DBMS LAB PROGRAMS

1. 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, Programme_id, No-of_Copies)

BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date)

LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address)

Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each Programme, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5. Create a view of all books and its number of copies that are currently available in the Library.

Solution:

Entity-Relationship Diagram

Analyze the Relationships and Participation Constraints:

1. PUBLISHER and BOOK: **(1:N)**

Cardinality Ratio: One-to-Many (**1:N**) from PUBLISHER to BOOK. One publisher can publish many books, but each book has only one publisher.

Participation Constraints: Mandatory (Full participation) on the BOOK side. Every book must have a publisher.

2. BOOK and BOOK_AUTHORS: **(1:N)**

Cardinality Ratio: One-to-Many (**1:N**) from BOOK to BOOK_AUTHORS. One book can have multiple authors, but each author is associated with one book.

Participation Constraints: Mandatory on the BOOK side. Every book must have at least one author.

3. BOOK and BOOK_COPIES: (**1:N**)

Cardinality Ratio: One-to-Many (**1:N**) from BOOK to BOOK_COPIES. One book can have multiple copies in different library programmes, but each copy is associated with one book.

Participation Constraints: Mandatory on the BOOK side. Every book must have at least one copy in a library programme.

4. LIBRARY_PROGRAMME and BOOK_COPIES: (**1:N**)

Cardinality Ratio: One-to-Many (**1:N**) from LIBRARY_PROGRAMME to BOOK_COPIES. One library programme can have multiple copies of different books, but each copy is associated with one library programme.

Participation Constraints: Mandatory on the BOOK_COPIES side. Every copy must be associated with a library programme.

5. BOOK_COPIES and BOOK_LENDING: (**M:N**)

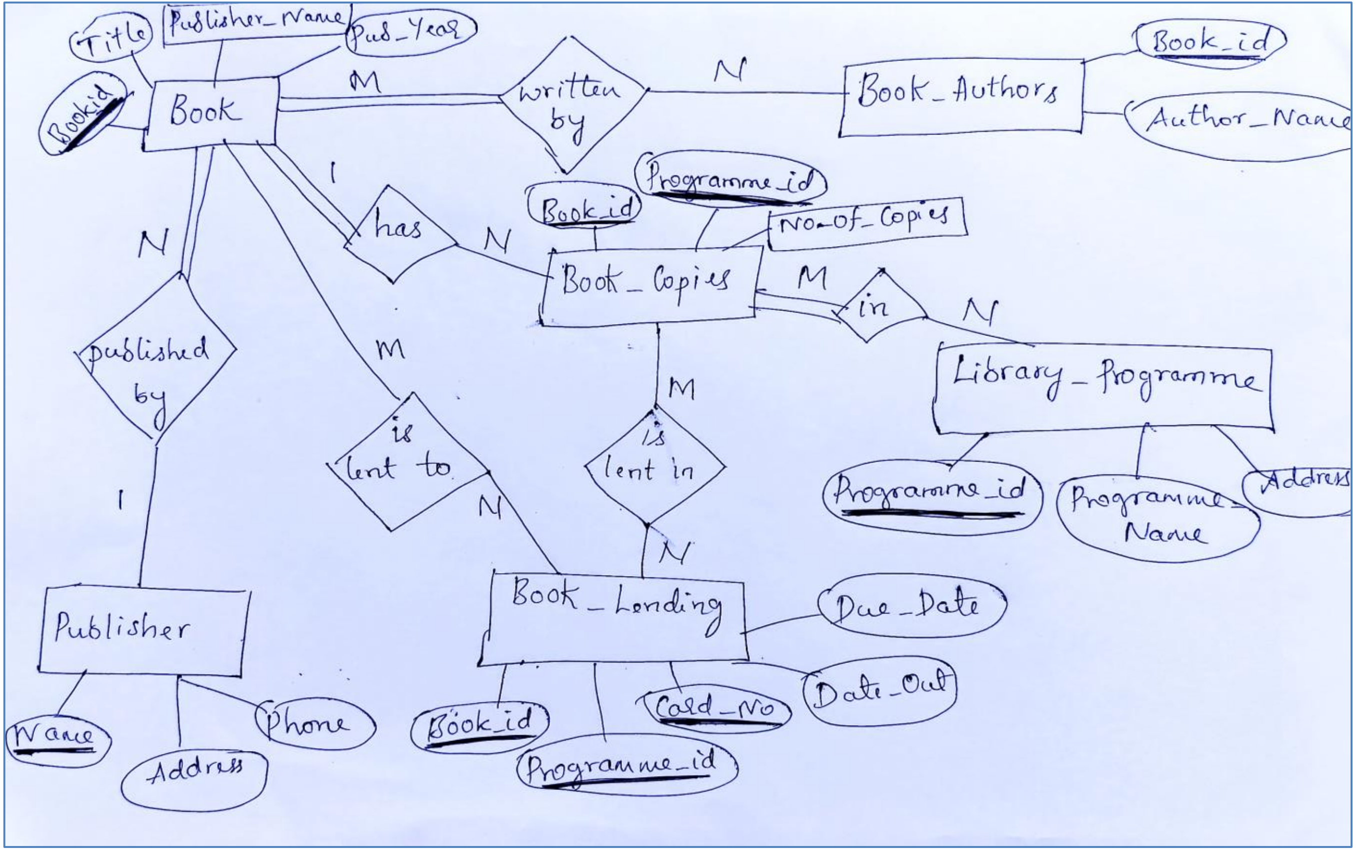
Cardinality Ratio: Many-to-Many (**M:N**) between BOOK_COPIES and BOOK_LENDING. Each book copy can be lent to multiple library programmes, and each library programme can borrow multiple book copies.

