Day - 9 Assignment (RDBMS and SQL) **Ankit Kumar Keshri**

Assignment 3: Design a database schema for a library system, including tables, fields, and constraints like NOT NULL, UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between tables.

```
CREATE TABLE Authors (
    author_id INT AUTO_INCREMENT PRIMARY KEY,
    author_name VARCHAR(100) NOT NULL
);

-- Table: Books
CREATE TABLE Books (
    book_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    isbn VARCHAR(13) UNIQUE,
    author_id INT,
    genre VARCHAR(100),
    publication_year INT,
    copies_available INT CHECK(copies_available >= 0),
    FOREIGN KEY (author_id) REFERENCES Authors(author_id)
);

-- Table: Members
CREATE TABLE Members (
    member_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    member_name VARCHAR(100) NOT NULL,
    email VARCHAR(255) UNIQUE,
    phone_number VARCHAR(20)
);

-- Table: Loans
CREATE TABLE Loans (
    loan_id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT,
    member_id INT,
    loan_date DATE NOT NULL,
    return_date DATE,
    FOREIGN KEY (book_id) REFERENCES Books(book_id),
    FOREIGN KEY (member_id) REFERENCES Members(member_id)
);

-- Table: Reservations
CREATE TABLE Reservations (
    reservation_id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT,
    member_id INT,
    reservation_date DATE NOT NULL,
    pickup_date DATE,
    status ENUM('Pending', 'Completed', 'Cancelled') DEFAULT 'Pending',
    FOREIGN KEY (book_id) REFERENCES Books(book_id),
    FOREIGN KEY (member_id) REFERENCES Members(member_id)
);
```

Assignment 4: Compose SQL statements to BEGIN a transaction, INSERT a new record into the 'orders' table, COMMIT the transaction, then UPDATE the 'products' table, and ROLLBACK the transaction.

```
-- BEGIN the transaction
BEGIN;

-- INSERT a new record into the 'orders' table
INSERT INTO orders (order_id, product_id, quantity, order_date)
VALUES (nextval('order_id_seq'), 'product_id_value', 'quantity_value', 'order_date_value');

-- COMMIT the transaction
COMMIT;

-- UPDATE the 'products' table
UPDATE products
SET column1 = 'value1', column2 = 'value2'
WHERE condition;

-- ROLLBACK the transaction
ROLLBACK;
```

Assignment 5: Begin a transaction, perform a series of INSERTs into 'orders', setting a SAVEPOINT after each, rollback to the second SAVEPOINT, and COMMIT the overall transaction.

```
-- BEGIN the transaction
BEGIN;

-- INSERT a new record into the 'orders' table and set a SAVEPOINT
SAVEPOINT savepoint1;
INSERT INTO orders (order_id, product_id, quantity, order_date)
VALUES (nextval('order_id_seq'), 'product_id_value1', 'quantity_value1',
'order_date_value1');

-- INSERT another record and set a SAVEPOINT
SAVEPOINT savepoint2;
INSERT INTO orders (order_id, product_id, quantity, order_date)
VALUES (nextval('order_id_seq'), 'product_id_value2', 'quantity_value2',
'order_date_value2');

-- Rollback to the second SAVEPOINT
ROLLBACK TO SAVEPOINT savepoint2;
```

```
-- COMMIT the overall transaction
COMMIT;
```