

Name: Bhavik Pahuja

UID: 24BAI70522

Course: BE-CSE (AI&ML)

Subject: Database Management Systems

Experiment: Library Management System Implementation

1. Aim of the Session

The aim of this practical is to design and implement a relational database schema for a Library Management System. This involves defining tables with specific constraints, establishing relationships between entities, and managing database security through role-based access control.

2. Objective of the Session

Upon completing this session, the following objectives were achieved:

- Developed table structures using Primary Keys, Foreign Keys, and Check Constraints for data validation.
- Gained proficiency in DML (Data Manipulation Language) operations, specifically INSERT, SELECT, UPDATE, and DELETE.
- Implemented DCL (Data Control Language) to manage user roles and granular permissions.
- Maintained referential integrity across multiple related tables (books, library_visitors, and book_issue).

3. Practical / Experiment Steps

The implementation was carried out through the following tasks:

1. Schema Definition: Created the base tables for books and library_visitors with specific constraints such as NOT NULL, UNIQUE, and CHECK (e.g., ensuring visitor age is 18+).
2. Relational Setup: Created the book_issue table to act as a transaction bridge, linking books and visitors via Foreign Keys.
3. Data Population: Populated the tables with initial records to test the schema's validity.

4. Operational Testing: Performed updates on user information and attempted deletion of records to observe constraint behavior.
5. Security Administration: Created a librarian role with login credentials and configured its access levels using GRANT and REVOKE commands.

4. Procedure of the Practical

The following steps were followed during the execution:

1. System Initialization: Logged into the database environment and established a connection to the server.
2. Database Creation: Initialized a new database to house the library management system.
3. Executing Table Scripts: Ran the CREATE TABLE commands in a specific sequence (creating parent tables before dependent transaction tables).
4. Data Entry: Executed INSERT statements to add sample books and visitor profiles.
5. Query Verification: Used SELECT queries to verify that the data was correctly stored and consistent across tables.
6. Data Modification: Tested the UPDATE and DELETE commands to ensure the system handles changes as intended.
7. Role Configuration: Defined the librarian role and assigned specific table privileges.
8. Security Verification: Tested and then revoked permissions to confirm the effectiveness of the security policy.
9. Record Maintenance: Saved the SQL script and took screenshots of the execution results.

5. I/O Analysis (Input / Output Analysis)

Input Queries

SQL

-- Table Creation

CREATE TABLE books(

id INT PRIMARY KEY,

```
    name VARCHAR(50) NOT NULL,  
    author_name VARCHAR(50) NOT NULL,  
    count INT CHECK(count>0)  
);
```

```
CREATE TABLE library_visitors(  
    user_id INT PRIMARY KEY,  
    user_name VARCHAR(20) NOT NULL,  
    age INT CHECK(age>=18) NOT NULL,  
    email VARCHAR(40) UNIQUE NOT NULL  
);
```

```
CREATE TABLE book_issue(  
    book_issue_id INT PRIMARY KEY,  
    book_id INT NOT NULL,  
    user_id INT NOT NULL,  
    FOREIGN KEY (book_id) REFERENCES books(id),  
    FOREIGN KEY (user_id) REFERENCES library_visitors(user_id),  
    book_issue_date DATE NOT NULL  
);
```

```
-- Data Manipulation  
INSERT INTO books VALUES(1, 'Hairy Popter', 'R. Snap', 1);  
INSERT INTO library_visitors VALUES(101, 'Robert', 20, 'abc@il.com');  
UPDATE library_visitors SET email='Robel.com' WHERE user_id = 101;
```

-- Role Management

```
CREATE ROLE librarian WITH LOGIN PASSWORD 'WHIPWHIP';
```

```
GRANT SELECT, INSERT, DELETE, UPDATE ON books TO librarian;
```

Output Details

- Schema Success: All tables were created successfully. The system correctly enforced the CHECK(age>=18) constraint, rejecting invalid entries.
- DML Results: The UPDATE query correctly modified the email field for user 101, and SELECT queries displayed the current state of all tables accurately.
- DCL Verification: The librarian role was successfully created and assigned the necessary privileges for library management tasks.
- Validation: Testing confirmed that after the REVOKE command, the librarian could no longer perform operations on the books table, ensuring the security policy is functional.
- We also confirmed the permissions of the role "librarian" by checking the table privileges.

```

    ▶ Run | +Tab | JSON | ⚡Ask AI
4   SELECT table_name, privilege_type
5   FROM information_schema.table_privileges
6   WHERE grantee = 'librarian'; 30ms

