agencia(id_agencia))
- Tour (id_tour INT PRIMARY KEY, descripción_tour VARCHAR(1000), precio_tour FLOAT, id_agencia INT, FOREIGN KEY(id_tour) REFERENCES Agencia_Tour(id_tour))

1.2 Justificación BCNF

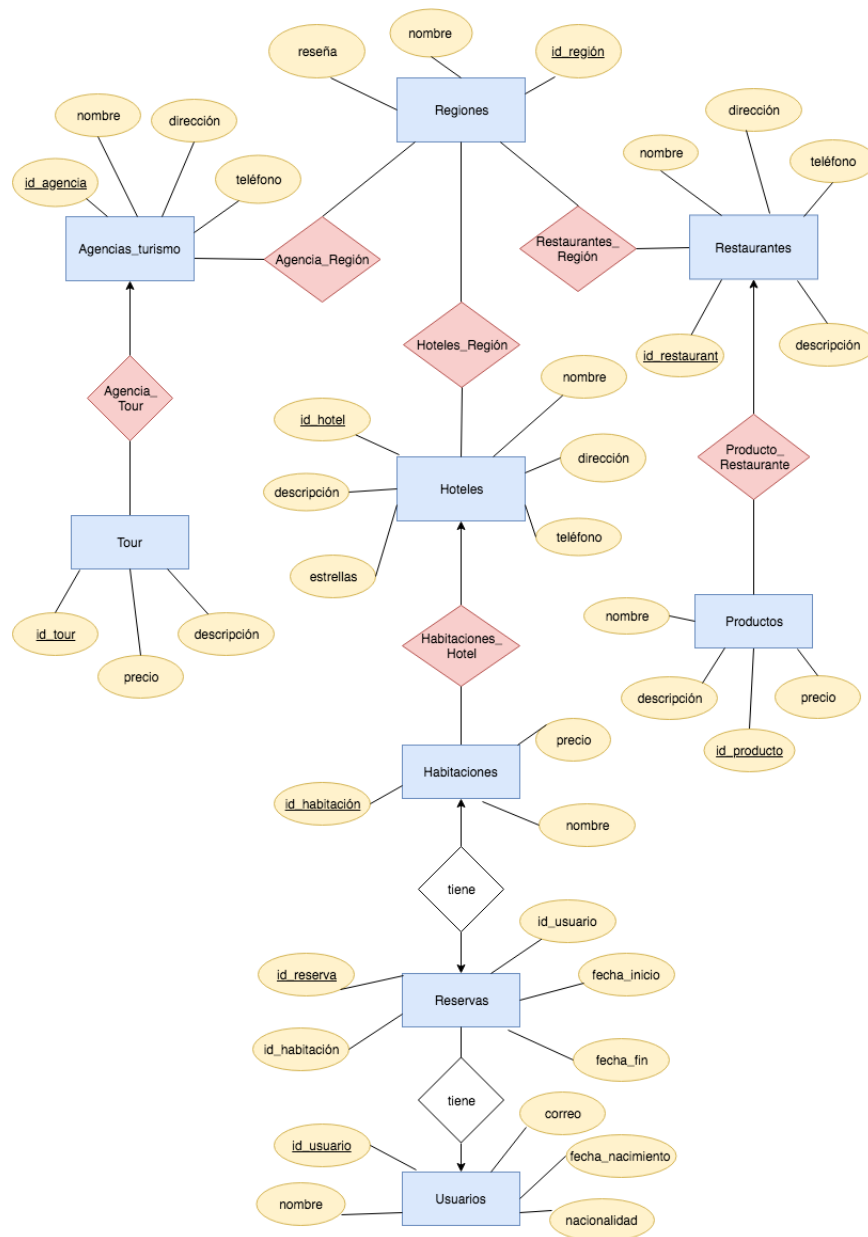
Para justificar que nuestro modelo está en BCNF justificaremos que cada una de nuestras tablas está en BCNF. No justificaremos cada una de las tablas intermedias ya que para ellas siempre ambos valores son PRIMARY KEY, por lo tanto, siempre estarán en BCNF.

También cuando decimos que x depende de y , nos referimos a que depende solamente de eso y no de otra cosa. EJ: precio depende de id_plato, solamente depende de id_plato y no de descripción y el resto de los atributos.

