

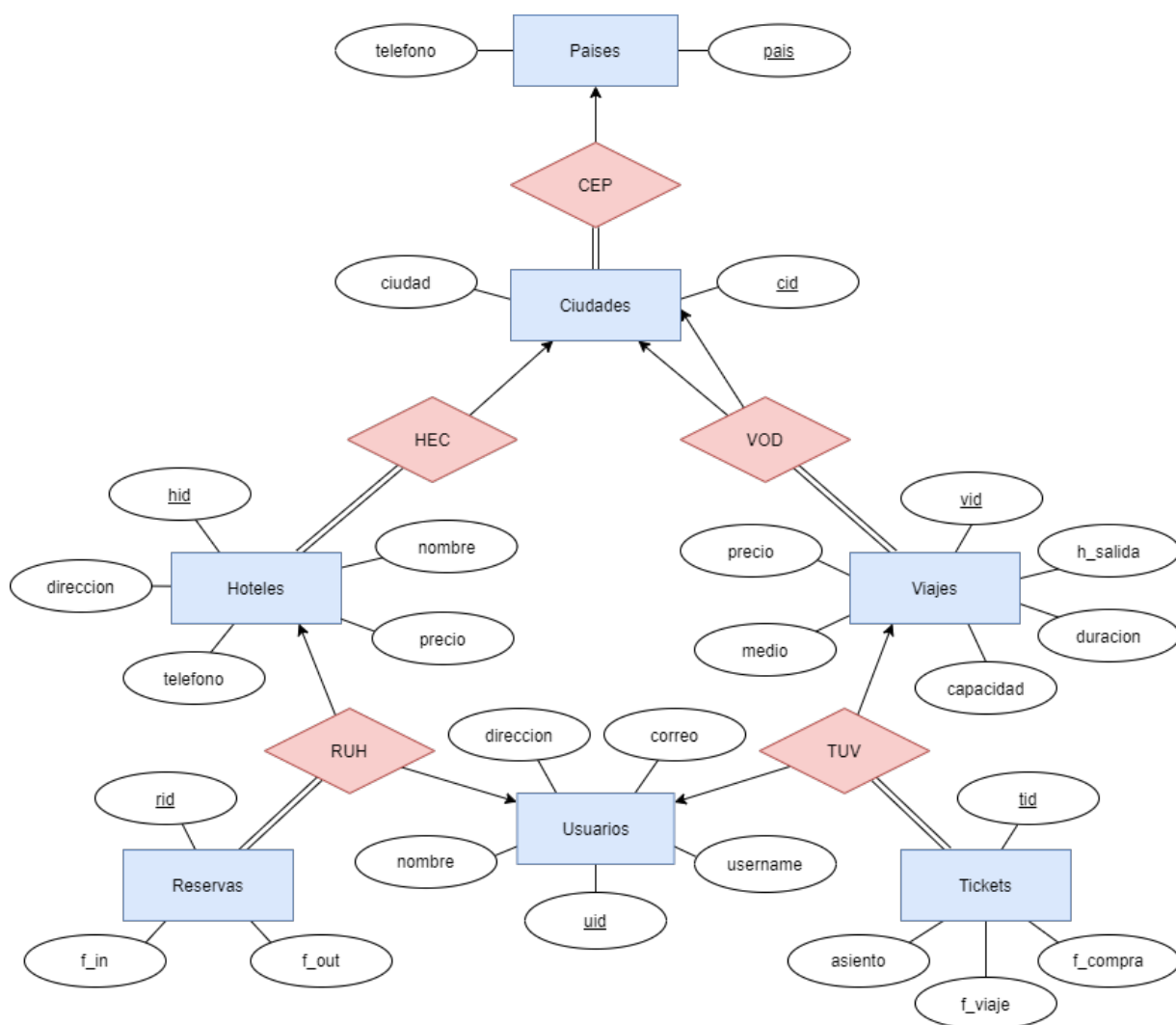


IIC2413-2 — Bases de Datos — 1' 2020

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, Reservas y Tickets son entidades para permitir que un usuario pueda reservar varias veces el mismo hotel (por eso rid y tid son PK), pero su existencia solo se permite si se le puede asociar un usuario y un hotel.

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 restricciones y permiten 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 esta en BCNF.

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

5. vid  $\rightarrow$  h\_salida, duracion, medio, capacidad, precio
6. vid  $\rightarrow$  o\_cid, d\_cid
7. tid  $\rightarrow$  asiento, f\_compra, f\_viaje
8. tid  $\rightarrow$  uid, vid
9. hid  $\rightarrow$  nombre, direccion, telefono, precio
10. hid  $\rightarrow$  cid
11. rid  $\rightarrow$  f\_in, f\_out
12. rid  $\rightarrow$  uid, hid

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

1. SELECT username, correo  
FROM Usuarios;
2. SELECT ciudad AS "Ciudades"  
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(Viajes.precio) AS "Total gastado"  
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  $\leq$  CURRENT\_DATE;
5. SELECT U.uid AS "Identificador", U.nombre AS "Nombre del usuario", R.f\_in AS "Fecha de inicio",  
R.f\_out AS "Fecha de término", H.nombre AS "Nombre del Hotel"  
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'  $\leq$  R.f\_in  
AND R.f\_out  $\leq$  '2020/03/31';

6. SELECT Usuarios.uid AS “Identificador”, Usuarios.nombre AS “Nombre del usuario”, SUM(Viajes.precio) AS “Total gastado”  
 FROM(SELECT tid FROM Tickets WHERE ‘INICIO INPUT’  $\leq$  Tickets.f\_compra  
 INTERSECT SELECT tid FROM Tickets WHERE Tickets.f\_compra  $\leq$  ‘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 “Identificador”, “Nombre del usuario”

6.- Suposiciones:

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