Participation Constraints: Optional (Partial Participation) on both sides. Not every book copy needs to be lent, and not every library programme needs to borrow a book copy.

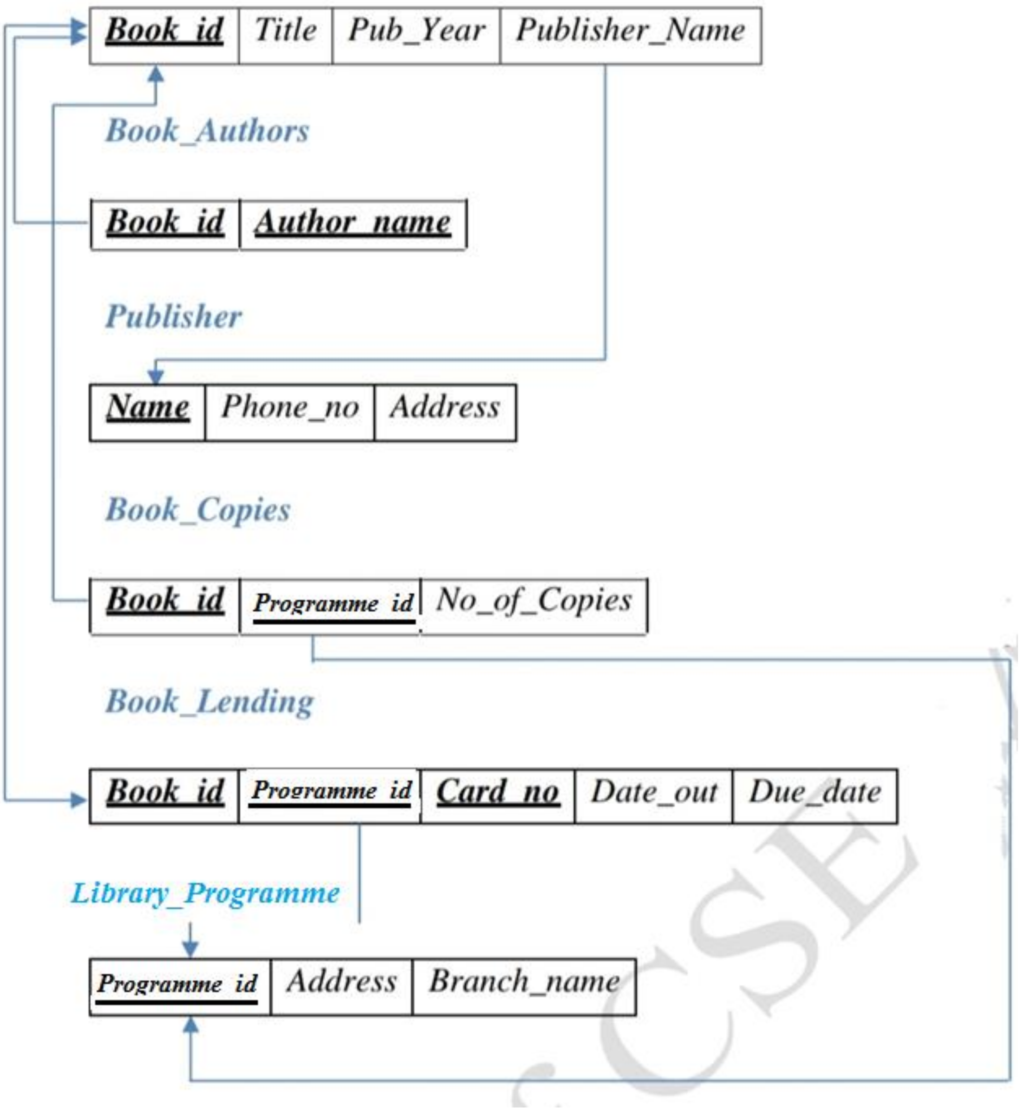
6. BOOK and BOOK_LENDING: (**M:N**)

Cardinality Ratio: Many-to-Many **(M:N)** between BOOK and BOOK_LENDING. Each book can be lent to multiple library programmes, and each library programme can borrow multiple books.

Participation Constraints: Optional on both sides. Not every book needs to be lent, and not every library programme needs to borrow a book.


