Semester: 4 **Year: 2022-23**

Department: *Artificial Intelligence & Data Science*	***Course Type: Core***
Course Title: *Database Management Systems Lab*	***Course Code:*** *21ADL48*
L-T-P: 0*-0-2*	***Credits:****03*
Total Contact Hours: 30 *hours*	***Duration of SEE:****03 hours*
SEE Marks*: 50*	***CIE Marks:*** *50*

At the end of the course, a student shall be able to:

Cos	**Course Outcome Description**	**Blooms Level**
1	Describe the components of database systems.	L1
3	Explain the principles of database design and implementation.	L2
4	Design a database system for a given application	L3
5	Implement a database using DBMS and interface it with an application	L3
6	Test and validate the developed database system	L3

Course Content of Database Management Systems Lab course:

Exercises in the specified problems shown below will be performed by students.
- Analyze and perform data modeling for a given application
- Convert the data model into a relational model
- Use DDL and DML commands in SQL queries
- Develop SQL commands to create and maintain database structure
- Interface application to the developed database system
- Multidimensional data modelling.

Exercise 1: Consider the following schema of a library database.

Book (Book_id, Title, Pub_name, Pub_year)

Book_Author (Book_id, Author_Name)

a) Retrieve book names published in 2017

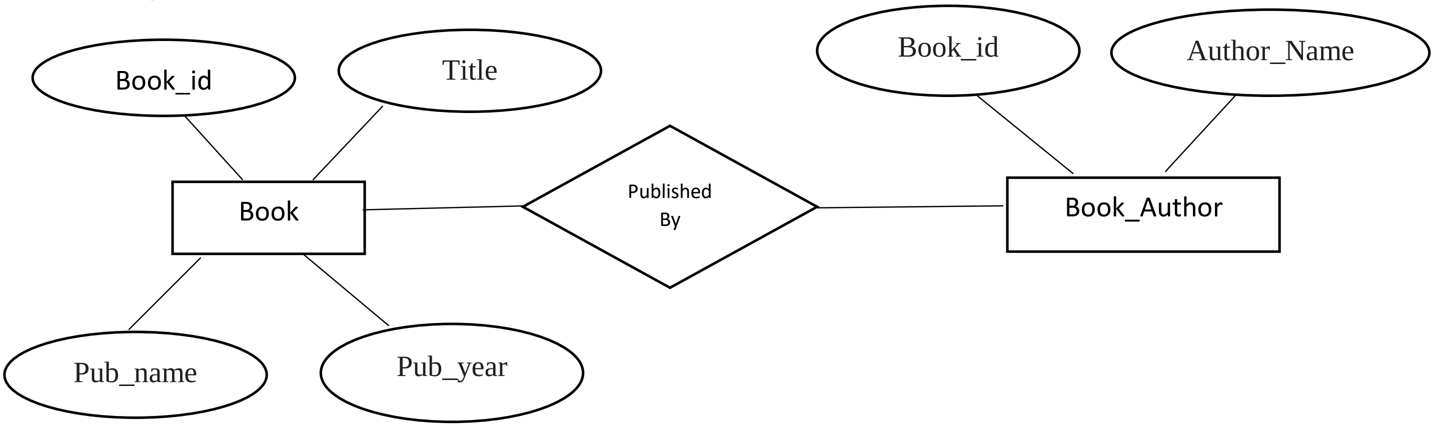
b) Delete data having book name “Harry porter”

c) Insert a row into “Book” schema

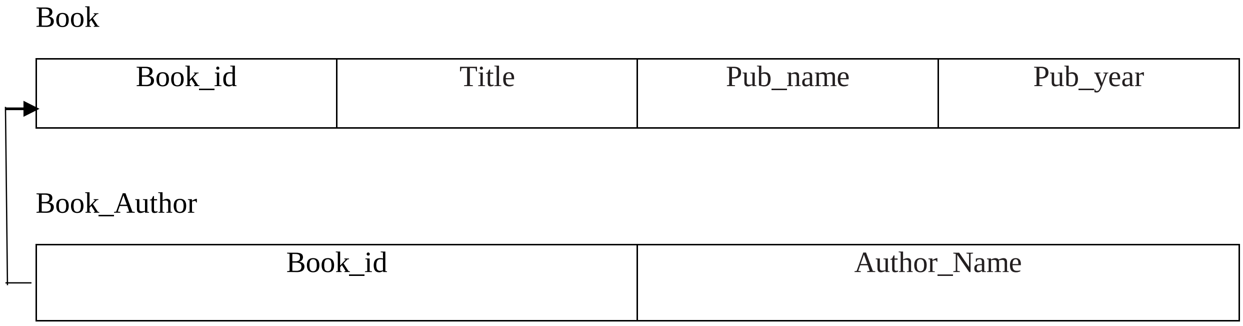
d) Check whether insertion possible in table “Book_Author”

