

# Esquema E2

Grupo: 23

## 2.1 Esquema Relacional:

- Usuarios(uid: int, username: varchar, unombre: varchar, correo: varchar, udir: varchar)
- Ciudades(cid: int, cnombre: varchar, pid: int (FK))
- Países(pid: int, pnombre: varchar, contacto: varchar)
- Destinos(did: int, cid1: int (FK), cid2: int (FK), salida: time, duracion: float, medio: varchar, max: int, precio: int)
- Tickets(tid: int, did: int (FK), uid: int (FK), asiento: int, fechac: timestamp, fechav: date)
- Hoteles(hid: int, hnombre: varchar, cid: int (FK), hdir: varchar, telefono: varchar, precio: int)
- Reservas(rid: int, hid: int (FK), uid: int (FK), fechai: date, fechat: date)

\*udir: dirección usuario; hdir: dirección hotel; cid1: id ciudad origen; cid2: id ciudad llegada; fechac: fecha de compra del ticket; fechav: fecha del vuelo; fechai: fecha del inicio de la reserva; fechat: fecha del termino de la reserva

## 2.2 Justificar Modelo:

### Dependencias Usuarios:

- uid  $\longrightarrow$  username, nombre, correo, udir
- username  $\longrightarrow$  uid, nombre, correo, udir

En este caso, se cumple con BCNF debido a que el lado izquierdo de ambas dependencias corresponde a una llave candidata.

El hecho de tener 2 atributos que determinan a todos los otros (incluyendose) es aceptable porque se elige un numero entero (ID) como PRIMARY KEY con el fin de optimizar la eficiencia de la base de datos. Por esta razón, esta tabla poseerá 2 columnas únicas, uid y username.

### Dependencias Ciudades:

- cid  $\longrightarrow$  ciudad, pid

En Ciudades, solo el id de ciudad (cid) será único, debido a que podrían existir dos ciudades con igual nombre pero en distintos países. Por esto, el nombre de ciudad, no determinará otros atributos, y por lo tanto, se encuentra en BCNF.

#### Dependencias Paises:

- $\text{pid} \rightarrow \text{pais, contacto}$
- $\text{pais} \rightarrow \text{pid, contacto}$

En este caso, se cumple con BCNF debido a que el lado izquierdo de ambas dependencias corresponde a una llave candidata.

El hecho de tener 2 atributos que determinan a todos los otros (incluyendose) es aceptable porque se elige un numero entero (ID) como PRIMARY KEY con el fin de optimizar la eficiencia de la base de datos. Por esta razón, esta tabla poseerá 2 columnas únicas, pid y pais. Este último es único debido a que no pueden existir dos paises con el mismo nombre.

#### Dependencias Destinos:

- $\text{did} \rightarrow \text{cid1, cid2, salida, duracion, medio, max, precio}$

Esta tabla cumple con BCNF, debido a que en su única dependencia, una llave primaria se encuentra definiendo el resto de los atributos.

Si bien se podría pensar que, cid1 junto a cid2 podrían definir el resto de los atributos, esto está equivocado pues podrían existir destinos entre estas dos ciudades con distinto horario de salida, duracion, medio, capacidad maxima o precio.

Similarmemente, esto ocurre para medio que se podría pensar que determinará el máximo de pasajeros, pero como ejemplo, podrían existir 2 aviones con distinta capacidad para pasajeros.

Por último, se podría considerar que cid1 junto a cid2 y medio podrían determinar la duración, pero esto es erróneo debido a que pueden existir escalas dependiendo del viaje o, siguiendo con el ejemplo anterior, un avión podría ser mas veloz que otro.

#### Dependencias Tickets:

- $\text{tid} \rightarrow \text{did, uid, asiento, fechac, fechav}$

La tabla Tickets cumple con BCNF, debido a que solo una PRIMARY KEY define otros atributos. Esta tabla posee una única dependencia pues un usuario podría comprar múltiples tickets para un mismo destino, si este está comprando para un acompañante o si quiere repetir ese destino en el futuro.

Además, podría suceder que un mismo usuario haya comprado 2 tickets con atributos iguales pero para distintos destinos.

Por último, se suma que podría existir un destino para el cual se venden asientos en 2 o mas aviones (u otro dependiendo del medio) que saldrán en la misma fecha. Entonces, el did junto al asiento y a la fecha del vuelo no determinaran al usuario.

#### Dependencias Hoteles:

- $\text{hid} \rightarrow \text{hnombre}, \text{cid}, \text{hdir}, \text{telefono}, \text{precio}$
- $\text{telefono} \rightarrow \text{hid}, \text{hnombre}, \text{cid}, \text{hdir}, \text{precio}$
- $\text{hdir} \rightarrow \text{hid}, \text{hnombre}, \text{cid}, \text{telefono}, \text{precio}$

Aquí, tanto hid como telefono y hdir corresponden a llaves candidatas, por lo tanto, se cumple con BCNF.

Se elige hid como PRIMARY KEY para optimizar la eficiencia de la base de datos.

Se asume que pueden existir 2 hoteles con el mismo nombre en una misma ciudad, que existe un único precio por hotel, que un hotel tiene un único contacto telefónico y que, claramente, dos o más hoteles no pueden compartir una misma dirección.

#### Dependencias Reservas:

- $\text{rid} \rightarrow \text{hid}, \text{uid}, \text{fechai}, \text{fecha}$

Reservas cumple con BCNF porque solo rid (PRIMARY KEY) define al resto de los atributos.

Además, es importante mencionar que un usuario podría realizar múltiples reservas en el mismo hotel si desea visitar este en otras fechas, e incluso podría tener múltiples reservas en el mismo hotel y mismas fechas de inicio y termino si es que va acompañado.