Title
Publisher_Name
Pub_Year
Bookid
Book
M
written by
N
Book_Authors
Book_id
Author_Name
1
has
N
Programme_id
Book_id
No_of_Copies
Book_Copies
M
in
N
Library_Programme
N
published by
1
M
is lent to
N
M
is lent in
N
Programme_id
Programme_Name
Address
Publisher
Name
Address
Phone
Book_Lending
Due_Date
Date_Out
Book_id
Programme_id
Card_No

Schema Diagram



Step 1: Create Database

```
create database Library;
use Library;
```

Step 2: Create Tables

```
CREATE TABLE PUBLISHER(
NAME VARCHAR(18) PRIMARY KEY,
ADDRESS VARCHAR(10),
PHONE VARCHAR(10));

CREATE TABLE BOOK(
BOOK_ID INTEGER PRIMARY KEY,
TITLE VARCHAR(20),
PUBLISHER_NAME VARCHAR(20),
PUB_YEAR INT(4),
FOREIGN KEY(PUBLISHER_NAME) REFERENCES PUBLISHER(NAME) ON
DELETE CASCADE
);

CREATE TABLE BOOK_AUTHORS(
BOOK_ID INTEGER,
AUTHOR_NAME VARCHAR(20),
PRIMARY KEY(BOOK_ID),
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE);

CREATE TABLE LIBRARY_PROGRAMME(
PROGRAMME_ID INTEGER PRIMARY KEY,
PROGRAMME_NAME VARCHAR(18),
ADDRESS VARCHAR(15));

CREATE TABLE BOOK_COPIES(
BOOK_ID INTEGER,
PROGRAMME_ID INTEGER,
NO_OF_COPIES INTEGER,
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY(PROGRAMME_ID) REFERENCES LIBRARY_PROGRAMME(PROGRAMME_ID) ON
DELETE CASCADE,
PRIMARY KEY(BOOK_ID,PROGRAMME_ID));
```

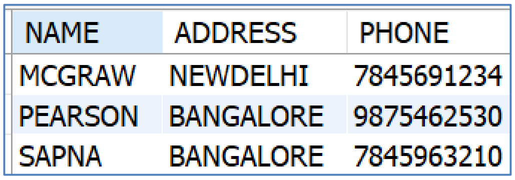
```
CREATE TABLE BOOK_LENDING(
BOOK_ID INTEGER,
PROGRAMME_ID INTEGER,
CARD_NO INTEGER,
DATE_OUT DATE,
DUE_DATE DATE,
PRIMARY KEY(BOOK_ID,PROGRAMME_ID,CARD_NO),
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY(PROGRAMME_ID) REFERENCES LIBRARY_PROGRAMME(PROGRAMME_ID) ON
DELETE CASCADE
);
```

Step 3: Insert Values into Tables

```
INSERT INTO PUBLISHER VALUES('PEARSON','BANGALORE','9875462530');
INSERT INTO PUBLISHER VALUES('MCGRAW','NEWDELHI','7845691234');
INSERT INTO PUBLISHER VALUES('SAPNA','BANGALORE','7845963210');

INSERT INTO BOOK VALUES(1111,'SE','PEARSON',2005);
INSERT INTO BOOK VALUES(2222,'DBMS','MCGRAW',2004);
INSERT INTO BOOK VALUES(3333,'ANOTOMY','PEARSON',2010);
INSERT INTO BOOK VALUES(4444,'ENCYCLOPEDIA','SAPNA',2010);

INSERT INTO BOOK_AUTHORS VALUES(1111,'SOMMERVILLE');
INSERT INTO BOOK_AUTHORS VALUES(2222,'NAVATHE');
INSERT INTO BOOK_AUTHORS VALUES(3333,'HENRY GRAY');
INSERT INTO BOOK_AUTHORS VALUES(4444,'THOMAS');

INSERT INTO LIBRARY_PROGRAMME VALUES(11,'CENTRAL TECHNICAL','MG
ROAD');
INSERT INTO LIBRARY_PROGRAMME VALUES(22,'MEDICAL','BH ROAD');
INSERT INTO LIBRARY_PROGRAMME VALUES(33,'CHILDREN','SS PURAM');
INSERT INTO LIBRARY_PROGRAMME VALUES(44,'SECRETARIAT','SIRAGATE');
INSERT INTO LIBRARY_PROGRAMME VALUES(55,'GENERAL','JAYANAGAR');

INSERT INTO BOOK_COPIES VALUES(1111,11,5);
INSERT INTO BOOK_COPIES VALUES(3333,22,6);
INSERT INTO BOOK_COPIES VALUES(4444,33,10);
INSERT INTO BOOK_COPIES VALUES(2222,11,12);
INSERT INTO BOOK_COPIES VALUES(4444,55,3);
INSERT INTO BOOK_LENDING VALUES(2222,11,1,'2017-01-10','2017-08-20');
INSERT INTO BOOK_LENDING VALUES(3333,22,2,'2017-07-09','2017-08-12');
INSERT INTO BOOK_LENDING VALUES(4444,55,1,'2017-04-11','2017-08-09');
INSERT INTO BOOK_LENDING VALUES(2222,11,5,'2017-08-09','2017-08-19');
INSERT INTO BOOK_LENDING VALUES(4444,33,1,'2017-06-10','2017-08-15');
INSERT INTO BOOK_LENDING VALUES(1111,11,1,'2017-05-12','2017-06-10');
INSERT INTO BOOK_LENDING VALUES(3333,22,1,'2017-07-10','2017-07-15');
```