- Usuarios: Como id_usuario es nuestra PRIMARY KEY, el nombre, fecha nacimiento, correo y nacionalidad dependen del usuario, en este caso de su id que es la PRIMARY KEY plt BCNF (Por lo tanto, está en BCNF)
- Regiones: Teniendo id_región como PRIMARY KEY, el nombre de una region, y su reseña dependen de la region, en este caso de su id que es nuevamente la PRIMARY KEY plt BCNF
- Hoteles: Tenemos id_hotel como la PRIMARY KEY, luego el nombre del hotel, su dirección, teléfono, descripción y estrellas dependen solo del hotel que en este caso esta identificado por su ID que es PRIMARY KEY, plt BCNF

- Habitaciones: id_habitación es PRIMARY KEY, el nombre de la habitación y su precio dependen de la id de habitación que es PRIMARY KEY, plt BCNF
- Reservas: id_reserva es nuestra PRIMARY KEY, luego el usuario que realiza la reserva, la habitación reservada y la fecha de inicio y término dependen de la reserva que es nuevamente PRIMARY KEY, plt BCNF
- Restaurantes: id_restaurant es nuestra PRIMARY KEY, el nombre del restaurant, su dirección, teléfono y la descripción dependen del restaurant, que es id_restaurant PRIMARY KEY, plt BCNF
- Productos: id_producto es PRIMARY KEY, el nombre, descripción, precio dependen solo del producto, en específico su id que es PRIMARY KEY, plt BCNF
- Agencias_turismo: id_agencia es PRIMARY KEY, donde el nombre de la agencia, su dirección y teléfono dependen solamente de la id de la agencia, plt BCNF
- Tour: Tenemos a id_tour como PRIMARY KEY, la descripción del tour y su precio dependen solo de ella plt BCNF.

2 Diagrama E/R



3 Creación de Tablas

- **Usuarios**

```
CREATE TABLE Usuarios(id_usuario INT PRIMARY KEY, nombre VARCHAR(50), fecha_nacimiento date, correo VARCHAR(50), nacionalidad varchar(50))
\COPY Usuarios from 'home/grupo25/data/usuario.csv' DELIMITER ','
CSV HEADER
```

- **Regiones**

```
CREATE TABLE Regiones(id_region INT PRIMARY KEY, nombre VARCHAR(50), reseña VARCHAR(1000))
\COPY Regiones from 'home/grupo25/data/region.csv' DELIMITER ','
CSV HEADER
```

Para las tablas habitaciones y hoteles se creó la siguiente tabla:

```
CREATE TABLE hoteles_habitaciones(nombre_hotel VARCHAR(50), direccion_hotel VARCHAR(200), telefono_hotel VARCHAR(15), descripcion_hotel VARCHAR(1000), estrellas INT, id_region INT, id_hotel INT, id_habitación INT, nombre_habitacion VARCHAR(100), precio_habitacion FLOAT)
\COPY hoteles_habitaciones from 'home/grupo25/data/hoteles_habitaciones.csv'
DELIMITER ',' CSV HEADER
```

- **Hoteles**

```
CREATE TABLE Hoteles2 AS
SELECT id_hotel, nombre_hotel, direccion_hotel, telefono_hotel, descripcion_hotel, estrellas
FROM hoteles_habitaciones
GROUP BY id_hotel, nombre_hotel, direccion_hotel, telefono_hotel, descripcion_hotel, estrellas
```

- **Hoteles_Región**

```
CREATE TABLE Hotel_Región AS
SELECT id_hotel, id_region
FROM hoteles_habitaciones
GROUP BY id_hotel, id_region
```

- **Habitaciones**

```
CREATE TABLE Habitaciones AS
SELECT id_habitacion, nombre_habitación, precio_habitacion
FROM hoteles_habitaciones
GROUP BY id_habitacion, nombre_habitacion, precio_habitacion
```

- **Habitación_Hotel**

```
CREATE TABLE Habitacion_Hotel AS
SELECT id_habitacion, id_hotel
FROM hoteles_habitaciones
GROUP BY id_habitacion, id_hotel
```

- **Reservas**

```
CREATE TABLE Reservas(id_reserva INT PRIMARY KEY, id_usuario
INT, id_habitación INT, fecha_inicio date, fecha_fin date)
\COPY Reservas from 'home/grupo25/data/reservas.csv' DELIMITER ','
CSV HEADER
```

