ASSIGNMENT 1-7

Assignment 1: Analyze 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.

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.

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.

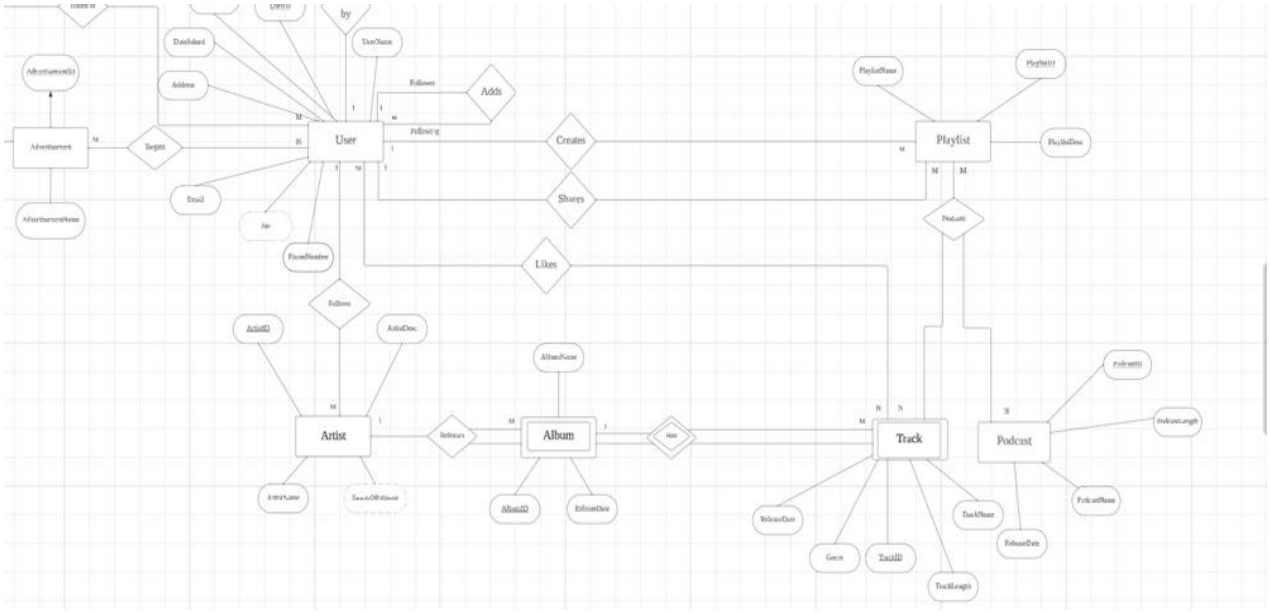
Assignment 4: Write SQL statements to CREATE a new database and tables that reflect the library schema you designed earlier. Use ALTER statements to modify the table structures and DROP statements to remove a redundant table.

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 analyze the impact on query execution.