```

	table_name	privilege_type
1	books	INSERT
2	books	SELECT
3	books	UPDATE
4	books	DELETE
5	library_visitors	INSERT
6	library_visitors	SELECT
7	library_visitors	UPDATE
8	library_visitors	DELETE
9	book_issue	INSERT
10	book_issue	SELECT
11	book_issue	UPDATE

▶ Run

```

    ✓ 1   INSERT INTO books (id, name, author_name, count)
    2     VALUES (3, 'The Great Gatsby', 'F. Scott Fitzgerald', 5); 30ms  AffectedRows: 1
    3

```

▶ Run | ⚡Select

```

    ✓ 68  GRANT SELECT, INSERT, DELETE, UPDATE
    69  ON book_issue
    70  TO librarian; 29ms
    71

```

▶ Run | ⚡Select

ON	X
GRANT SELECT, INSERT, DELETE, UPDATE	ON book_issue
	TO librarian

▶ Run | ⚡Select

```
64    ▶ Run | ▶ Select
65    GRANT SELECT, INSERT, DELETE, UPDATE
66    ON library_visitors
67    TO librarian; 13ms
68    ▶ Run | ▶ Select
69    GRANT SELECT, INSERT, DELETE, UPDATE
70    ON book_issue
71    TO librarian;
72    ▶ Run | ▶ Select
73    REVOKE SELECT, INSERT, DELETE, UPDATE
74    ON books
75    TO librarian;
76    ▶ Run | ▶ Select
77    GRANT SELECT, INSERT, DELETE, UPDATE
78    ON books
79    TO librarian; 36ms
80    ▶ Run | ▶ Select
81    GRANT SELECT, INSERT, DELETE, UPDATE
82    ON library_visitors
83    TO librarian;
84    ▶ Run | ▶ Select
85    GRANT SELECT, INSERT, DELETE, UPDATE
86    ON book_issue
87    TO librarian;
88    ▶ Run | ▶ Select
89    REVOKE SELECT, INSERT, DELETE, UPDATE
90    ON books
91    TO librarian;
```

GRANT SELECT, INSERT, DELETE, UPDATE ON library_visitors TO librarian


```
60    ▶ Run | ▶ Select
61    GRANT SELECT, INSERT, DELETE, UPDATE
62    ON books
63    TO librarian; 36ms
64    ▶ Run | ▶ Select
65    GRANT SELECT, INSERT, DELETE, UPDATE
66    ON library_visitors
67    TO librarian;
68    ▶ Run | ▶ Select
69    GRANT SELECT, INSERT, DELETE, UPDATE
70    ON book_issue
71    TO librarian;
72    ▶ Run | ▶ Select
73    REVOKE SELECT, INSERT, DELETE, UPDATE
74    ON books
75    TO librarian;
```

GRANT SELECT, INSERT, DELETE, UPDATE ON books TO librarian

```

    ▶ Run
57 CREATE ROLE librarian
58 WITH LOGIN PASSWORD 'NaNaGandiBaat'; 41ms
59
60 ▶ Run | ▶ Select
61 GRANT SELECT, INSERT, DELETE, UPDATE
62 ON books
63 TO librarian;
64 ▶ Run | ▶ Select
65 GRANT SELECT, INSERT, DELETE, UPDATE
66 ON library_visitors
67 TO librarian;
68 ▶ Run | ▶ Select
69 GRANT SELECT, INSERT, DELETE, UPDATE
70 ON book_issue
71 TO librarian;
72
73 ▶ Run | ▶ Select
74 REVOKE SELECT, INSERT, DELETE, UPDATE
75 ON books;

```

Result ×

Search Results Export Cost: 41ms

CREATE ROLE librarian ↵WITH LOGIN PASSWORD 'NaNaGandiBaat'

```

C: > Users > bhavi > .dbclient > query > 1761209482832@127.0.0.1@5432@dbms_class@public > New_Query_17694179
39
40 ▶ Run | +Tab | JSON | Ask AI
41 SELECT *
42 FROM library_visitors;
43
44 ▶ Run | ▶ Select
45 INSERT INTO book_issue
46 VALUES
47 (1234,1,101,'2026-01-07');
48
49 ▶ Run | +Tab | JSON | Ask AI
50 SELECT *
51 FROM book_issue;
52
53 ▶ Run | ▶ Select
54 DELETE
55 FROM books
56 WHERE id = 3;
57
58 ▶ Run | +Tab | JSON | Ask AI
59 SELECT *
60 FROM books; 3ms
61
62

