

PG-DAC AUGUST 25
Database Assignment2

Answers

Schema Design & Table Creation (DDL & DML Commands)

1. Define a schema for a Library Management System with the following entities: - Books - Authors - Members - Borrow_Records

Entities:

Books (book_id, title, author_id, published_year, available_copies, genre)

Authors (author_id, name, country)

Members (member_id, name, email, phone)

Borrow_Records (record_id, member_id, book_id, borrow_date, return_date)

2. Write the SQL command to create a table Authors with the following fields: - author_id (Primary Key, INT) - name (VARCHAR(100)) - country (VARCHAR(50))

```
CREATE TABLE Authors (  
    author_id INT PRIMARY KEY,  
    name VARCHAR(100),  
    country VARCHAR(50)  
);
```

3. Write the SQL command to create a table Books with the following fields: - book_id (Primary Key, INT) - title (VARCHAR(150)) - author_id (Foreign Key referencing Authors) - published_year (YEAR) - available_copies (INT)

```
CREATE TABLE Books (  
    book_id INT PRIMARY KEY,  
    title VARCHAR(150),  
    author_id INT,  
    published_year YEAR,  
    available_copies INT,  
    FOREIGN KEY (author_id) REFERENCES Authors(author_id)  
);
```

4. Write the SQL command to create a table Members with: - member_id (Primary Key, INT) - name (VARCHAR(100)) - email (VARCHAR(100), unique) - phone (VARCHAR(15))

```
CREATE TABLE Members (  
    member_id INT PRIMARY KEY,  
    name VARCHAR(100),  
    email VARCHAR(100) UNIQUE,  
    phone VARCHAR(15)  
);
```

5. Write the SQL command to create a table Borrow_Records with: - record_id (Primary Key, INT) - member_id (Foreign Key referencing Members) - book_id (Foreign Key referencing Books) - borrow_date (DATE) - return_date (DATE)

```
CREATE TABLE Borrow_Records (  
    record_id INT PRIMARY KEY,  
    member_id INT,  
    book_id INT,  
    borrow_date DATE,  
    return_date DATE,  
    FOREIGN KEY (member_id) REFERENCES Members(member_id),  
    FOREIGN KEY (book_id) REFERENCES Books(book_id)  
);
```

6. Modify the Books table to add a column genre of type VARCHAR(50).

```
ALTER TABLE Books  
ADD genre VARCHAR(50);
```

7. Write the SQL command to drop the Borrow_Records table.

```
DROP TABLE Borrow_Records;
```

8. Insert 3 records into the Authors table.

```
INSERT INTO Authors (author_id, name, country)  
VALUES  
(1, 'J.K. Rowling', 'United Kingdom'),  
(2, 'George R.R. Martin', 'United States'),  
(3, 'Yuval Noah Harari', 'Israel');
```

9. Insert 5 books into the Books table.

```
INSERT INTO Books (book_id, title, author_id, published_year, available_copies, genre)  
VALUES  
(1, 'Harry Potter and the Philosopher\'s Stone', 1, 1997, 5, 'Fantasy'),  
(2, 'Harry Potter and the Deathly Hallows', 1, 2007, 3, 'Fantasy'),  
(3, 'A Game of Thrones', 2, 1996, 4, 'Fantasy'),  
(4, 'A Clash of Kings', 2, 1998, 2, 'Fantasy'),  
(5, 'Sapiens: A Brief History of Humankind', 3, 2014, 6, 'Non-fiction');
```

10. Insert 3 members into the Members table.

```
INSERT INTO Members (member_id, name, email, phone)  
VALUES  
(1, 'Alice Johnson', 'alice@example.com', '1234567890'),  
(2, 'Bob Smith', 'bob@example.com', '9876543210'),  
(3, 'Charlie Davis', 'charlie@example.com', '5556667777');
```

11. Insert 4 borrow records into the Borrow_Records table.

```
INSERT INTO Borrow_Records (record_id, member_id, book_id, borrow_date, return_date)
VALUES
(1, 1, 1, '2025-09-01', '2025-09-10'),
(2, 2, 3, '2025-09-05', '2025-09-20'),
(3, 1, 5, '2025-09-15', NULL),
(4, 3, 2, '2025-09-18', NULL);
```

12. Write an SQL query to select all books where published_year is after 2015.

```
SELECT * FROM Books
WHERE published_year > 2015;
```

13. Write a SQL query to create a foreign key & primary key relationship between two tables.

```
ALTER TABLE Books
ADD CONSTRAINT fk_author
FOREIGN KEY (author_id) REFERENCES Authors(author_id);
```

14. Write an SQL query to find all members who have borrowed the book with title 'Database Systems'.

```
SELECT m.name, m.email
FROM Members m
JOIN Borrow_Records br ON m.member_id = br.member_id
JOIN Books b ON br.book_id = b.book_id
WHERE b.title = 'Database Systems';
```

15. Update the available_copies column of a specific book (choose any book) by reducing it by 1 after it is borrowed.

```
UPDATE Books
SET available_copies = available_copies - 1
WHERE book_id = 1;
```

16. Delete a record from Members where member_id = 3.

```
DELETE FROM Members
WHERE member_id = 3;
```

17. Update a Book name record from Book table with id = 1.

```
UPDATE Books
SET title = 'Harry Potter and the Sorcerer\'s Stone'
WHERE book_id = 1;
```

18. Write an SQL query to list all books along with their authors' names.

```
SELECT b.title, a.name AS author_name  
FROM Books b  
JOIN Authors a ON b.author_id = a.author_id;
```

19. Write an SQL query to delete all books from the Books table where the published_year is before 2000.

```
DELETE FROM Books  
WHERE published_year < 2000;
```

20. Write an SQL query to find all books that are never borrowed (i.e., no records in Borrow_Records).

```
SELECT b.title  
FROM Books b  
LEFT JOIN Borrow_Records br ON b.book_id = br.book_id  
WHERE br.book_id IS NULL;
```