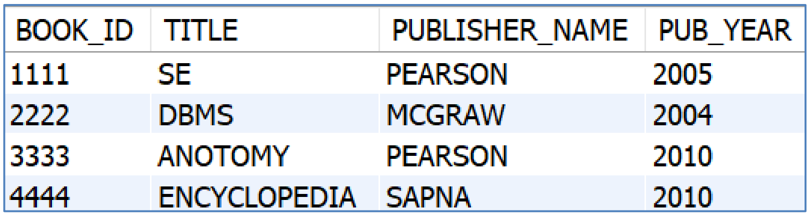
Step 4: Display table contents

```
SELECT * FROM BOOK;
```



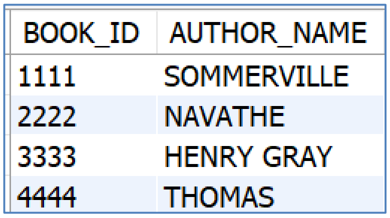
NAME	ADDRESS	PHONE
MCGRAW	NEWDELHI	7845691234
PEARSON	BANGALORE	9875462530
SAPNA	BANGALORE	7845963210

```
SELECT * FROM BOOK;
```



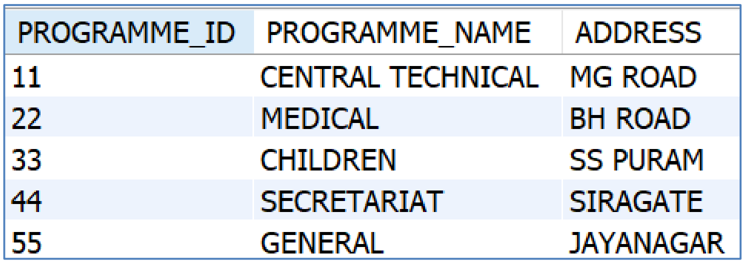
BOOK_ID	TITLE	PUBLISHER_NAME	PUB_YEAR
1111	SE	PEARSON	2005
2222	DBMS	MCGRAW	2004
3333	ANOTOMY	PEARSON	2010
4444	ENCYCLOPEDIA	SAPNA	2010

```
SELECT * FROM BOOK_AUTHORS;
```



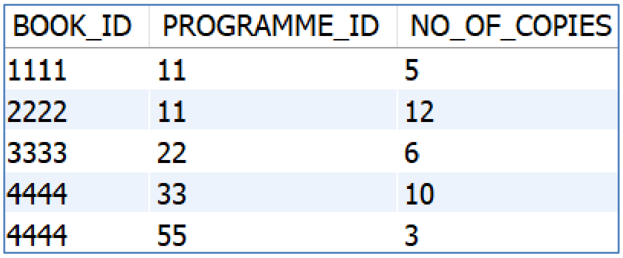
BOOK_ID	AUTHOR_NAME
1111	SOMMERVILLE
2222	NAVATHE
3333	HENRY GRAY
4444	THOMAS

```
SELECT * FROM LIBRARY_PROGRAMME;
```



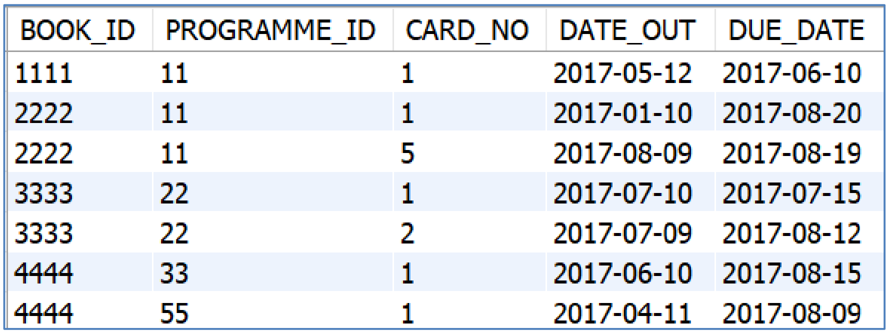
PROGRAMME_ID	PROGRAMME_NAME	ADDRESS
11	CENTRAL TECHNICAL	MG ROAD
22	MEDICAL	BH ROAD
33	CHILDREN	SS PURAM
44	SECRETARIAT	SIRAGATE
55	GENERAL	JAYANAGAR

```
SELECT * FROM BOOK_COPIES;
```



BOOK_ID	PROGRAMME_ID	NO_OF_COPIES
1111	11	5
2222	11	12
3333	22	6
4444	33	10
4444	55	3

```
SELECT * FROM BOOK_LENDING;
```



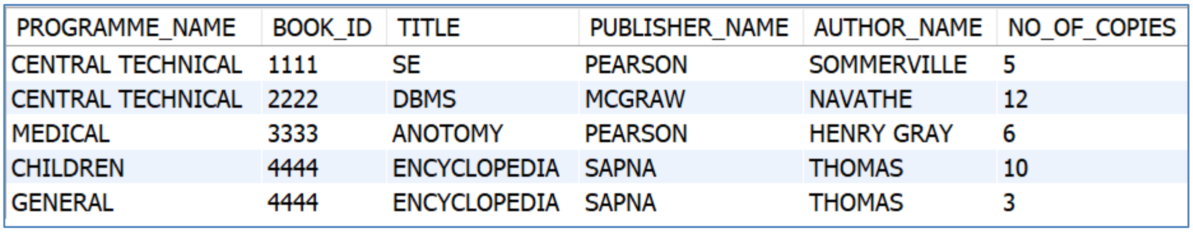
BOOK_ID	PROGRAMME_ID	CARD_NO	DATE_OUT	DUE_DATE
1111	11	1	2017-05-12	2017-06-10
2222	11	1	2017-01-10	2017-08-20
2222	11	5	2017-08-09	2017-08-19
3333	22	1	2017-07-10	2017-07-15
3333	22	2	2017-07-09	2017-08-12
4444	33	1	2017-06-10	2017-08-15
4444	55	1	2017-04-11	2017-08-09

