



Pontificia Universidad Católica de Chile
Facultad de Ingeniería
Departamento de Ciencias de la Computación

Entrega 2

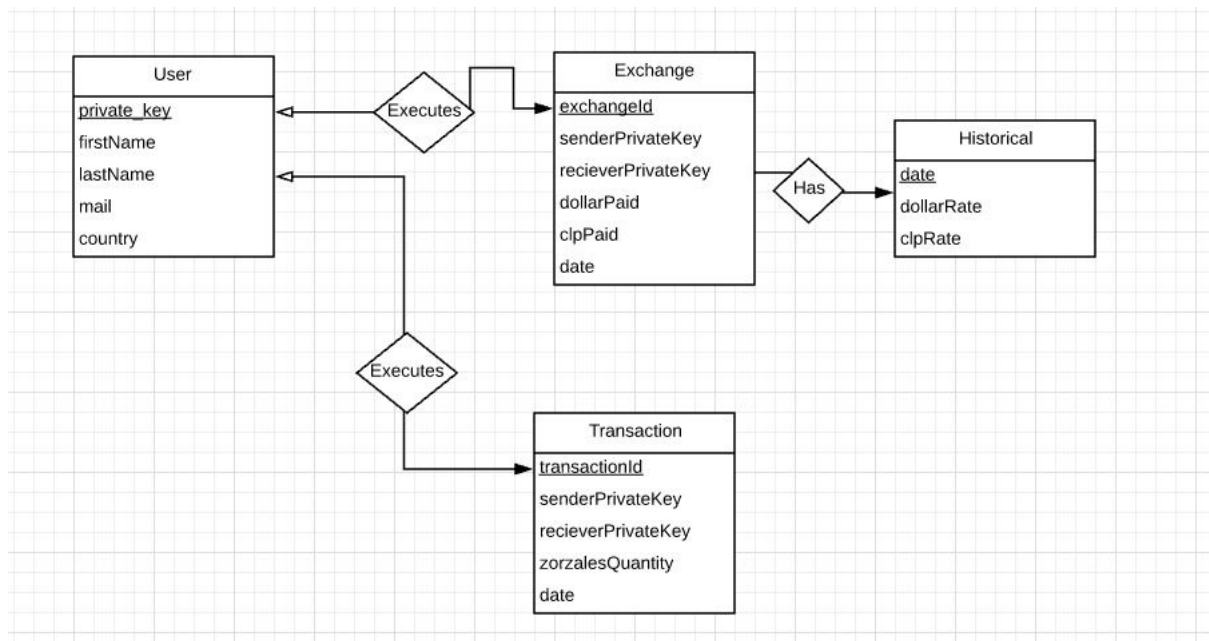
Curso: Bases de Datos
Profesor: Juan Reutter

Nombres Integrantes: Andrés Jahr
Milan Rafaeli

16/04/2018

1. Modelación

1.1 Diagrama E/R



Se crearon las entidades User, Exchange, Transaction e Historical. Las relaciones se explican como que un User(usuario) ejecuta muchos Exchange(interCambios) los cuales tienen también un Historical lo que quiere decir que para cada intercambio existe una fecha en la cual se tiene un valor de dólar por zorzal (dollarRate) y un valor de clp por zorzal(clpRate) específico. Por otro lado, un User también ejecuta muchas transacciones.

1.2 Tablas y Dependencias

User(private_key, firstName, lastName, mail, country)

Se generó una tabla User que representa al usuario del programa, bajo los atributos private_key(llave privada) siendo este la llave de la tabla, firstName(primer nombre del usuario), lastName(apellido del usuario), mail(mail del usuario), country(país del usuario). No se generó ninguna dependencia funcional(aparte de la llave), ya que la única forma de saber el resto de los atributos es con la llave que sería la private_key.

Exchange(exchangeId, senderPrivateKey, receiverPrivateKey, dollarPaid, clpPaid, date)

Se genero una tabla Exchange que representa la compra y venta de zorzales por dólares o CLP. Tiene los atributos exchangeId siendo este el número de identificación de la transacción por lo que es llave, senderPrivateKey que representa el private_key del usuario que está enviando el dinero para recibir los zorzales, receiverPrivateKey que representa el private_key del usuario que está recibiendo el dinero para enviar los zorzales, dollarPaid que sería la cantidad de dólares que está pagando el usuario, clpPaid la cantidad en clp que está pagando el usuario ,date(fecha) la fecha del momento del intercambio. No hay dependencia funcional(aparte de la llave) debido a que solo con la llave, el exchangeId se pueden saber el resto de los atributos.