Solution:

ER Digram:



Schema Diagram:



Book

Book_id	Title	Pub_name	Pub_year

Book_Author

Book_id	Author_Name

Creation of Database:

create database library;

Table creation:

Create table Book (Book_id int primary key, title varchar(50), pub_name varchar(50), Pub_year year);

Create table Book_Authors (Book_id int, Author_Name varchar(50), foreign key (Book_id) references Book(Book_id) on delete cascade);

Insertion into table:

Book:

```
Insert into Book ( Book_id, title, pub_name, Pub_year)
Values( 1000, ‘DBMS’, ‘Pearson’, 2017),
(1001, ‘Fundamentals of DB’, ‘Penguin’, 1997),
(1002, ‘Harry Potter’, ‘Book Hound’, 2017);
```

Book_Authors:

```
Insert into Book_Authors(Book_id, Author_Name)
Values(1000, ‘B Navathe’),
(1001, ‘B Navathe’),
(1002, ‘JK Rowling’);
```

a) Retrieve book names published in 2017

```
select title
from Book
where Pub_year = 2017;
```

b) Delete data having book name “Harry potter”

```
delete
from book
where title = ‘Harry Potter’;
```

c) Insert a row into “Book” schema

```
Insert into Book ( Book_id, title, pub_name, Pub_year)
values(1003, ‘Operating System’, ‘Wiley’, 1960);
```

d) Check whether insertion possible in table "Book_Author"

i. Insertion in child table (Book_author) is possible if primary key of parent table (Book) is same with foreign key of child table (Book_author). Following query will insert values into Book_Author:

```
Insert into Book_Authors(Book_id, Author_Name),
values(1003, 'Galvin');
```

ii. Insertion in child table (Book_author) is not possible if primary key of parent table (Book) is not same with foreign key of child table (Book_author).

```
Insert into Book_Authors(Book_id, Author_Name),
values(1004, 'Galvin');
```

Exercise 2: Consider the following schema for a Library Database:
BOOK (Book_ id, Title, Publisher_ Name, Pub_ Year)
BOOK_AUTHORS (Book_ id, Author_ Name)
PUBLISHER (Name, Address, Phone)
BOOK_COPIES (Book_ id, Branch_ id, No_ of_ Copies)
BOOK_LENDING (Book_ id, Branch_ id ,Card_ No, Date_ Out, Due_ Date)
LIBRARY_BRANCH (Branch_ id, Branch_ Name, Address)

a) Retrieve details of all books: title, name of publisher, authors, number of copies in each branch.
b) Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
c) Delete a book from Book table, update the contents of other tables to reflect this data manipulation operation.
d) Create a view of all books and its number of copies that are currently available in the library.

Solution:

ER Digram:

Schema Diagram:

Table Creation:

```
CREATE TABLE PUBLISHER(NAME VARCHAR(20) PRIMARY KEY,ADDRESS
VARCHAR(20),PHONE REAL);

CREATE TABLE BOOK(BOOK_ID INT PRIMARY KEY,TITLE
CHAR(10),PUBLISHER_NAME CHAR(15),PUBLISHER_YEAR INT,FOREIGN
KEY(PUBLISHER_NAME) REFERENCES PUBLISHER(NAME) ON DELETE CASCADE);

CREATE TABLE LIBRARY_BRANCH(BRANCH_ID INT PRIMARY KEY,BRANCH_NAME
CHAR(10),ADDRESS VARCHAR(15));
```

```
CREATE TABLE BOOK_AUTHORS(BOOK_ID INT,AUTHOR_NAME CHAR(10),FOREIGN
KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE);

CREATE TABLE BOOK_LENDING(BOOK_ID INT,BRANCH_ID INT,CARD_NO
INT,DATE_OUT DATE,DUE_DATE DATE,PRIMARY
KEY(BOOK_ID,BRANCH_ID,CARD_NO),FOREIGN KEY(BOOK_ID) REFERENCES
BOOK(BOOK_ID) ON DELETE CASCADE,FOREIGN KEY(BRANCH_ID) REFERENCES
LIBRARY_BRANCH(BRANCH_ID) ON DELETE CASCADE);

CREATE TABLE BOOK_COPIES(BOOK_ID INT,BRANCH_ID INT,NO_OF_COPIES
INT,FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE
CASCADE,FOREIGN KEY(BRANCH_ID) REFERENCES LIBRARY_BRANCH(BRANCH_ID)
ON DELETE CASCADE,PRIMARY KEY(BOOK_ID,BRANCH_ID));
```

