

Day-8 _Assignment:

Manjula Nannuri

Assignment 1: Analyse a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.

ER Diagram Analysis and Normalization

Business Scenario: Let's assume the scenario is about an e-commerce business.

Entities:

- **Customer:** CustomerID (PK), Name, Email, Phone
- **Product:** ProductID (PK), Name, Description, Price, Stock
- **Order:** OrderID (PK), OrderDate, CustomerID (FK)
- **OrderDetail:** OrderDetailID (PK), OrderID (FK), ProductID (FK), Quantity, UnitPrice
- **Category:** CategoryID (PK), CategoryName
- **ProductCategory:** ProductID (FK), CategoryID (FK)

Relationships:

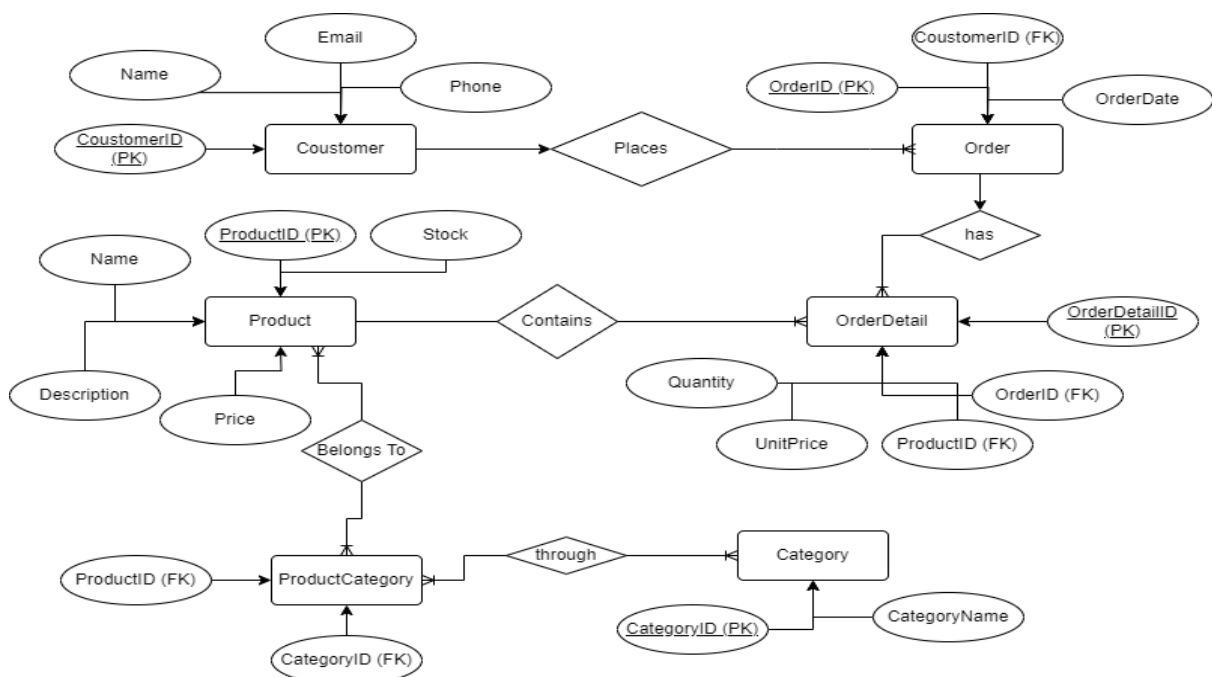
- Customer places Orders
- Order has OrderDetails
- OrderDetail contains Products
- Products belong to Categories through ProductCategory

Attributes and Cardinality:

- A customer can place many orders (1:N)
- An order can have many order details (1:N)
- An order detail refers to one product (N:1)
- A product can belong to many categories (M:N), managed through ProductCategory

Normalization:

- 1NF: Ensure atomicity of attributes.
- 2NF: Remove partial dependencies.
- 3NF: Remove transitive dependencies.



Assignment 2: 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.

Library System Database Schema

Tables and Fields:

- **Books:** BookID (PK), Title, Author, Publisher, YearPublished, ISBN (UNIQUE)
- **Members:** MemberID (PK), Name, Address, Phone, Email (UNIQUE)
- **Loans:** LoanID (PK), BookID (FK), MemberID (FK), LoanDate, ReturnDate
- **Authors:** AuthorID (PK), Name, Bio
- **BookAuthors:** BookID (FK), AuthorID (FK)

Constraints:

- **NOT NULL:** Ensure necessary fields are not left empty.
- **UNIQUE:** Ensure uniqueness where necessary.
- **CHECK:** Enforce rules (e.g., CHECK (YearPublished > 1900)).