Transaction(transactionId, senderPrivateKey, receiverPrivateKey, zorzalesQuantity, date)

Se generó una tabla Transaction que representa el envío y recibo de zorzales entre usuarios. El transactionId es llave y es el identificador de esta transacción, senderPrivateKey es el private_key del usuario que envía los zorzales, receiverPrivateKey es el private_key del usuario que recibe los zorzales, zorzalesQuantity es la cantidad de zorzales que son enviados, date es la fecha en que se realizó la transacción. No hay dependencia funcional(aparte de la llave) ya que solo con la llave, el transactionId, se pueden saber el valor de los otros atributos de la tabla.

Historical(date, dollarRate, clpRate)

Se generó una tabla Historical con el fin de representar cuántos dolares o pesos clp se necesitan en un día específico para comprar 1 zorzal. En este caso date es la llave e indica el día en que se está midiendo el valor del zorzal, dollarRate sería cuántos dólares necesito para comprar un zorzal y clpRate cuántos clp necesito para comprar un zorzal. No hay dependencia funcional mas alla de la llave ya que date es la única que hace una relación directa con los otros atributos.

Nuestro esquema se encontrarán en BCNF debido a que toda dependencia funcional no trivial es llave, es decir, todas nuestras tablas están en BCNF por ende nuestro esquema también.

1.3 Esquema Final

User

privateKey (primary_key, int not null)
firstName (varchar(20) not null)
lastName (varchar(20) not null)
mail (text not null)
country (text not null)

Exchange

exchangeId (serial not null, primary key)
senderPrivateKey (int not null, foreign key(privateKey) references User)
recieverPrivateKey (int not null, foreign key(privateKey) references User)
dollarPaid (real not null default 0)
clpPaid (real not null default 0exc)
date (timestamp default current_timestamp)

Historical

Date (timestamp not null, primary key)
dollarRate (real not null)
clpRate (real not null)

Transaction

transactionId (serial not null, primary key)
senderPrivateKey (int not null, foreign key(privateKey) references User)
recieverPrivateKey (int not null, foreign key(privateKey) references User)
zorzaesQuantity (real not null)
date (timestamp default current timestamp)

1.4 Justificación BCNF

Nuestro modelo se encuentra en BCNF ya que para cada tabla, la única dependencia funcional (no trivial) que tienen es su propia llave. Como todas las tablas están en BCNF el esquema también.

1.5 Comparación Entrega 1

1. Sacamos el atributo "Wallet" del usuario porque era redundante ya que la cantidad de zorzales es posible calcularla a partir de las tablas Exchange y Transaction. Esto debido a que con una consulta se puede podría sumar y restar las intercambios o transacciones de zorzales de un usuario y obtener su monto actual. Por ende no se encontró necesario tenerlo como un atributo aparte.
2. Sacamos el atributo zorzalesQuantity en la tabla Exchange porque es posible obtener este valor de los atributos amountPaid (con esto sabemos cuanto dinero se está cambiando), isDollar(sabemos si es en dolar o clp el valor), date(sabemos el valor del dolar o clp por zorzal y podemos hacer la conversión). Esto para eliminar redundancia.
3. Sacamos userId ya que teniamos una llave única privateKey, por ende era redundante e innecesario tener un userId como primary key. Solo con uno de los dos bastaba para tener dependencia funcional sobre el resto y funcionar como llave.
4. Cambiamos el atributo isDollar de Exchange por dollarPaid y clpPaid. Esto para hacer más flexibles nuestras consultas. El objetivo de isDollar era ocuparlo como un booleano que nos dijera si el amountPaid era en dólares o en clp. Con dollarPaid y clpPaid se rellena con el valor que se quiere cambiar en dólares o en pesos o ambos al mismo tiempo. Por default igual agregamos que si no ponemos ningún valor a clpPaid o a dollarPaid este se reemplazará por un cero.