```

books ×

Search Results Export Cost: 7ms Total 2

	id	*name	*author_name	count
1	Hairy Popter	R. Snap	1	
2	Revengers	Stan Man	3	

```

C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
39
40     ▷ Run | +Tab | JSON | Ask AI
41     SELECT *
42     FROM library_visitors;
43
44     ▷ Run | Select
45     INSERT INTO book_issue
46     VALUES
47     (1234,1,101,'2026-01-07');
48
49     ▷ Run | Select
50     DELETE
51     FROM books
52     WHERE id = 3;  3ms
53
54     ▷ Run | Select
55     SELECT *
56     FROM books;
56

```

book_issue					
	book_issue_id	book	user	book_issue_date	
	1234	1	101	2026-01-07	

```

C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
39
40     ▷ Run | +Tab | JSON | Ask AI
41     SELECT *
42     FROM library_visitors;
43
44     ▷ Run | Select
45     INSERT INTO book_issue
46     VALUES
47     (1234,1,101,'2026-01-07');
48
49     ▷ Run | Select
50     DELETE
51     FROM books
52     WHERE id = 3;
53
54     ▷ Run | Select
55     SELECT *
56     FROM books;
56

```

book_issue					
	book_issue_id	book	user	book_issue_date	
	1234	1	101	2026-01-07	

```

C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
39
40     ▷ Run | +Tab | JSON | Ask AI
41     SELECT *
42     FROM library_visitors;
43
44     ▷ Run | Select
45     ✓ INSERT INTO book_issue
46     VALUES
47     (1234,1,101,'2026-01-07'); 24ms AffectedRows: 1
48
49     ▷ Run | Select
50     SELECT *
51     FROM book_issue;
52
53     ▷ Run | Select
54     DELETE
55     FROM books
56     WHERE id = 3;
57
58     ▷ Run | Select
59     SELECT *
60     FROM books;
61

```

library_visitors			
<input type="button" value="Search Results"/> <input type="button" value="Export"/> Cost: 31ms < 1 > Total 1			
	user_id	user_name	age
	Q 101	Robert	20
			Robel.com

```

C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
39
40     ▷ Run | +Tab | JSON | Ask AI
41     ✓ SELECT *
42     FROM library_visitors; 3ms
43
44     ▷ Run | Select
45     INSERT INTO book_issue
46     VALUES
47     (1234,1,101,'2026-01-07');
48
49     ▷ Run | Select
50     SELECT *
51     FROM book_issue;
52
53     ▷ Run | Select
54     DELETE
55     FROM books
56     WHERE id = 3;
57
58     ▷ Run | Select
59     SELECT *
60     FROM books;
61

```

library_visitors			
<input type="button" value="Search Results"/> <input type="button" value="Export"/> Cost: 31ms < 1 > Total 1			
	user_id	user_name	age
	Q 101	Robert	20
			Robel.com


```

C: > Users > bhavi > .dbclient > query > 1761209482832@127.0.0.1@5432@dbms_class@public > New_Query_176941795
21 |      BOOK_ISSUE_DATE DATE NOT NULL
22 );
23
24     ▶ Run | Select | Ask AI
25 INSERT INTO books
26 VALUES
27 (1, 'Hairy Popter', 'R. Snap', 1),
28 (2, 'Revengers', 'Stan Man', 3);
29
30     ▶ Run | +Tab | JSON | Ask AI
31 SELECT *
32 FROM books;
33
34     ▶ Run | Select
35 INSERT INTO library_visitors
36 VALUES
37 (101, 'Robert', 20, 'abc@il.com'); 29ms AffectedRows: 1
38
39     ▶ Run | Select
40 UPDATE library_visitors
41 SET email='Robel.com'
42 WHERE user_id = 101;
43

```

books				
	id	* name	* author_name	count
	integer	varchar(50)	varchar(50)	integer
	> 1	Hairy Popter	R. Snap	1
	> 2	Revengers	Stan Man	3

```
C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
21      |          BOOK_ISSUE_DATE DATE NOT NULL
22  );
23
24  INSERT INTO books
25  VALUES
26  (1, 'Hairy Popter', 'R. Snap', 1),
27  (2, 'Revengers', 'Stan Man', 3);
28
29  SELECT *
30  FROM books;  2ms
31
32  INSERT INTO library_visitors
33  VALUES
34  (101, 'Robert', 20, 'abc@il.com');
35
36  UPDATE library_visitors
37  SET email='Robel.com'
38  WHERE user_id = 101;
39
```

books					
	Q	id	* name	* author_name	count
	Q	integer	varchar(50)	varchar(50)	integer
	>	1	Hairy Popter	R. Snap	1
	>	2	Revengers	Stan Man	3

```
23
24 ✓ 24 INSERT INTO books
25 VALUES
26 (1, 'Hairy Popter', 'R. Snap', 1),
27 (2, 'Revengers', 'Stan Man', 3); AffectedRows: 2 7ms
28
29 ▶ Run | +Tab | JSON | Ask AI
30 SELECT *
31 FROM books;
32
33 ▶ Run | Select
34 INSERT INTO library_visitors
35 VALUES
36 (101, 'Robert', 20, 'abc@il.com');
37
38 ▶ Run | Select
39 UPDATE library_visitors
40 SET email='Robel.com'
41 WHERE user_id = 101;
42
43
44
```