Insertion of values to tables:

```
INSERTINTO PUBLISHER
VALUES(‘ANUAMMUBOOKS’,’SRIRAMRKNAGAR’,8797546021),
(‘BHAVANISTORE’,’KARNATAKABAR’,9900778821),
(‘RAJBOOKSTALL’,’SATALLITE@12’,9252459821),
(‘SNEHABANDHU’,’MANDYA123’,9380577730),
(‘SPOO@SHBOOKS’,’NEARBEMLARCH’,8969361282),
(‘SRI SAI’,’DEVEGOWDANAHUNDI’,’7090454640’);

INSERT INTO BOOK VALUES(100,’OOMD’,’SRI SAI’,1996),
VALUES(101,’CMPTRNTWK’,’SNEHABANDHU’,1992),
(102,’AUTOMATA’,’RAJBOOKSTALL’,2000),
(103,’CSHARP’,’SPOO@ASHBOOKS’,2012),
(104,’MANAGEMENT’,’SRI SAI’,1996),
(105,’DATABASEMS’,’BHAVANISTORE’,1899);

INSERT INTO LIBRARY_BRANCH
VALUES(1000,’BRANCH1’,’GROUNDFLOAR’),
(1001,’BRANCH2’,’GFROWF’),
(1002,’BRANCH1’,’LFTSIDEROW7’),
(1003’,’BRANCH3’,’RGHTSIDEROW7’),
(‘1004’,’BRANCH4’,’OPPOSITEFLOAR’),
(‘1005’,’BRANCH5’,’ECEFLOAR’);

INSERT INTO BOOK_AUTHORS
VALUES(100,’NAWATHI’),
(100,’NAWATHI’),
(‘101’,’THRIPATI’),
(‘102’,’PADMAREDDY’),
(103,”KRISHNA”);

INSERT INTO BOOK_LENDING
VALUES(“100”,”1000”,”50”,”2017/01/03”,”2017/06/26”),
(“101”,”1001”,”50”,”2017/01/26”,”2017/06/26”),
(“102”,”1002”,”50”,”2017/01/15”,”2017/06/26”),
(“103”,”1003”,”50”,”2018/01/26”,”2018/06/03”),
(“104”,”1004”,”54”,”2017/01/03”,”2017/06/03”);

INSERT INTO BOOK_COPIES
VALUES(“101”,”1000”,”3”),
(“100”,”1004”,”4”),
(“101”,”1006”,”2”),
(“103”,”1003”,”5”),
(“104”,”1002”,”5”);
```

a) Retrieve details of all books: title, name of publisher, authors, number of copies in each branch.

```
Select title, Pub_name, Author_name, No_of_copis
From Book, Book_copies, Book_Author
Where Book.Book_id = Book_copies.Book_id and Book.Book_ID = Book_Author.Book_id;
```

b) Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.

```
SELECT CARD_NO
FROM BOOK_LENDING
WHERE DATE_OUT BETWEEN “2017/01/03” AND “2017/06/26”
GROUP BY CARD_NO
HAVING COUNT(*)>=3;
```

c) Delete a book from Book table, update the contents of other tables to reflect this data manipulation operation.

```
Delete from Book
Where Book_id = 100;
```

Updating those tables which are related to “Book” table:

```
Select *
From Book_authors;

Select *
From Publisher;

Select *
From Book_copies;

Select *
From Book_lending;

Select *
From Library_branch;
```

d) Create a view of all books and its number of copies that are currently available in the library.

Create view book_avail as
Select title, no_of_copies
From Book, Book_copies, Library_branch
Where Book_copies.branch_id = Library_branch.branch_id and Book.book_id = Book_copies.book_id;

Exercise 3: Consider the following schema foe Order Database:

SALESMAN(Salesman_ id, Name, City, Commission)
CUSTOMER(Customer_ id, Customer_ name, Grade, Salesman_ id)
ORDERS(Order_ no, Purchase_ amount, Order_ date, Customer_ id, Salesman_ id)

a) List all salesman and indicate those have and don’t have customers in their cities(use UNION operation).
b) Update the age to NULL for Name = ‘rajeshwari’. Note that the NULL used here is MySQL NULL and not string NULL.
c) Retrieve the commission of salesman who has ‘oo’ in their names.
d) Among parent table and child table check for which table deletion is possible or not. Justify your answer.

