

EVALUACIÓN MÓDULO 3

Cuestiones a tener en cuenta al diseñar la base de datos:

1. ¿Qué tipo de relación existe entre las entidades Usuario y Transacción?

R: Existe una relación 1:N ya que un usuario puede realizar muchas transacciones

2. ¿Cómo podemos asegurarnos de que cada transacción está asociada al usuario emisor y receptor correctos?

R: A través del user_id, cada usuario tiene una llave primaria user_id y esta es una llave foránea en transacciones, tenemos una para el emisor sender_user_id y una para el receptor receiver_user_id

3. ¿Cómo manejaremos la relación entre la entidad Transacción y la entidad Moneda?

R: Entidad Moneda tiene una llave primaria que la identifica, y esta es una llave foránea en transacción. Entonces cada transacción está asociada a una única moneda.

4. ¿Cuál debería ser la clave primaria de cada entidad para garantizar la unicidad?

R: En usuario user_id, en Moneda currency_id, en transacciones transaccion_id.

Estas son tipo entero, auto incrementables no nulos.

Como transacciones depende de las otras dos tablas, primero se crea la tabla usuario y moneda y luego la tabla transacciones.

5. ¿Qué restricciones debemos aplicar para mantener la integridad de los datos?

R: Las tablas deben tener una llave primaria única, la relación entre las tablas se realiza a través de las claves foráneas, también se incorporan restricciones de tipo de datos (entero, cadena, fecha), restricciones de rango (valores mínimos y máximos). La fecha se genera de manera automática cuando se realizan las transacciones y las claves primarias son auto incrementables. También se puede controlar el acceso a los datos para evitar modificaciones no autorizadas.

DDL

```
-- creación de la base de datos
CREATE DATABASE IF NOT EXISTS modulo3;

-- Selección la base de datos
USE modulo3;

-- Creación de la tabla usuarios
• CREATE TABLE IF NOT EXISTS Usuarios(
    user_id int auto_increment primary key,
    nombre varchar(80) NOT NULL,
    correo_electronico varchar(100) NOT NULL,
    contrasena varchar(100) NOT NULL,
    saldo decimal(10,2) NOT NULL);

-- Creación de la tabla moneda
• CREATE TABLE IF NOT EXISTS Monedas(
    currency_id int auto_increment primary key NOT NULL,
    currency_name varchar(50) NOT NULL,
    currenct_symbol varchar(5) NOT NULL);

-- Creación de la tabla transacciones
• CREATE TABLE IF NOT EXISTS Transacciones(
    transaccion_id int auto_increment primary key NOT NULL,
    sender_user_id int NOT NULL,
    receiver_user_id int NOT NULL,
    valor decimal(10,2) NOT NULL,
    transaction_date timestamp default current_timestamp NOT NULL,
    currency_id int NOT NULL,
    foreign key (sender_user_id) references Usuarios(user_id),
    foreign key (receiver_user_id) references Usuarios(user_id),
    foreign key (currency_id) references Monedas(currency_id));

/*
Sentencia para DDL modificar el nombre de la columna correo_electronico por email
*/
ALTER TABLE Usuarios
CHANGE COLUMN correo_electronico email varchar(100);
```

DML

```
-- Insertar usuarios, transacciones y monedas de muestra
INSERT INTO Usuarios (nombre, correo_electronico, contrasena, saldo)
VALUES
    ('Juan Perez Gonzalez', 'usuario@example.cl', sha2('123456',256), 500),
    ('María Peña Peña', 'usuario2@example.cl', sha2('1234567',256),800),
    ('Jose Pedro Morales', 'usuario3@example.cl', sha2('1234568',256),600);
SELECT * FROM Usuarios;

INSERT INTO Monedas (currency_name, currenct_symbol)
VALUE
    ('Dolar' , '$'),
    ('Euro' , '€' ),
    ('libra', '£');

INSERT INTO Transacciones(sender_user_id, receiver_user_id, valor, currency_id)
VALUES
    (1, 2 , 50, 1),
    (1, 3 , 100, 1),
    (2, 3, 200, 2),
    (3, 2, 150, 3);

/*
Consulta para obtener el nombre de la moneda elegida por un usuario específico
*/
SELECT currency_name FROM monedas WHERE currency_name = 'dolar';
```

```

/*
Consulta para obtener las transacciones REALIZADAS por un usuario específico
*/

SELECT
    emisor.nombre AS 'emisor',
    receptor.nombre AS 'receptor',
    transacciones.valor,
    transacciones.transaction_date As 'Fecha transacción'
FROM
    Transacciones
JOIN Usuarios AS emisor ON Transacciones.sender_user_id = emisor.user_id
JOIN Usuarios AS receptor ON Transacciones.receiver_user_id = receptor.user_id
WHERE
    Transacciones.sender_user_id = 1;

```

```

/*
Consulta para obtener todos los usuarios registrados
*/

SELECT user_id, nombre, correo_electronico, saldo FROM usuarios;

```

```

/*
Consulta para obtener todas las monedas registradas
*/

SELECT currency_id As 'id',
    currency_name As 'Nombre Moneda',
    currenct_symbol As 'Simbolo'
FROM monedas;

```

```

/*
Consulta para obtener todas las transacciones registradas
*/

Select transaction_id As 'id TRansacción',
    sender_user_id As 'Id Emisor',
    receiver_user_id As 'Id Receptor',
    valor,
    transaction_date As 'Fecha transacción'
FROM transacciones;

```

```

/*
Consulta para obtener todas las transacciones REALIZADAS por un usuario específico
*/
SELECT
    emisor.nombre AS 'emisor',
    receptor.nombre AS 'receptor',
    transacciones.valor,
    transacciones.transaction_date As 'Fecha transacción'
FROM
    Transacciones
JOIN Usuarios AS emisor ON Transacciones.sender_user_id = emisor.user_id
JOIN Usuarios AS receptor ON Transacciones.receiver_user_id = receptor.user_id
WHERE
    Transacciones.sender_user_id = 1;

```

```

/*
Sentencia DML para modificar el campo correo electrónico de un usuario específico
*/
UPDATE Usuarios
SET correo_electronico = 'nuevo_correo@example.com'
WHERE user_id = 1;

SELECT * FROM usuarios;

```

```

/*
Consulta para obtener todas las transacciones RECIBIDAS por un usuario específico
*/
SELECT
    receptor.nombre AS 'receptor',
    emisor.nombre AS 'emisor',
    transacciones.valor,
    transacciones.transaction_date As 'Fecha transacción'
FROM
    Transacciones
JOIN Usuarios AS receptor ON Transacciones.receiver_user_id = receptor.user_id
JOIN Usuarios AS emisor ON Transacciones.sender_user_id = emisor.user_id
WHERE
    Transacciones.receiver_user_id = 2;

```

/*
Sentencia para eliminar los datos de una transacción (eliminado de la fila completa)
*/

- **SELECT** * **FROM** Transacciones;
- **DELETE FROM** Transacciones
WHERE transaccion_id = 2;

/*
Generar consultas SQL para: obtener todos los usuarios
*/

- **SELECT** * **FROM** Usuarios;

DIAGRAMA EER

