

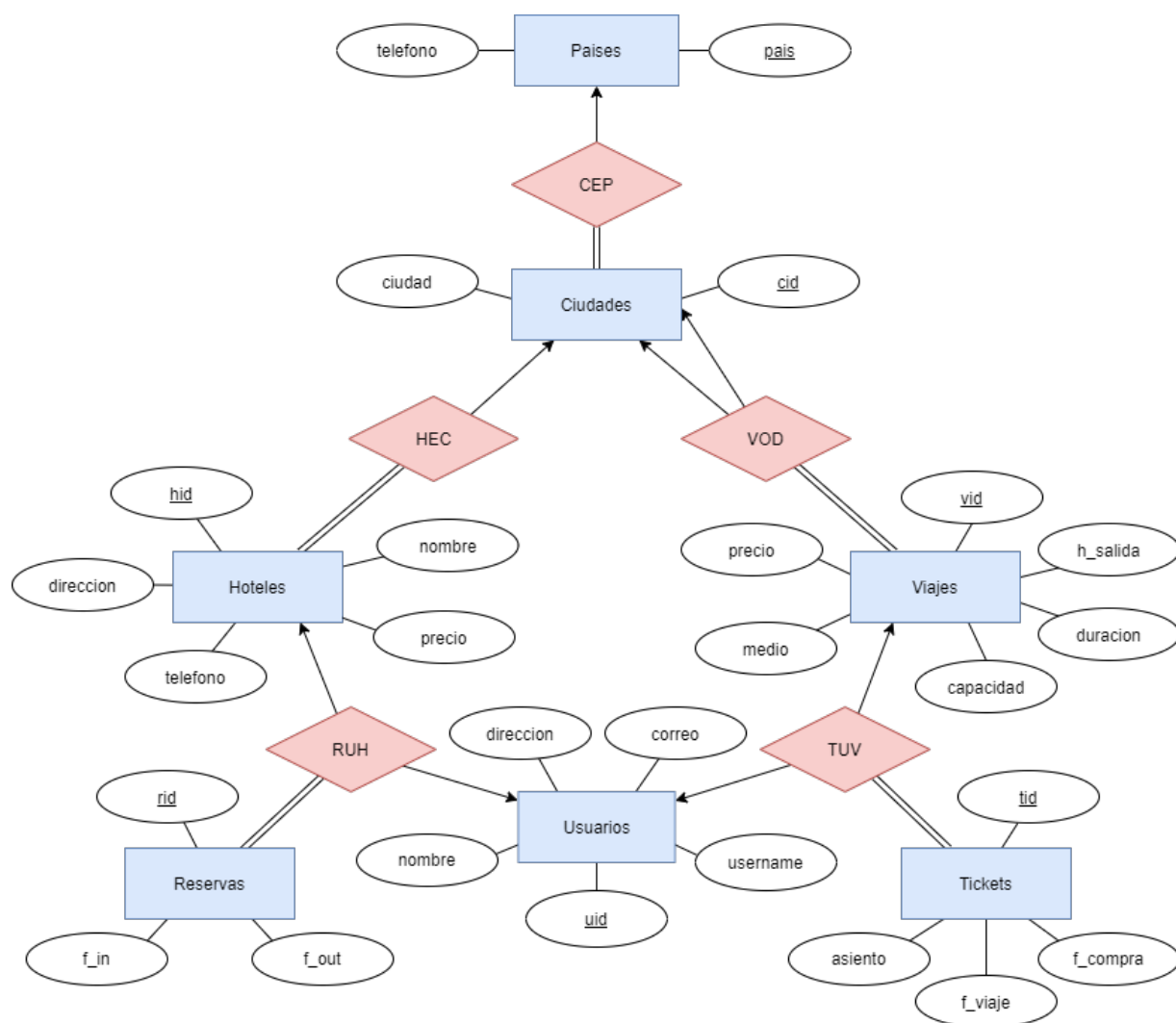


IIC2413-2 — Bases de Datos — 1' 2020

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. Países(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:

1. SELECT username, correo
FROM Usuarios;
2. SELECT ciudad
FROM ciudades, cep
WHERE ciudades.cid = cep.cid AND cep.pais = 'PAIS INPUT';
3. SELECT CEP.pais
FROM Usuarios, RUH, Hoteles, HEC, Ciudades, CEP
WHERE Usuario.username = 'USERNAME INPUT';
4. SELECT Usuario.uid, SUM(Viajes.precio)
FROM Usuarios, TUV, Tickets, Viajes
WHERE Usuarios.uid = 'UID INPUT' AND Tickets.f_compra \leq HOY
GROUP BY Usuario.uid, SUM(Viajes.precio);
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 R.f_in \geq '2020/01/01', R.f_out \leq '2020/03/31';
6. SELECT Usuarios.uid, Usuarios.nombre, SUM(Viajes.precio) AS Total gastado
FROM (SELECT * FROM Tickets WHERE Tickets.f_compra \geq 'INICIO INPUT'
INTERSECT SELECT * FROM Tickets WHERE Tickets.f_compra \leq 'FIN INPUT') AS Foo,
Usuarios, TUV, Viajes
GROUP BY Usuarios.uid, Usuarios.nombre, Total gastado;