Step 5: Execute Queries:

/* 1) Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each PROGRAMME, etc.*/

```
SELECT PROGRAMME_NAME, B.BOOK_ID,TITLE,
PUBLISHER_NAME,AUTHOR_NAME, NO_OF_COPIES
FROM BOOK B, BOOK_AUTHORS BA, BOOK_COPIES BC,
LIBRARY_PROGRAMME LB WHERE B.BOOK_ID = BA.BOOK_ID AND
BA.BOOK_ID = BC.BOOK_ID AND
BC.PROGRAMME_ID = LB.PROGRAMME_ID;
```

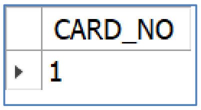
Output:



PROGRAMME_NAME	BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES
CENTRAL TECHNICAL	1111	SE	PEARSON	SOMMERVILLE	5
CENTRAL TECHNICAL	2222	DBMS	MCGRAW	NAVATHE	12
MEDICAL	3333	ANOTOMY	PEARSON	HENRY GRAY	6
CHILDREN	4444	ENCYCLOPEDIA	SAPNA	THOMAS	10
GENERAL	4444	ENCYCLOPEDIA	SAPNA	THOMAS	3

```
/* 2) 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 '2017-01-01' AND '2017-06-30'
GROUP BY CARD_NO
HAVING COUNT(*) > 3;
```

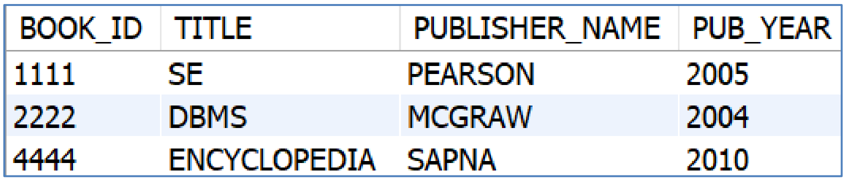
Output:



CARD_NO
1

```
/* 3) Delete a book in BOOK table. Update the contents of other tables to
reflect this data manipulation operation. */
DELETE FROM BOOK
WHERE BOOK_ID = '3333';

SELECT * FROM BOOK;
```

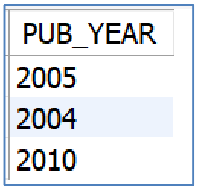
Output:



BOOK_ID	TITLE	PUBLISHER_NAME	PUB_YEAR
1111	SE	PEARSON	2005
2222	DBMS	MCGRAW	2004
4444	ENCYCLOPEDIA	SAPNA	2010

```
/* 4) Partition the BOOK table based on year of publication. Demonstrate
its working with a simple query. */
CREATE VIEW V_PUBLICATION AS
SELECT PUB_YEAR
FROM BOOK;

SELECT * FROM V_PUBLICATION;
```

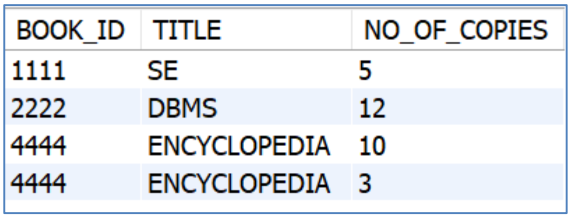
Output:



PUB_YEAR
2005
2004
2010

```
/* 5) Create a view of all books and its number of copies that are
currently available in the Library. */

CREATE VIEW BOOKS_AVAILABLE AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM LIBRARY_PROGRAMME L, BOOK B, BOOK_COPIES C
WHERE B.BOOK_ID = C.BOOK_ID AND L.PROGRAMME_ID=C.PROGRAMME_ID;

SELECT * FROM BOOKS_AVAILABLE;
```

Output:



