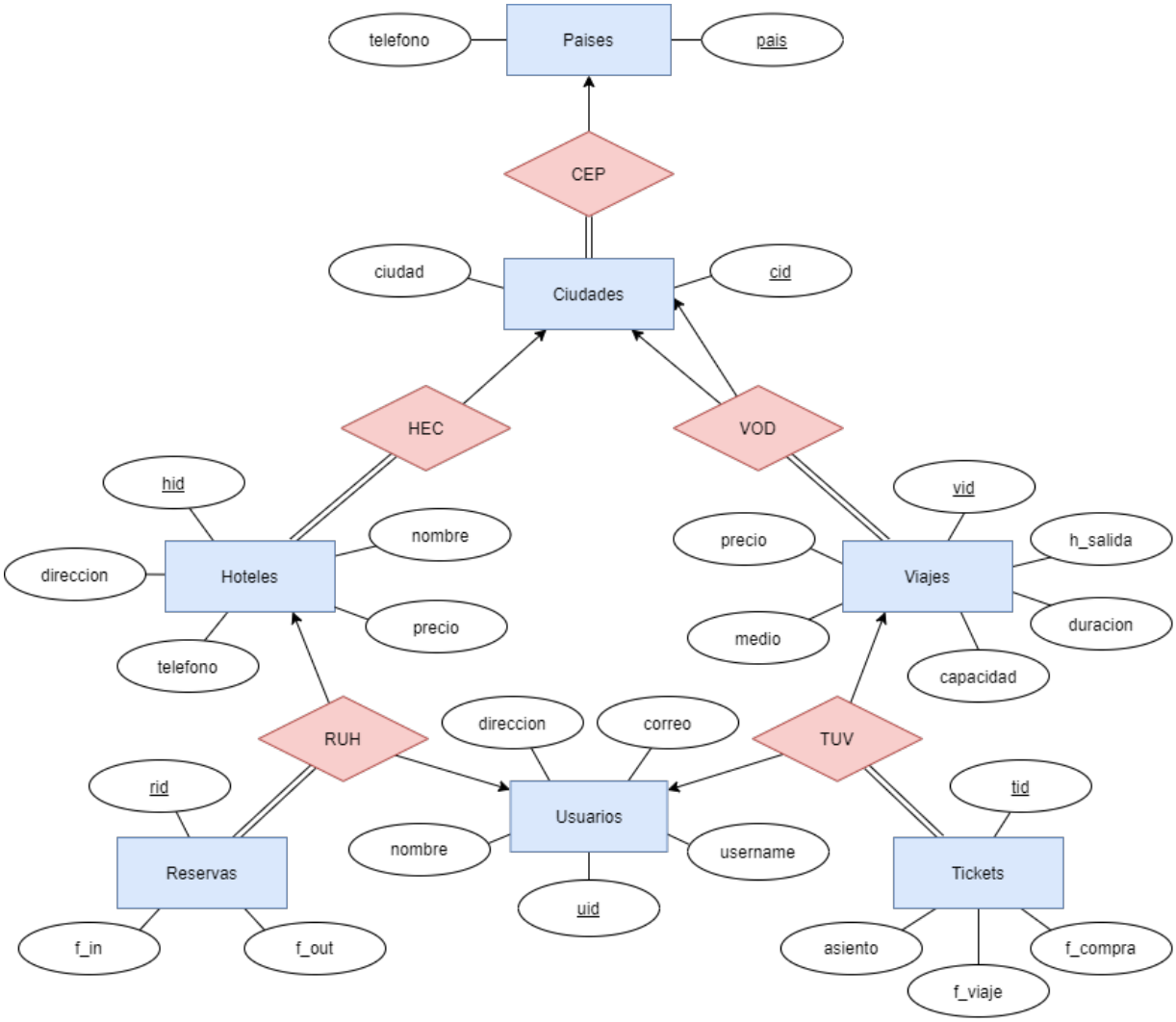
Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2413-2 — Bases de Datos — 1' 2020

Integrantes: Diego Bustamante y Lucas Muñoz

Entrega 2 - Reporte

1.- **Diagrama Entidad/Relación** de nuestra base de datos:



2.- **Esquema** de nuestra base de datos. Las entidades tienen nombres completos y las relaciones abreviaciones de las entidades que relaciona excepto: CEP = Ciudad En Pais, HEC = Hotel En Ciudad y VOD = Viajes, Origen y Destino. Todos los atributos que se repiten en varias tablas son Primary Key de la única entidad en la que participan, en el resto (en las relaciones) son **Llaves Foráneas**. Excepto en VOD que para no repetir un atributo se optó por usar otro nombre, pero o_cid y d_cid son cid como llave foránea.