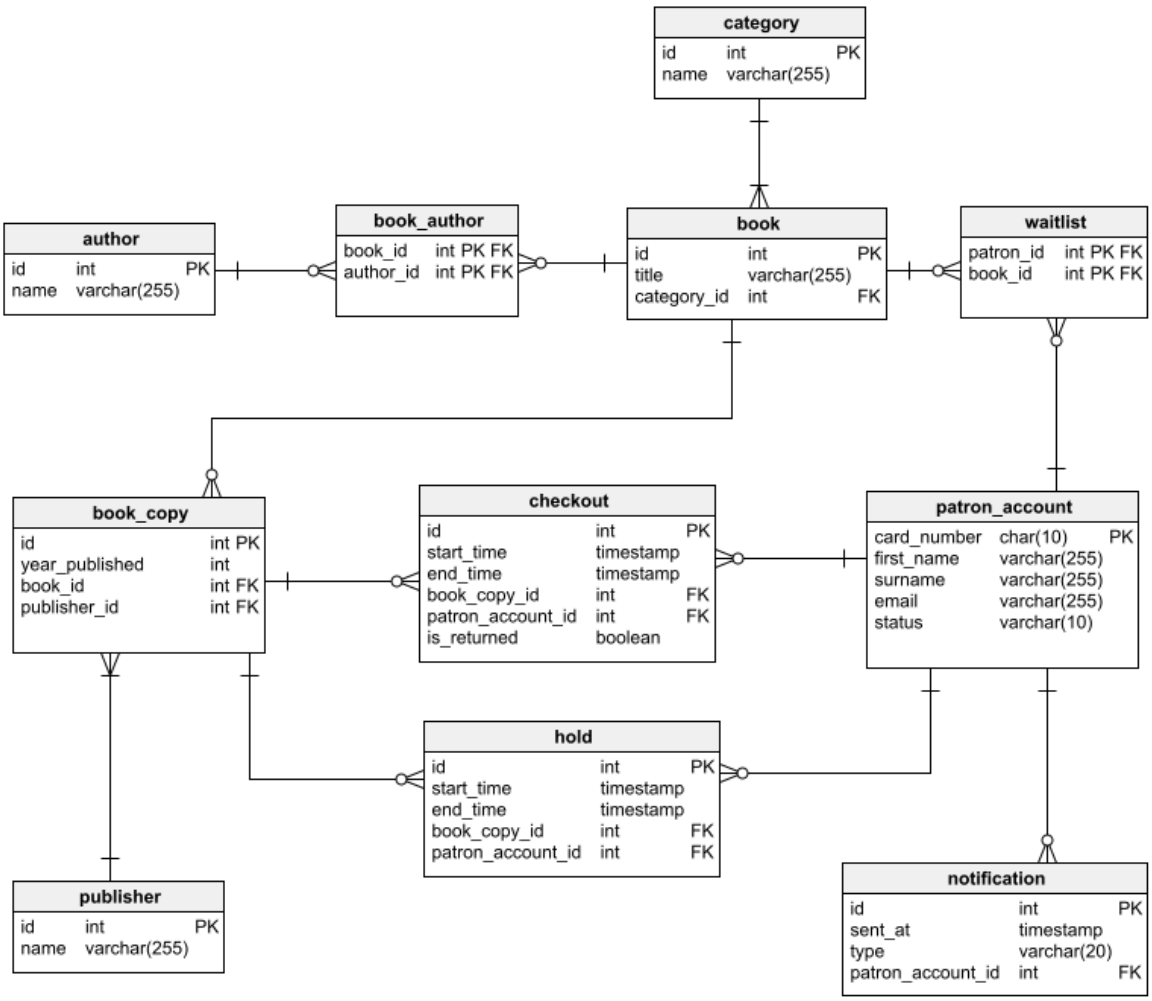
Assignment 6: Create a new database user with specific privileges using the CREATE USER and GRANT commands. Then, write a script to REVOKE certain privileges and DROP the user.

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

ASSIGNMENT-1



ASSIGNMENT-2



Assignment 3:

ACID Properties and SQL Transaction

ACID Properties:

1. **Atomicity**: Ensures that all operations within a transaction are completed successfully; if not, the transaction is aborted.
2. **Consistency**: Ensures that the database transitions from one valid state to another.
3. **Isolation**: Ensures that concurrent transactions do not interfere with each other.
4. **Durability**: Ensures that once a transaction is committed, it remains in the system even in the event of a failure.

