



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC 2413 — Bases de datos — 1' 2020

Proyecto - Entrega 2

Declaración de esquema

Entidades

`cuentas(username:text PK, nombreusuario:text, direccionusuario:text)`

- username, *username* del usuario.
- `nombreusuario`, Nombre completo de la persona que abrió la cuenta.
- `direccionusuario`, direccion de la persona que abrió la cuenta.

`usuarios(uid:text PK, username:text FK, correo:text)`

- uid, uid del usuario.
- `username`, *username* de la persona que abrió la cuenta.
- `correo`, correo de la persona que abrió la cuenta.

`ciudades(cid:int PK, nombreciudad:text, pid:int FK)`

- cid, id de la ciudad.
- `nombreciudad`, nombre de la ciudad.
- `pid`, id del país al que pertenece la ciudad.

`paises(id:int PK, nombrepais:text, fonocontacto:text)`

- pid, id del país.
- `nombrepais:text`, nombre del país.
- `fonocontacto`, telefono de contacto de la empresa en ese país.

datos_viaje(Did:int PK, cid_origen:int FK, cid_destino:int FK, horasalida:time, duracion:int ,medio:text, capacidad:int, precio:int)

- Did, id de los datos del viaje.
- cid_origen, id del ciudad de origen.
- cid_destino, id de la ciudad de destino.
- horasalida, hora de partida del viaje.
- duracion, tiempo de duración del viaje .
- medio, vehículo que se utilizará para el transporte de pasajeros.
- capacidad, capacidad del tipo de transporte.
- precio, precio del viaje.

tickets_comprados(tid:int PK , Did:int FK, uid:int FK, asiento:int, fechacompra:TIMESTAMP, fechaviaje:date)

- tid, id del *ticket*
- Did, id de los datos del viaje a realizar.
- uid , id del usuario, en este caso el comprador.
- asiento, número de asiento.
- fechacompra, fecha en que se realizó la compra del ticket.
- fechaviaje, fecha de salida del viaje

reservas(rid:int PK , hid:int FK, uid:int FK, fechainicio:date, fechatermino:date)

- rid, id de la reserva.
- hid, id del hotel donde se realizó la reserva.
- uid, id del usuario que realizo la reserva.
- fechainicio, fecha de ingreso a la reserva.
- fechatermino, fecha de salida de la reserva.

hoteles(hid:int PK, nombrehotel:text, direccionhotel:text, telefono:text, cid:int FK)

- hid, id del hotel.
- nombrehotel, nombre del hotel.
- direccionhotel, dirección del hotel.

- telefono, telefono de contacto del hotel.
- cid, id de la ciudad donde se encuentra el hotel.

costo_hotel(hid:int PK, precionoche:int)

- hid, id del hotel.
- precionoche, precio por noche en el hotel.