BOOK_ID	TITLE	NO_OF_COPIES
1111	SE	5
2222	DBMS	12
4444	ENCYCLOPEDIA	10
4444	ENCYCLOPEDIA	3

****

2. Consider the following schema for Order Database:

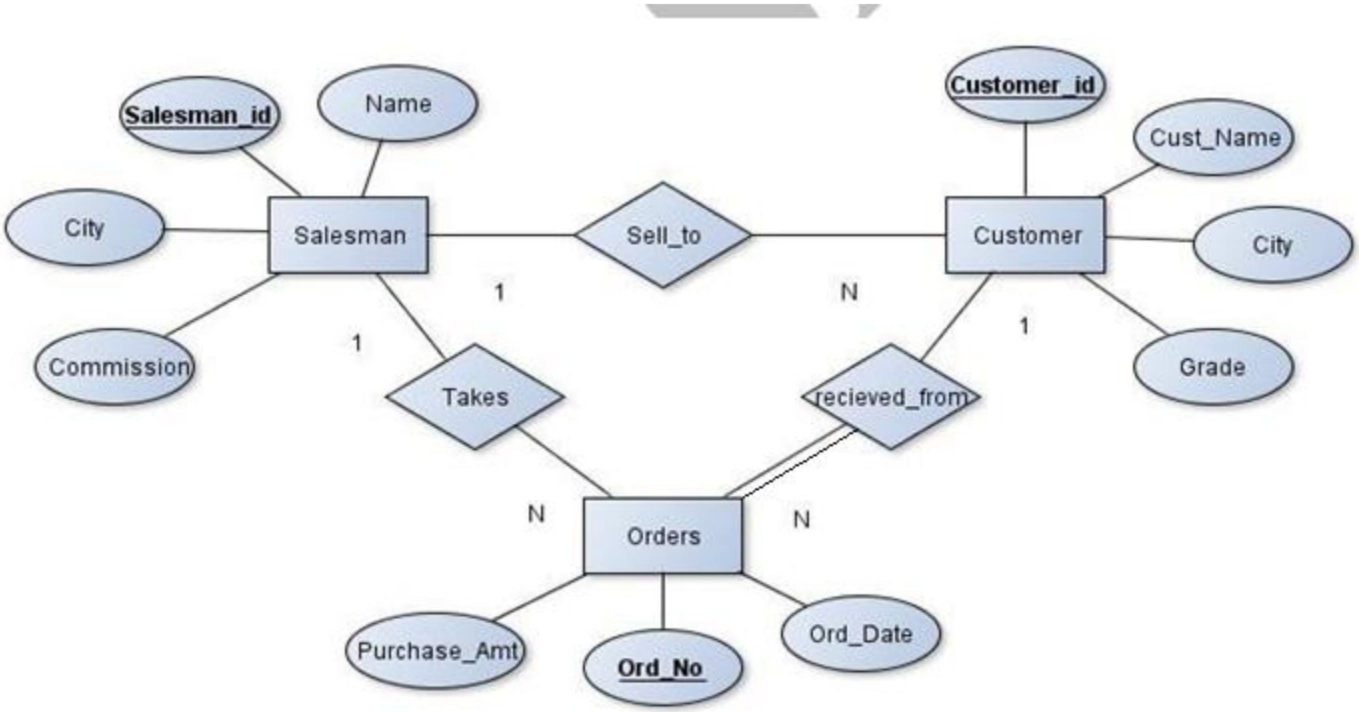
SALESMAN (*Salesman_id, Name, City, Commission*)
CUSTOMER (*Customer_id, Cust_Name, City, Grade,Salesman_id*)
ORDERS (*Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id*)

Write SQL queries to

1. **Count the customers with grades above Bangalore's average.**
2. **Find the name and numbers of all salesmen who had more than one customer.**
3. **List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)**
4. **Create a view that finds the salesman who has the customer with the highest order of a day.**
5. **Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.**

Solution:

Entity-Relationship Diagram



Analyze the Relationships and Participation Constraints:

1. SALESMAN and CUSTOMER: **(1:N)**

Cardinality Ratio: One-to-Many (**1:N**) from SALESMAN to CUSTOMER. One salesman can have multiple customers, but each customer is associated with one salesman.

Participation Constraints: Optional on the CUSTOMER side. Not every customer must be assigned to a salesman.

2. CUSTOMER and ORDERS: (**1:N**)

Cardinality Ratio: One-to-Many **(1:N)** from CUSTOMER to ORDERS. One customer can place multiple orders, but each order is associated with one customer.

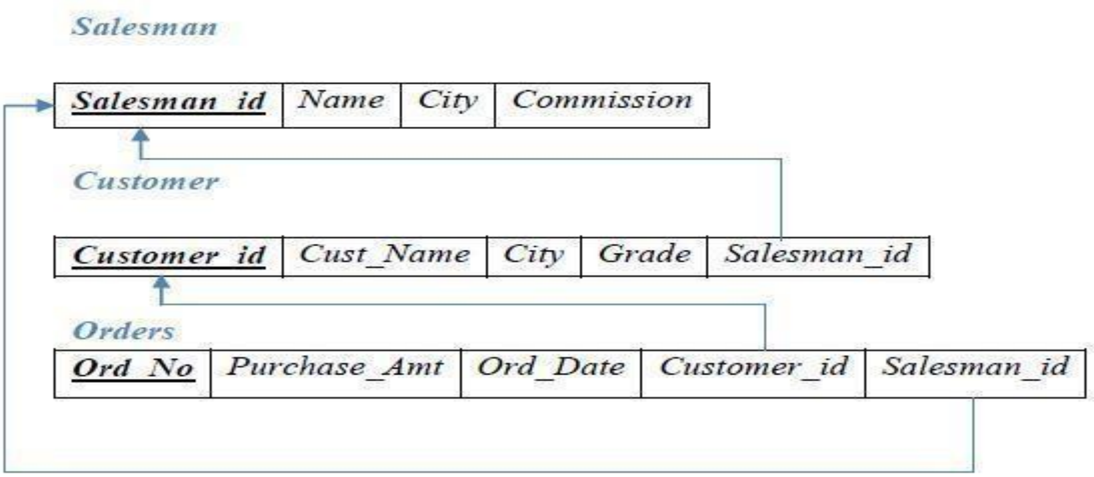
Participation Constraints: Mandatory on the ORDERS side. Every order must be associated with a customer.