Quisimos añadir 2 restricciones adicionales. Primero, en las relaciones usamos como llave solo el atributo n en las n:1 para forzar dicha relación. Segundo, modelamos Reservas y Tickets como entidades para permitir que un usuario pueda reservar varias veces el mismo hotel (por eso rid y tid son PK en RUH y TUV respectivamente), pero su existencia solo se permite si se le puede asociar un usuario y un hotel en el caso de las Reservas o un usuario y un viaje en el caso de los Tickets.

1. Usuarios(uid: int, username: string, nombre: string, correo: string, direccion: string)
2. Ciudades(cid: int, ciudad: string)
3. Paises(pais: string, teléfono: string)
4. CEP(cid: int, pais: string)
5. Viajes(vid: int, h_salida: time, duracion: int, medio: string, capacidad: int, precio: float)
6. VOD(vid: int, o_cid: int, d_cid: int)
7. Tickets(tid: int, asiento: int, f_compra: date, f_viaje: date)
8. TUV(tid: int, uid: int, vid: int)
9. Hoteles(hid: int, nombre: string, direccion: string, telefono: string, precio: float)
10. HEC(hid: int, cid: int)
11. Reserva(rid: int, f_in: date, f_out: date)
12. RUH(rid: int, uid: int, hid: int)

3.- **Justificación**: Cada tabla tiene su propia y única dependencia funcional. Estas dependencias logran las restricciones y permiten las relaciones descritas anteriormente. En cada una de las no triviales, el conjunto que determina al resto de los atributos es llave primaria. Además, no hay subconjuntos que puedan reducir el tamaño del conjunto de llaves, todas son minimales. Entonces, nuestra base de datos está en **BCNF**.

1. uid $\rightarrow$ username, nombre, correo, direccion
2. cid $\rightarrow$ ciudad
3. pais $\rightarrow$ telefono
4. cid $\rightarrow$ pais

5. vid → h_salida, duracion, medio, capacidad, precio

6. vid → o_cid, d_cid

7. tid → asiento, f_compra, f_viaje

8. tid → uid, vid

9. hid → nombre, direccion, telefono, precio

10. hid → cid

11. rid → f_in, f_out

12. rid → uid, hid

4-. **Consultas** (se utiliza el operador '∼*' de Psql para hacer consultas regex (autocompletado) case-insensitive en el caso de los inputs con strings):

1. SELECT username, correo
   FROM Usuarios;

2. SELECT ciudad
   FROM Ciudades NATURAL JOIN CEP
   WHERE CEP.pais ∼* '.*PAIS INPUT.*';

3. SELECT DISTINCT CEP.pais
   FROM Usuarios AS U, RUH, Hoteles AS H, HEC, CEP
   WHERE U.uid = RUH.uid AND H.hid = RUH.hid AND H.hid = HEC.hid AND HEC.cid = CEP.cid
   AND U.username ∼* '.*USERNAME INPUT.*';

4. SELECT SUM(V.precio)
   FROM Usuarios AS U, TUV, Tickets AS T, Viajes AS V
   WHERE U.uid = TUV.uid AND TUV.tid = T.tid AND TUV.vid = V.vid AND U.uid = 'UID INPUT'
   AND T.f_compra ≤ CURRENT_DATE;

5. SELECT U.uid, U.nombre, R.f_in, R.f_out, H.nombre
   FROM Usuarios AS U, RUH, Reservas AS R, Hoteles AS H
   WHERE U.uid = RUH.uid AND R.rid = RUH.rid AND H.hid = RUH.hid AND '2020/01/01' ≤ R.f_in
   AND R.f_out ≤ '2020/03/31';

6. SELECT U.uid, U.nombre, SUM(V.precio)
   FROM(SELECT tid FROM Tickets WHERE 'INICIO INPUT' ≤ Tickets.f_compra
   INTERSECT SELECT tid FROM Tickets WHERE Tickets.f_compra ≤ 'FIN INPUT') AS T,
   Usuarios AS U, TUV, Viajes AS V
   WHERE U.uid = TUV.uid AND V.vid = TUV.vid AND T.tid = TUV.tid
   GROUP BY U.uid, U.nombre

5.- **Supuestos**:

1. En el archivo 'usuarios-reservas-hoteles.csv' descartamos el dato '4,70,París' porque la correspondencia entre los datos y las columnas no tiene sentido.

2. En el archivo 'usuarios-reservas-hoteles.csv' consideramos las 5 líneas finales como personas que hay que incluir en la base de datos pero que posiblemente no han reservado un hotel.

3. En la consulta 4 consideramos los tickets comprados hasta (incluyendo) la fecha que se realiza la consulta.