SQL Transaction Example:

```
BEGIN TRANSACTION;

-- Locking the records involved

SELECT * FROM Accounts WHERE AccountID = 1 FOR UPDATE;

SELECT * FROM Accounts WHERE AccountID = 2 FOR UPDATE;

-- Debit from Account 1

UPDATE Accounts SET Balance = Balance - 100 WHERE AccountID = 1;

-- Credit to Account 2

UPDATE Accounts SET Balance = Balance + 100 WHERE AccountID = 2;

COMMIT;
```

ISOLATION Levels

-- Read Uncommitted

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

-- Read Committed

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

-- Repeatable Read

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

-- Serializable

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

ASSIGNMENT-4

```
-- Create tables
CREATE TABLE author (
    id INT PRIMARY KEY,
    name VARCHAR(255)
);

CREATE TABLE book (
    id INT PRIMARY KEY,
    title VARCHAR(255),
    category_id INT,
    year_published INT,
    book_copy_id INT,
    publisher_id INT,
    FOREIGN KEY (category_id) REFERENCES category(id),
    FOREIGN KEY (publisher_id) REFERENCES publisher(id)
);

CREATE TABLE category (
    id INT PRIMARY KEY,
    name VARCHAR(255)
);

CREATE TABLE book_author (
    book_id INT,
    author_id INT,
    PRIMARY KEY (book_id, author_id),
    FOREIGN KEY (book_id) REFERENCES book(id),
    FOREIGN KEY (author_id) REFERENCES author(id)
);
```

```
CREATE TABLE publisher (
    id INT PRIMARY KEY,
    name VARCHAR(255)
);

CREATE TABLE check_item (
    id INT PRIMARY KEY,
    start_time TIMESTAMP,
    end_time TIMESTAMP,
    book_copy_id INT,
    patron_account_id INT,
    FOREIGN KEY (book_copy_id) REFERENCES book_copy(id),
    FOREIGN KEY (patron_account_id) REFERENCES patron_account(id)
);

CREATE TABLE patron_account (
    id INT PRIMARY KEY,
    card_number VARCHAR(20),
    first_name VARCHAR(255),
    surname VARCHAR(255),
    email VARCHAR(100),
    status VARCHAR(20)
);

CREATE TABLE notification (
    id INT PRIMARY KEY,
    type VARCHAR(20),
    sent_at TIMESTAMP,
    account_id INT,
    FOREIGN KEY (account_id) REFERENCES patron_account(id) );

-- Alter statement
ALTER TABLE redundant_table_name ADD COLUMN new_column_name datatype;
-- Drop statement
DROP TABLE IF EXISTS redundant_table_name;
```

Assignment 5:

Creating and Dropping Indexes

Create Index:

CREATE INDEX idx_books_title ON Books (Title);

Impact of Index:

- Indexes improve query performance by allowing faster retrieval of records.
- Without the index, the database must perform a full table scan for queries.

Drop Index:

DROP INDEX idx_books_title ON Books;

RETRIEVALS WERE FASTER WHEN THERE WAS AN INDEX CREATED.

ASSIGNMENT-6

Create User and Grant Privileges:

CREATE USER 'library_user'@'localhost' IDENTIFIED BY 'password';

GRANT SELECT, INSERT, UPDATE, DELETE ON LibraryDB. TO 'library_user'@'localhost';*

Revoke Privileges and Drop User:

REVOKE UPDATE, DELETE ON LibraryDB. FROM 'library_user'@'localhost';*

DROP USER 'library_user'@'localhost';

ASSIGNMENT-7

-- Insert Records into Books table

INSERT INTO Books (Title, Author, ISBN, Publisher, YearPublished) VALUES

('Book1', 'Author1', '1234567890123', 'Publisher1', 2020),

('Book2', 'Author2', '1234567890124', 'Publisher2', 2021);

-- Insert Records into Members table

INSERT INTO Members (FirstName, LastName, Email, PhoneNumber) VALUES

('John', 'Doe', 'john.doe@example.com', '555-1234'),

('Jane', 'Smith', 'jane.smith@example.com', '555-5678');

-- Update Records in Books table

UPDATE Books SET Publisher = 'New Publisher' WHERE BookID = 1;

-- Delete Records from Members table

DELETE FROM Members WHERE MemberID = 2;

-- Bulk Insert operation for loading data from external source into Books table

LOAD DATA INFILE '/path/to/books.csv' INTO TABLE Books

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;