3. SALESMAN and ORDERS: (**1:N**)

Cardinality Ratio: One-to-Many **(1:N)** from SALESMAN to ORDERS. One salesman can handle multiple orders, but each order is handled by one salesman.

Participation Constraints: Optional on the ORDERS side. Not every order must be assigned to a salesman.

Schema Diagram



Step 1: Create Database

```
CREATE DATABASE ORDERS11;
USE ORDERS11;
```

Step 2: Create Tables

```
CREATE TABLE SALESMAN (
SALESMAN_ID INT (4),
NAME VARCHAR (20),
CITY VARCHAR (20),
COMMISSION VARCHAR (20),
PRIMARY KEY(SALESMAN_ID));
```

```
CREATE TABLE CUSTOMER (
CUSTOMER_ID INT (4),
CUST_NAME VARCHAR (20),
CITY VARCHAR (20),
GRADE INT (3),
SALESMAN_ID INT (4),
PRIMARY KEY (CUSTOMER_ID),
FOREIGN KEY(SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE SET NULL);
```

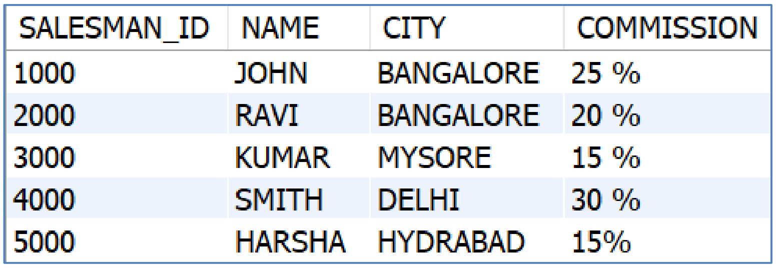
```
CREATE TABLE ORDERS (
ORD_NO INT(5),
PURCHASE_AMT FLOAT(10, 2),
ORD_DATE DATE,
CUSTOMER_ID INT (4),
SALESMAN_ID INT (4),
PRIMARY KEY (ORD_NO),
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID) ON DELETE CASCADE,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE CASCADE);
```

Step 3: Insert Values into Tables

```
INSERT INTO SALESMAN VALUES (1000, 'JOHN','BANGALORE','25 %');
INSERT INTO SALESMAN VALUES (2000, 'RAVI','BANGALORE','20 %');
INSERT INTO SALESMAN VALUES (3000, 'KUMAR','MYSORE','15 %');
INSERT INTO SALESMAN VALUES (4000, 'SMITH','DELHI','30 %');
INSERT INTO SALESMAN VALUES (5000, 'HARSHA','HYDRABAD','15%');

INSERT INTO CUSTOMER VALUES (10, 'PREETHI','BANGALORE', 100, 1000);
INSERT INTO CUSTOMER VALUES (11, 'VIVEK','MANGALORE', 300, 1000);
INSERT INTO CUSTOMER VALUES (12, 'BHASKAR','CHENNAI', 400, 2000);
INSERT INTO CUSTOMER VALUES (13, 'CHETHAN','BANGALORE', 200, 2000);
INSERT INTO CUSTOMER VALUES (14, 'MAMATHA','BANGALORE', 400, 3000);

INSERT INTO ORDERS VALUES (50, 5000, '2017-05-04', 10, 1000);
INSERT INTO ORDERS VALUES (55, 1000, '2017-05-04', 10, 1000);
INSERT INTO ORDERS VALUES (56, 300, '2017-05-04', 10, 2000);
INSERT INTO ORDERS VALUES (51, 450, '2017-01-20', 10, 2000);
INSERT INTO ORDERS VALUES (52, 1000, '2017-02-24', 13, 2000);
INSERT INTO ORDERS VALUES (53, 3500, '2017-04-13', 14, 3000);
INSERT INTO ORDERS VALUES (54, 550, '2017-03-09', 12, 2000);
INSERT INTO ORDERS VALUES (57, 450, '2017-03-09', 12, 2000);
INSERT INTO ORDERS VALUES (58, 350, '2017-03-09', 12, 2000);
INSERT INTO ORDERS VALUES (60, 150, '2017-03-09', 12, 1000);
INSERT INTO ORDERS VALUES (61, 200, '2017-03-09', 12, 3000);
```

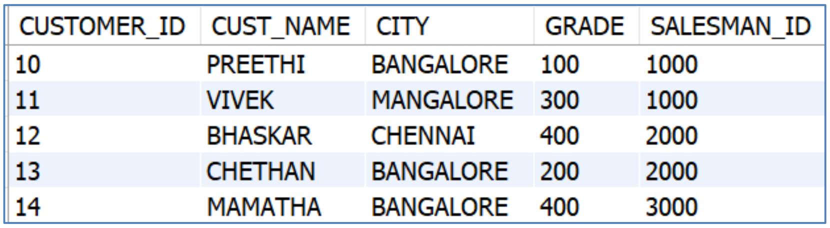
Step 4: Display table contents

```
select * from SALESMAN;
```