Assignment 3: Explain the ACID properties of a transaction in your own words. Write SQL statements to simulate a transaction that includes locking and demonstrate different isolation levels to show concurrency control.

ACID Properties and Transaction Simulation

ACID Properties:

- **Atomicity:** Ensures that all operations in a transaction are completed; if not, the transaction is aborted.
- **Consistency:** Ensures the database remains in a consistent state before and after the transaction.
- **Isolation:** Ensures that transactions are executed independently.
- **Durability:** Ensures that once a transaction is committed, it remains in the system even in case of a failure.

```
mysql> -- Set Isolation Level to Serializable
mysql> SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;
Query OK, 0 rows affected (0.00 sec)

mysql> -- Start the first transaction
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> -- Perform operations
mysql> UPDATE Books SET Author = 'R Kipling' WHERE BookID = 3;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> -- Commit Transaction
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> -- Change isolation level
mysql> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> -- Demonstrate Concurrency Control
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> -- Perform a SELECT with a row lock
mysql> SELECT * FROM Books WHERE BookID = 1 FOR UPDATE;
+-----+-----+-----+-----+-----+-----+
| BookID | Title                | Author          | Publisher                | YearPubl |
ished | ISBN                | Genre          |                          |          |
+-----+-----+-----+-----+-----+-----+
|      1 | The Lord of the Rings | J.R.R. Tolkien | Houghton Mifflin Harcourt | 1954     |
  9780261102694 | Fantasy |                          |                          |          |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> -- Commit the second transaction
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

Assignment 4: Write SQL statements to CREATE a new database and tables that reflect the library schema you designed earlier. Use ALTER statements to modify the table structures and DROP statements to remove a redundant table.

SQL Statements for Database and Table Creation

-- Create Database

```
mysql> CREATE DATABASE LibraryDB;
Query OK, 1 row affected (0.01 sec)

mysql> USE LibraryDB;
Database changed
mysql>
```

--Create Table Books

```
MySQL 8.0 Command Line Client

mysql> CREATE TABLE Books ( BookID INT PRIMARY KEY AUTO_INCREMENT, Title VARCHAR(255) NOT NULL, Author VARCHAR(255) NOT NULL, Publisher VARCHAR(255), YearPublished YEAR, ISBN VARCHAR(13) UNIQUE );
Query OK, 0 rows affected (0.08 sec)

mysql> DESC Books;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| BookID | int | NO | PRI | NULL | auto_increment |
| Title | varchar(255) | NO | | NULL | |
| Author | varchar(255) | NO | | NULL | |
| Publisher | varchar(255) | YES | | NULL | |
| YearPublished | year | YES | | NULL | |
| ISBN | varchar(13) | YES | UNI | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)
```

--Create Table Members

```
MySQL 8.0 Command Line Client

mysql> CREATE TABLE Members ( MemberID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(255) NOT NULL, Address VARCHAR(255), Phone VARCHAR(20), Email VARCHAR(255) UNIQUE);
Query OK, 0 rows affected (0.06 sec)

mysql> DESC Members;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| MemberID | int | NO | PRI | NULL | auto_increment |
| Name | varchar(255) | NO | | NULL | |
| Address | varchar(255) | YES | | NULL | |
| Phone | varchar(20) | YES | | NULL | |
| Email | varchar(255) | YES | UNI | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

--Create Table Loans

```
MySQL 8.0 Command Line Client

mysql> CREATE TABLE Loans (LoanID INT PRIMARY KEY AUTO_INCREMENT, BookID INT NOT NULL, MemberID INT NOT NULL, LoanDate DATE NOT NULL, ReturnDate DATE, FOREIGN KEY (BookID) REFERENCES Books(BookID), FOREIGN KEY (MemberID) REFERENCES Members(MemberID));
Query OK, 0 rows affected (0.06 sec)

mysql> DESC Loans;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| LoanID | int | NO | PRI | NULL | auto_increment |
| BookID | int | NO | MUL | NULL | |
| MemberID | int | NO | MUL | NULL | |
| LoanDate | date | NO | | NULL | |
| ReturnDate | date | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

--Create Table Authors