Result

Execution completed in 82ms

```
C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
13 );
14
15 ✓ CREATE TABLE book_issue(
16     book_issue_id INT PRIMARY KEY,
17     book_id INT NOT NULL,
18     user_id INT NOT NULL,
19     FOREIGN KEY (book_id) REFERENCES books(id),
20     FOREIGN KEY (user_id) REFERENCES library_visitors(user_id),
21     book_issue_date DATE NOT NULL
22 ); 82ms
23
24 ▶ Run | Select | Ask AI
25 INSERT INTO books
26 VALUES
27 (1, 'Hairy Popter', 'R. Snap', 1),
28 (2, 'Revengers', 'Stan Man', 3);
29
30 ▶ Run | +Tab | JSON | Ask AI
31 SELECT *
32 FROM books;
33
34 ▶ Run | Select
```

Result

Execution completed in 82ms

```

C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
    ▶ Run | ▶ Select | ▶ Ask AI
1 CREATE TABLE books(
2     id INT PRIMARY KEY,
3     name VARCHAR(50) NOT NULL,
4     author_name VARCHAR(50) NOT NULL,
5     count INT CHECK(count>0)
6 );
7
8 CREATE TABLE library_visitors(
9     user_id INT PRIMARY KEY,
10    user_name VARCHAR(20) NOT NULL,
11    age INT CHECK(age>=18) NOT NULL,
12    email VARCHAR(40) UNIQUE NOT NULL
13 );
14 35ms
15
16 CREATE TABLE book_issue(
17     book_issue_id INT PRIMARY KEY,
18     book_id INT NOT NULL,
19     user_id INT NOT NULL,
20     FOREIGN KEY (book_id) REFERENCES books(id),
21     FOREIGN KEY (user_id) REFERENCES library_visitors(user_id),
22
Result ×
⊕ ⌛ Q Search Results 🔍 + + 🗑️ ⏪ ⏩ ⏵ Export ⏴ ⏵ Cost: 35ms < >
Execution completed in 35ms

```



```

C: > Users > bhavi > .dbclient > query > 1761209482832@@127.0.0.1@5432@dbms_class@public > New_Query_176941795
    ▶ Run | ▶ Select | ▶ Ask AI
1 CREATE TABLE books(
2     id INT PRIMARY KEY,
3     name VARCHAR(50) NOT NULL,
4     author_name VARCHAR(50) NOT NULL,
5     count INT CHECK(count>0)
6 );
7 55ms
8
9 CREATE TABLE library_visitors(
10    user_id INT PRIMARY KEY,
11    user_name VARCHAR(20) NOT NULL,
12    age INT CHECK(age>=18) NOT NULL,
13    email VARCHAR(40) UNIQUE NOT NULL
14 );
15
16 CREATE TABLE book_issue(
17     book_issue_id INT PRIMARY KEY,
18     book_id INT NOT NULL,
19     user_id INT NOT NULL,
20     FOREIGN KEY (book_id) REFERENCES books(id),
21     FOREIGN KEY (user_id) REFERENCES library_visitors(user_id),
22
Result ×
⊕ ⌛ Q Search Results 🔍 + + 🗑️ ⏪ ⏩ ⏵ Export ⏴ ⏵ Cost: 55ms < >
Execution completed in 55ms

```

The screenshot shows a SQL query editor interface. At the top, there are buttons for 'Run' and 'Select'. Below that, the code is displayed:

```
72 REVOKE SELECT, INSERT, DELETE, UPDATE  
73 ON books  
74 FROM librarian; 29ms
```

Below the code, there is a toolbar with various icons: a plus sign, a minus sign, a search bar labeled 'Search Results', a gear icon, an envelope icon with a '1' notification, a plus sign, a minus sign, a refresh icon, a double arrow icon, an export icon, and a cost indicator 'Cost: 29ms'. The status bar at the bottom shows the same SQL command and its execution time.

6. Learning Outcome

This practical session provided significant insights into:

- Structural Logic: Understanding how Foreign Keys and Check Constraints maintain high data quality and prevent logical errors.
- Security Implementation: Learning to manage database security through roles rather than individual user permissions.
- Practical Application: Applying SQL fundamentals to a real-world scenario (Library Management), demonstrating how relational databases handle complex interactions between entities.