Solution:

ER – Diagram:

Schema Digram:

Table creation:

CREATE TABLE SALESMAN(SALESMAN_ID INT PRIMARY KEY,NAME CHAR(20),CITY CHAR(20),COMMISSION REAL);

CREATE TABLE CUSTOMER(CUSTOMER_ID INT PRIMARY KEY,CUSTOMER_NAME CHAR(20),CITY CHAR(20),GRADE INT,SALESMAN_ID INT,FOREIGN KEY(SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE SET NULL);

CREATE TABLE ORDERS(ORDER_NO INT PRIMARY KEY,PURCHASE_AMOUNT REAL,ORDER_DATE DATE,CUSTOMER_ID INT,FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID),SALESMAN_ID INT,FOREIGN KEY(SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE CASCADE);

Insertion into table:

INSERT INTO SALESMAN

```
VALUES(1000,'BHUVANESH K','BANGLORE',12000),
(1001,'PRAGATHI','BANGLORE','15000'),
(1002,'VASUDEVAPRASAD','MANGLORE',10000),
(1003,'GURUPRASADKONDI','MYSORE@123',19000),
(1004,'SRISAI SALERS','MANDYA',25000),
(1005,'BOOKTRIBUTERS','HANUMANTHNAGAR',16200),
(1006,'SHOWROOMGALAXY','BIDAR',14900);

INSERT INTO CUSTOMER
VALUES(100,'SPOORTHIASHWIN','BIDAR',10,1000),
(101,'AMRUTHANUKASHYAP','MANGLORE',8,1000),
(102,'RAJESHWARI','MANGLORE',5,1002),
(103,'NISHAKGOWDA','MANGLORE',9,1001),
(104,'SATHWIK','MANDYA','10','1004'),
(105,'GURUDEVANAYAK','BIDAR',7,1002),
(106,'VASUMURTHI','MYSORE@123',9,1003);

INSERT INTO ORDERS
VALUES(10,23000,'2019-09-17',100,1000),
(11,26455,'2019-06-03',102,1001),
(12,30002,'2019-01-03',103,1002),
(13,17003,'2019-07-17',105,1003),
(14,45255,'2019-05-13',106,1004),
(15,32000,'2019-07-23',101,1000),
(16,27000,'2019-05-23',104,1002),
(17,26255,'2019-09-17',104,1000);
```

To list all salesmen and indicate those who have and don't have customers in their cities using the UNION, operation we need to perform following steps:

Step 1: Retrieve the salesmen who have customers in their cities.

```
SELECT S.Name AS SalesmanName, S.City AS SalesmanCity, 'Has Customers' AS Status
FROM SALESMAN S natural JOIN CUSTOMER C
where S.city = C.city;
```

Step 2: Retrieve the salesmen who don't have customers in their cities.

```
SELECT S.Name AS SalesmanName, S.City AS SalesmanCity, 'No Customers' AS Status
FROM SALESMAN S JOIN CUSTOMER C ON S.SALESMAN_ID = C.SALESMAN_ID
where S.City != C.City;
```

Step 3: Combine the results using the UNION operation to get a single result set.

```
SELECT S.Name AS SalesmanName, S.City AS SalesmanCity, 'Has Customers' AS Status
FROM SALESMAN S natural JOIN CUSTOMER C
where S.city = C.city
UNION
SELECT S.Name AS SalesmanName, S.City AS SalesmanCity, 'No Customers' AS Status
```

```
FROM SALESMAN S JOIN CUSTOMER C ON S.SALESMAN_ID = C.SALESMAN_ID
where S.City != C.City;
```

b) Update the city to NULL for Name = ‘rajeshwari’. Note that the NULL used here is MySQL NULL and not string NULL.

```
UPDATE CUSTOMER
SET city = NULL
WHERE CUSTOMER_NAME LIKE 'rajeshwari';
```

C) Retrieve the commission of salesman who has ‘oo’ in their names.

```
SELECT Commission
FROM SALESMAN
WHERE Name LIKE '%oo%';
```

d) Among parent table and child table check for which table deletion is possible or not. Justify your answer.

```
DROP TABLE salesman;

DROP TABLE customer;

DROP TABLE orders;
```