```
MySQL 8.0 Command Line Client
mysql> CREATE TABLE Authors (AuthorID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(255) NOT NULL, Bio TEXT);
Query OK, 0 rows affected (0.03 sec)

mysql> DESC Authors;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| AuthorID | int | NO | PRI | NULL | auto_increment |
| Name | varchar(255) | NO | | NULL | |
| Bio | text | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

--Create Table BookAuthors

```
MySQL 8.0 Command Line Client
mysql> CREATE TABLE BookAuthors (BookID INT NOT NULL, AuthorID INT NOT NULL, FOREIGN KEY (BookID) REFERENCES Books(BookID), FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID), PRIMARY KEY (BookID, AuthorID));
Query OK, 0 rows affected (0.05 sec)

mysql> DESC BookAuthors;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| BookID | int | NO | PRI | NULL | |
| AuthorID | int | NO | PRI | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

-- Alter Table

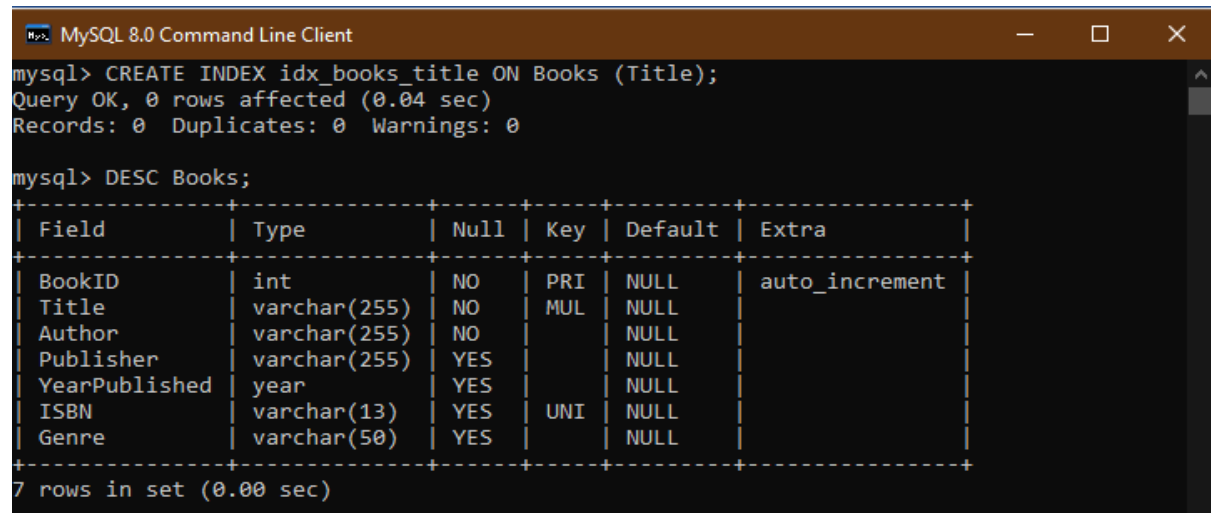
```
MySQL 8.0 Command Line Client
mysql> ALTER TABLE Books ADD COLUMN Genre VARCHAR(50);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESC BOOKS;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| BookID | int | NO | PRI | NULL | auto_increment |
| Title | varchar(255) | NO | | NULL | |
| Author | varchar(255) | NO | | NULL | |
| Publisher | varchar(255) | YES | | NULL | |
| YearPublished | year | YES | | NULL | |
| ISBN | varchar(13) | YES | UNI | NULL | |
| Genre | varchar(50) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Assignment 5: Demonstrate the creation of an index on a table and discuss how it improves query performance. Use a DROP INDEX statement to remove the index and analyse the impact on query execution.