Diagrama E/R

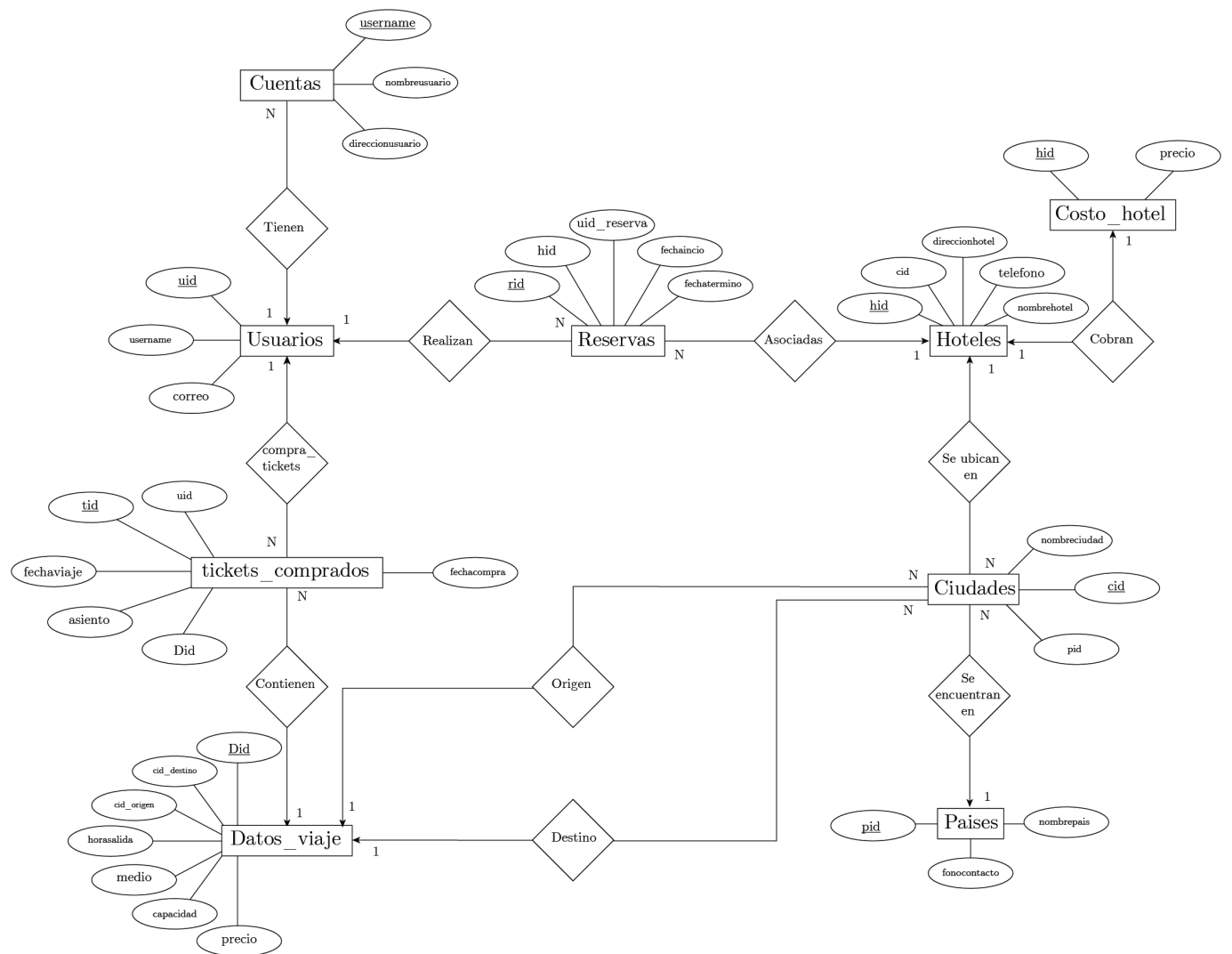


Figura 1: Diagrama E/R modelado

Justificación BNF o 3NF

i. `cuentas(username:text PK, nombreusuario:text, direccionusuario:text)`

$\text{username} \rightarrow \text{nombreusuario}$

$\text{username} \rightarrow \text{direccionusuario}$

Este esquema está en BNF.

ii. `usuarios(uid:text PK, username:text FK, correo:text)`

$\text{uid} \rightarrow \text{username}$

$\text{uid} \rightarrow \text{correo}$

$\text{username}, \text{correo} \rightarrow \text{uid}$

Este esquema se encuentra en 3NF, ya que notamos que username y correo son una llave minimal.

iii. `ciudades(cid:int PK, nombreciudad:text, pid:int FK)`

$\text{cid} \rightarrow \text{nombreciudad}$

$\text{cid} \rightarrow \text{pid}$

El esquema de ciudades se encuentra en BCNF.

iv. `paises(id:int PK, nombrepais:text, fonocontacto:int)`

$\text{pid} \rightarrow \text{nombrepais}$

$\text{pid} \rightarrow \text{fonocontacto}$

$\text{nombrepais}, \text{fonocontacto} \rightarrow \text{pid}$

El esquema de paises, está en 3NF, ya que si sabemos el nombre del pais y el teléfono podemos conocer el pid, sin embargo, pid es la llave primaria.

v. `datos_viaje(Did:int PK, cid_origen:int FK, cid_destino:int FK, horasalida:time, duracion:time, medio:text, capacidad:int, precio:int)`

$\text{Did} \rightarrow \text{cid_origen}, \text{cid_destino}, \text{horasalida}, \text{duracion}, \text{medio}, \text{capacidad}, \text{precio}$

Este esquema está en BCNF, ya que Did determina toda la información del lado derecho y no existen llaves minimales.

vi. `tickets_comprados(tid:int PK , Did:int FK, uid:int FK, asiento:int, fechacompra:TIMESTAMP, fechaviaje:date)`

$tid \rightarrow \text{Did, uid, asiento, fechacompra, fechaviaje}$

Este esquema está en BCNF, ya que `tid` determina toda la información del lado derecho y no existen llaves minimales.

vii. `reservas(rid:int PK , hid:int FK, uid:int, fechainicio:date, fechatermino:date)`

$rid \rightarrow \text{hid, uid, fechainicio, fechatermino}$

Este esquema está en BCNF, ya que `rid` determina toda la información del lado derecho y no existen llaves minimales.

viii. `hoteles(hid:int PK, nombrehotel:text, direccionhotel:text, telefono:text, cid:int FK)`

$hid \rightarrow \text{nombrehotel, direccionhotel, telefono, cid}$

Este esquema está en BCNF, ya que `hid` determina toda la información del lado derecho y no existen llaves minimales.

ix. `costo_hotel(hid:int PK, precionoche:int)`

$hid \rightarrow \text{precionoche}$

Este esquema está en BCNF, ya que `hid` determina el precio.

Consultas SQL implementadas

Consulta 1

```
SELECT username, correo
FROM usuarios
```

Consulta 2

```
SELECT nombreciudad, nombrepais
FROM ciudades natural join paises
WHERE nombrepais ~* '$pais'
```

Consulta 3

```
SELECT nombrepais, username, fechainicio
FROM usuarios natural join reservas natural join hoteles natural join ciudades natural join paises
WHERE username ~* '$username' AND fechainicio <current_date
```

Consulta 4

```
”SELECT uid, precio
FROM tickets_comprados natural join datos_viaje natural join usuarios
WHERE uid=$id
```

```
SELECT SUM(precio)
FROM tickets_comprados natural join datos_viaje natural join usuarios
WHERE uid=$id
```

Consulta 5

```
SELECT uid, username, fechainicio, fechatermino, nombrehotel
FROM usuarios natural join reservas natural join hoteles
WHERE fechainicio >='2020-01-01' and fechatermino <='2020-03-31'
```

Consulta 6

```
SELECT uid, username, SUM(precio)
FROM usuarios natural join tickets_comprados natural join datos_viaje
WHERE fechacompra >='$fecha_inicio 00:00:00' and fechacompra <= '$fecha_termino 23:59:59'
GROUP BY uid
```