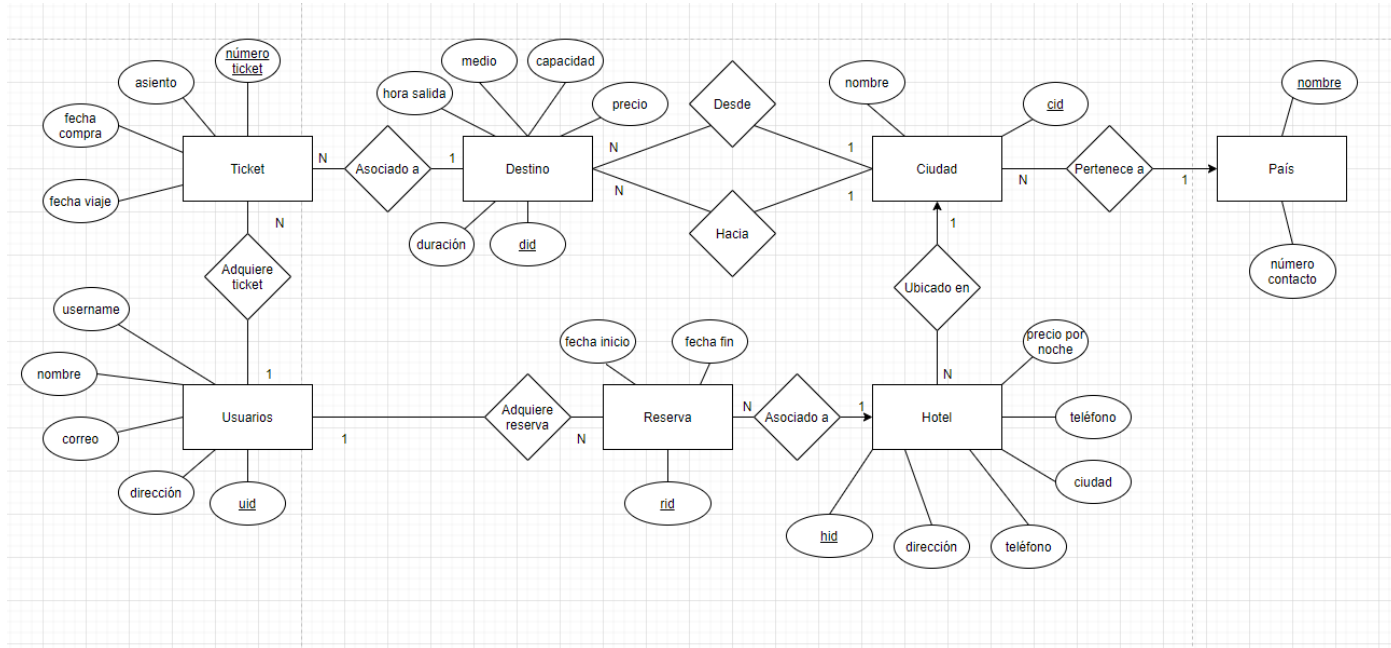


Entrega 2: grupo 15

Modelo Entidad/Relación



Modelo relacional

Usuario (uid: int, username: text UNIQUE, nombre: text, correo: text, dirección: text)

Ciudad (cid: int, nombre: text, País.nombre:text FK)

País (nombre: text, número contacto: text)

Destino (did: int, hora salida: time, duración: float, medio: text, capacidad: int, precio: int, ciudad origen: int FK(ciudad.did), ciudad destino: int FK(ciudad.did))

Ticket (número ticket: int, asiento: int, fecha compra: timestamp, fecha viaje: date, Destino.did: int FK, Usuario.uid: int FK)

Hotel (hid: int, nombre: text, dirección: text, Ciudad.cid:int FK, teléfono: text, precio por noche: int)

Reserva (rid: int, fecha inicio: date, fecha fin: date, username: text, Hotel.hid: int FK, Usuario.uid: int FK)

Justificación

Usuario: Cada uid es una identificación única de cada usuario, y se le asigna apenas se registra, donde además indica cada parámetro(username, nombre, correo y dirección.. Al tener el uid sabremos qué usuario es, por lo que sabremos su información.El username es del tipo “unique”, para asegurarse de que en la columna username no existan dos valores iguales.

uid → username, nombre, correo, dirección

username →uid, nombre, correo, dirección

También podemos decir que al saber el username del usuario, sabremos todos sus datos (ya que es único). Aún así, no dejará de ser llave uid, y seguirá cumpliendo con 3NF por la razón de que esta dependencia define al conjunto completo y no un subconjunto, entonces no se podrán separar los datos en más tablas. Además, estas 2 columnas (uid, username) entregan información, por lo que se deben mantener en la base de datos, y no es repetición de información.