Creating and Dropping Index

-- Create Index



```
mysql> CREATE INDEX idx_books_title ON Books (Title);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESC Books;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| BookID | int | NO | PRI | NULL | auto_increment |
| Title | varchar(255) | NO | MUL | NULL | |
| Author | varchar(255) | NO | | NULL | |
| Publisher | varchar(255) | YES | | NULL | |
| YearPublished | year | YES | | NULL | |
| ISBN | varchar(13) | YES | UNI | NULL | |
| Genre | varchar(50) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

-- Query to search for a book by title

```
SELECT * FROM Books WHERE Title = 'Sample Book';
```

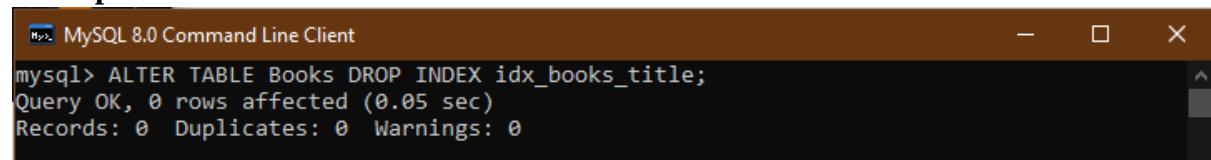
Query without Index:

Without an index, the database performs a full table scan to find rows where Title matches 'Sample Book'. This is inefficient, especially for large tables, because each row must be checked.

Query with Index:

With the index idx_books_title in place, the database uses the index to directly locate the rows where Title matches 'Sample Book'. This drastically reduces the number of rows that need to be examined and speeds up the query execution.

-- Drop Index



```
mysql> ALTER TABLE Books DROP INDEX idx_books_title;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Assignment 6: Create a new database user with specific privileges using the CREATE USER and GRANT commands. Then, write a script to REVOKE certain privileges and DROP the user.

Creating and Managing Database Users

```
MySQL 8.0 Command Line Client
mysql> -- Create User
mysql> CREATE USER 'lib_user'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.04 sec)

mysql> -- Grant Privileges
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON LibraryDB.* TO 'lib_user'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> -- Revoke Privileges
mysql> REVOKE DELETE ON LibraryDB.* FROM 'lib_user'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> -- Drop User
mysql> DROP USER 'lib_user'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Assignment 7: Prepare a series of SQL statements to INSERT new records into the library tables, UPDATE existing records with new information, and DELETE records based on specific criteria. Include BULK INSERT operations to load data from an external source.

SQL Statements for Data Manipulation

```
MySQL 8.0 Command Line Client
mysql> -- Inserting multiple records (one query per record)
mysql> INSERT INTO Books (Title, Author, Publisher, YearPublished, ISBN, Genre) VALUES
('The Jungle Book', 'Rudyard Kipling', 'Macmillan', 1984, '9780261102785', 'Childish'),
('Pride and Prejudice', 'Jane Austen', 'Penguin Classics', 1913, '9780140435225',
 'Romance');
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

-- Update Existing Records

```
MySQL 8.0 Command Line Client
mysql> -- Update based on ISBN
mysql> UPDATE Books SET Publisher = 'Houghton Mifflin Harcourt'
WHERE ISBN = '9780261102694';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

-- Delete Records

```
MySQL 8.0 Command Line Client
mysql> -- Delete based on Genre
mysql> DELETE FROM Books WHERE Genre = 'Romance';
Query OK, 1 row affected (0.01 sec)
```

-- Books.csv File

	A	B	C	D	E	F
1	Title	Author	Publisher	YearPublished	ISBN	Genre
2	The Hitchhiker's Guide to the Galaxy	Douglas Adams	Pan Books	1979	9.78E+12	Science Fiction
3	The Three-Body Problem	Cixin Liu	Tor Books	2006	9.78E+12	Science Fiction
4	One Hundred Years of Solitude	Gabriel Garcia Marquez	Penguin Random House	1967	9.78E+12	Magical Realism
5	Invisible Man	Ralph Ellison	Random House	1952	9.78E+12	Fiction
6	Beloved	Toni Morrison	Alfred A. Knopf	1987	9.78E+12	Historical Fiction
7	Pride and Prejudice	Jane Austen	Penguin Classics	1813	9.78E+12	Romance
8	The Lord of the Rings	J.R.R. Tolkien	Allen & Unwin	1954	9.78E+12	Fantasy
9	Dune	Frank Herbert	Chilton Books	1965	9.78E+12	Science Fiction
10	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1925	9.78E+12	Fiction
11	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	9.78E+12	Fiction
12						

-- Bulk Insert

```
mysql> LOAD DATA LOCAL INFILE 'C:\Users\nikhi\OneDrive\Desktop\Books.csv'
      INTO TABLE Books
      FIELDS TERMINATED BY ','
      LINES TERMINATED BY '\n'
      IGNORE 1 LINES;
```