Para las tablas restaurantes y productos se creó la siguiente tabla:

```
CREATE TABLE restaurante_platos (id_restaurant INT, nombre_restaurant
VARCHAR(50), dirección_restaurant VARCHAR(500), teléfono_restaurant
VARCHAR(15), descripción_restaurant VARCHAR(1000), id_región INT,
id_producto INT, nombre_producto VARCHAR(30), descripción_producto
VARCHAR (50), precio_producto FLOAT, PRIMARY KEY(id_restaurant,
id_producto))
\COPY restaurante_platos from 'home/grupo25/data/restaurante_platos.csv'
DELIMITER ',' CSV HEADER
```

- **Restaurante**

```
CREATE TABLE Restaurante AS
SELECT id_restaurant, nombre_restaurant, direccion_restaurant, telefono_restaurant,
descripcion_restaurant
FROM restaurante_platos
GROUP BY id_restaurant, nombre_restaurant, direccion_restaurant, tele-
fono_restaurant, descripcion_restaurant
```

- **Restaurante_Región**

```
CREATE TABLE Restaurante_Region AS
SELECT id_restaurant, id_region
FROM restaurante_platos
GROUP BY id_restaurant, id_region
```

- **Productos_Restaurant**

```
CREATE TABLE Productos_Restaurant AS
SELECT id_producto, id_restaurant
FROM restaurante_platos GROUP BY id_producto, id_restaurant
```

- **Productos**

```
CREATE TABLE Productos AS
```

```
SELECT id_producto, nombre_producto, descripcion_producto, precio_producto
FROM restaurante_platos GROUP BY id_producto, nombre_producto,
descripcion_producto, precio_producto
```

Para las tablas `agencia_turismo` y `tour` se creó la siguiente tabla:

```
CREATE TABLE ATR(nombre_agencia VARCHAR(30), direccion_agencia
VARCHAR(50), telefono_agencia INT, id_agencia INT, id_region INT,
id_tour INT, descripcion_tour VARCHAR(1000), precio_tour FLOAT, PRI-
MARY KEY (id_agencia, id_region, id_tour))
\COPY ATR from 'home/grupo25/data/agencia_agenciaregion_tour.csv'
DELIMITER ',' CSV HEADER
```

- **Agencia_turismo**

```
CREATE TABLE Agencia_turismo AS
SELECT id_agencia, nombre_agencia, direccion_agencia, telefono_agencia
FROM ATR
GROUP BY id_agencia, nombre_agencia, direccion_agencia, telefono_agencia
```

- **Agencia_Región**

```
CREATE TABLE Agencia_region AS
SELECT id_agencia, id_region
FROM ATR
GROUP BY id_agencia, id_region
```

- **Agencia_Tour**

```
CREATE TABLE Agencia_Tour AS
SELECT id_tour, id_agencia
FROM ATR
GROUP BY id_tour, id_agencia
```

- **Tour**

```
CREATE TABLE Tour AS
SELECT id_tour, descripcion_tour, precio_tour
FROM ATR
GROUP BY id_tour, descripcion_tour, precio_tour
```

4 Consultas

- **Consulta 1: Dado un número de región encontrar los platos de todos los restaurantes en esa región**

```
SELECT P.id_producto, P.nombre_producto, P.descripcion_producto, P.precio_producto,
PR.id_restaurant
FROM Productos AS P, productos_restaurantes AS PR
```

WHERE PR.id_producto = P.id_producto AND PR.id_restaurant IN (SELECT r.id_restaurant FROM Restaurantes AS r, Restaurantes_Region AS RR WHERE R.id_restaurant = RR.id_restaurant AND RR.id_region = i)

- **Consulta 2: Dado un número de estrellas, muestre todas las habitaciones de hoteles con más de esa cantidad de estrellas, junto al nombre del hotel en el que está**
SELECT HA.id_habitación, HA.nombre_habitación, HA.precio_habitacion
HA.id_hotel, HO.estrellas
FROM Habitaciones AS HA, Hoteles AS HO, Habitacion_Hotel AS HH
WHERE HH.id_hotel = HO.id_hotel AND HA.id_habitacion = HH.id_habitacion
AND HO.estrellas > i