Ciudad: cada ciudad tiene un código único, es por esto que teniendo el código conozco su nombre y el país al que pertenece, dándose la relación:

cid → nombre, País.nombre

País: Cada país tiene un nombre y no existen dos con el mismo nombre. Además la agencia tiene solamente un número de contacto por país, es por esto que da la relación:

nombre → número contacto

Destino: El identificador del destino (did) es único y está asociado a cada destino, por lo que conociéndolo se tiene la información exacta de la hora de salida, la duración, el medio de transporte, la capacidad, el precio y de la ciudad de origen y destino. Además todos los otros datos son independientes entre sí, por lo que no será posible generar dependencias de subconjuntos.

did → hora salida, duración, medio, capacidad, precio, ciudad origen ,ciudad destino

Ticket: Al comprar un ticket, éste posee una identificación exclusiva para evitar falsificaciones, es por esto que tiene un número de ticket único, que está asociado a un

asiento, una fecha de viaje, a un destino, a un usuario y a la fecha en que lo compró, para asegurar que un ticket permita el ingreso a sólo una persona, en el asiento, fecha, hora y destino comprado.

número ticket → asiento, fecha compra, fecha viaje, Usuario.uid, Destino.did

Además, vemos que no existe otra dependencia, ya que podrían haber 2 viajes con igual id de destino, y que un usuario haya comprado el mismo número de asiento para estos 2 viajes en la misma fecha.

Hotel: Cada hotel posee un identificador único (hid), por lo que teniendo el hid, se conoce la dirección, ciudad, teléfono y precio por noche del hotel buscado.

hid → dirección, Ciudad.cid, teléfono, precio por noche

Se puede ver que no hay más dependencias, ya que pueden existir varios hoteles con el mismo nombre (por ejemplo de la misma cadena) que incluso están ubicados en la misma ciudad, cobrando lo mismo y que hasta podrían estar ubicados en distintos lugares pero con un nombre de dirección igual.

Reserva: Cada reserva tiene un identificador exclusivo de esa reserva, por lo que al conocerlo se obtiene el usuario que hizo la reserva, el hotel donde la hizo y la fecha de inicio y término de la reserva.

rid → fecha inicio, fecha fin, Usuario.uid, Hotel.hid

Aquí claramente podemos ver que no hay más dependencias, ya que un usuario puede hacer más de 1 reserva en el mismo hotel en las mismas fechas. Por lo que para diferenciar esto, es necesario el rid.

Debido a lo anterior, como en cada relación solo las llaves primarias definen a todos demás atributos, se observa que el diagrama se encuentra en 3NF.

Consultas

- 1) SELECT username, correo FROM usuarios;
- 2) SELECT nombreciudad FROM ciudades WHERE nombrepais='pais_ejemplo';
- 3) USERNAME='usuario_ejemplo'
SELECT nombrepais FROM ciudades, hoteles, reservas, usuarios WHERE
usuarios.username='usuario_ejemplo' AND reservas.uid=usuarios.uid AND
reservas.hid=hoteles.hid AND hoteles.nombreciudad=ciudades.cid AND
reservas.fechainicio<now() GROUP BY nombrepais;
- 4) UID='1'

```
SELECT SUM(destinos.precio) FROM destinos, tickets WHERE  
destinos.did=tickets.did AND tickets.uid=1 GROUP BY tickets.uid AND  
tickets.fechacompra<=now();
```

- 5)

```
SELECT usuarios.uid, usuarios.nombreusuario, reservas.fechainicio,  
reservas.fechatermino, hoteles.nombrehotel FROM usuarios, reservas, hoteles  
WHERE usuarios.uid=reservas.uid AND hoteles.hid=reservas.hid AND  
reservas.fechainicio>='2020-01-01' AND reservas.fechatermino<='20  
20-03-31';
```

- 6) Fecha 1:F1, Fecha 2:F2

```
SELECT usuarios.uid, SUM(destinos.precio) FROM usuarios, tickets, destinos  
WHERE usuarios.uid=tickets.uid AND tickets.did=destinos.did AND  
tickets.fechacompra<='F1 AND tickets.fechacompra>='F2' GROUP BY usuarios.uid;
```