- **Consulta 3: Dado un id de usuario i, una fecha de inicio a y una fecha de fin b. Entregar todas las reservas de habitaciones de ese usuario entre esas fechas**
SELECT *
FROM Reservas
WHERE id_usuario = i AND fecha_inicio > a AND fecha_fin < b

- **Consulta 4: Entregue los tours de las agencias que están solo en una región**
SELECT T.*, A.id_agencia
FROM Tour AS T, Agencia_tour AS A
WHERE T.id_tour = A.id_tour AND A.id_agencia NOT IN
(SELECT A1.id_agencia FROM Agencia_turismo AS A1, Agencia_region AS AR1, Agencia_region AS AR2 WHERE AR1.id_region != AR2.id_region
AND A1.id_agencia = AR1.id_agencia AND A1.id_agencia = AR2.id_agencia)
ORDER BY A.id_agencia

- **Consulta 5: Para cada región, entregue la habitación que ha sido reservada más veces.**
CREATE VIEW cantidad_reservas(id_habitación, número_reservas) AS
SELECT id_habitación, COUNT(id_reserva)
FROM reservas
GROUP BY id_habitación

CREATE VIEW consulta5 AS
SELECT hr.id_region, MAX(cr.numero_reservas)
FROM hoteles_region hr, hoteles_habitaciones hh, cantidad_reservas cr
WHERE hr.id_hotel = hh.id_hotel AND hh.id_habitacion = cr.id_habitacion
GROUP BY hr.id_region
ORDER BY hr.id_region;

SELECT c.id_region, hh.id_habitacion, c.max


```

FROM consulta5 c, hoteles_habitaciones hh, cantidad_reservas cr, hoteles_region hr
WHERE hr.id_region = hh.id_region AND hh.id_hotel = hr.id_hotel AND
hh.id_habitacion = cr.id_habitacion AND c.max = cr. numero_reservas
AND hr.id_region = c.id_region
ORDER BY c.id_region DESC, hh.id_habitacion;

```

- **Consulta 6: Entregue todos los usuarios que han reservado la habitación más barata en la región II.**

```

SELECT u.id_usuario, u.nombre
FROM Reservas AS r, Habitaciones AS ha, Hoteles AS ho, Usuarios AS u,
Habitacion_Hotel AS hh, Hoteles_region AS hr
WHERE hh.id_hotel = ho.id_hotel AND ha.id_habitacion = hh.id_habitacion
AND r.id_habitacion = ha.id_habitacion AND hr.id_hotel = ho.id_hotel
AND hr.id_region = 2 AND u.id_usuario = r.id_usuario AND ha.precio_habitacion
= (SELECT MIN(Habitaciones.precio_habitacion) FROM Habitaciones,
Hoteles, Habitacion_Hotel, Hoteles_region WHERE Habitacion_Hotel.id_hotel
= Hoteles.id_hotel AND Habitacion_Hotel.id_habitacion = Habitaciones.id_habitacion
AND Hoteles_region.id_hotel = Hoteles.id_hotel AND Hoteles_region.id_region
= 2)

```

- **Consulta 7: Dado un id de reserva, muestre el nombre del usuario que hizo la reserva junto al monto total que paga por esa reserva**

```

SELECT reservas.id_reserva, usuarios.nombre, habitaciones.precio_habitación
FROM reservas INNER JOIN usuarios ON (reservas.id_usuario = usuarios.id_usuario)
INNER JOIN habitaciones ON (habitaciones.id_habitación
= reservas.id_habitación)
WHERE reservas.id_reserva = i

```

- **Consulta 8: Dado un número i, entrega la i-ésima habitación más cara**

```

SELECT * FROM Habitaciones as h1
WHERE h1.precio_habitación = (SELECT h2.precio_habitación FROM
Habitaciones as h2 order by h2.precio_habitación DESC LIMIT 1 OFFSET
i-1)

```

5 Supuestos

- En la consulta 5 se muestran todas las habitaciones que tengan una cantidad de reservas igual al máximo correspondiente a su región.

6 Página Web

Todas las consultas mencionadas en la sección 4 se pueden visualizar en nuestra página web, para ello debes ingresar a: [El zelda](#) (Es clickeable). En algunas consultas se pide ingresar parámetros, favor ingresar parámetros válidos, así obtendrá la consulta que se desea. Por ejemplo, cuando se piden ID poner un número y cuando se piden fechas ponerlas como 'AAA-MM-DD'.