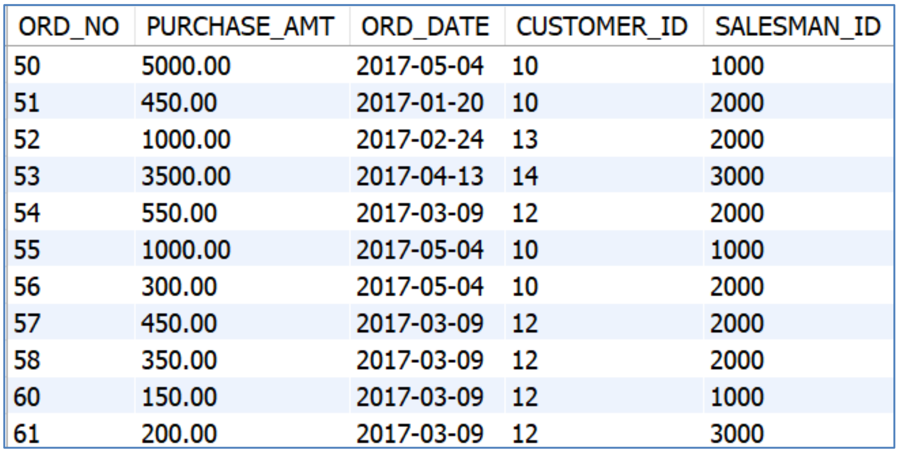
SALESMAN_ID	NAME	CITY	COMMISSION
1000	JOHN	BANGALORE	25 %
2000	RAVI	BANGALORE	20 %
3000	KUMAR	MYSORE	15 %
4000	SMITH	DELHI	30 %
5000	HARSHA	HYDRABAD	15%

select * from CUSTOMER;



CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
10	PREETHI	BANGALORE	100	1000
11	VIVEK	MANGALORE	300	1000
12	BHASKAR	CHENNAI	400	2000
13	CHETHAN	BANGALORE	200	2000
14	MAMATHA	BANGALORE	400	3000

select * from ORDERS;



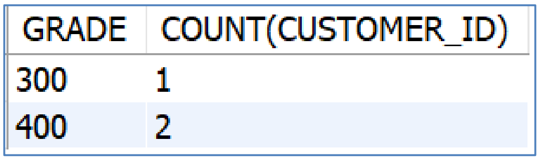
ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
50	5000.00	2017-05-04	10	1000
51	450.00	2017-01-20	10	2000
52	1000.00	2017-02-24	13	2000
53	3500.00	2017-04-13	14	3000
54	550.00	2017-03-09	12	2000
55	1000.00	2017-05-04	10	1000
56	300.00	2017-05-04	10	2000
57	450.00	2017-03-09	12	2000
58	350.00	2017-03-09	12	2000
60	150.00	2017-03-09	12	1000
61	200.00	2017-03-09	12	3000

Step 5: Execute Queries:

-- 1. Count the customers with grades above Bangalore's average.

```
SELECT GRADE, COUNT(CUSTOMER_ID)
FROM CUSTOMER
GROUP BY GRADE
HAVING GRADE >  (SELECT AVG(GRADE) FROM CUSTOMER WHERE CITY='BANGALORE');
```

Output:

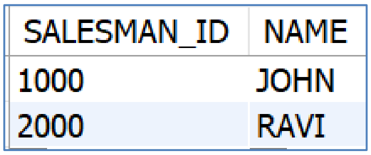


GRADE	COUNT(CUSTOMER_ID)
300	1
400	2

-- 2. Find the name and numbers of all salesmen who had more than one customer.

```
SELECT SALESMAN_ID, NAME
FROM SALESMAN A
WHERE 1 < (SELECT COUNT(*) FROM CUSTOMER WHERE SALESMAN_ID=A.SALESMAN_ID);
```

Output:

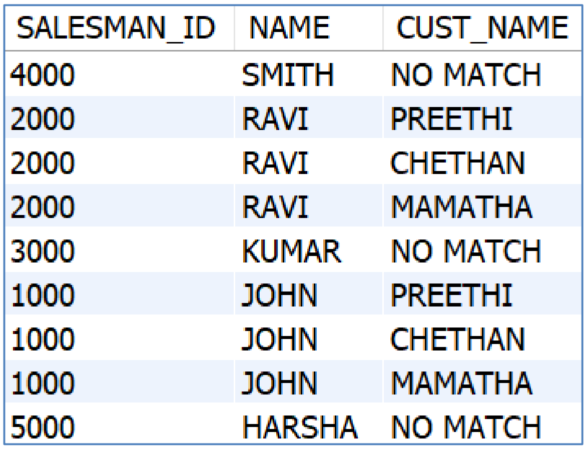


SALESMAN_ID	NAME
1000	JOHN
2000	RAVI

/* 3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation) */

```
SELECT SALESMAN.SALESMAN_ID, NAME, CUST_NAME
FROM SALESMAN, CUSTOMER
WHERE SALESMAN.CITY = CUSTOMER.CITY
UNION
SELECT SALESMAN_ID, NAME, 'NO MATCH'
FROM SALESMAN
WHERE NOT CITY = ANY (SELECT CITY FROM CUSTOMER) ORDER BY 2 DESC;
```

Output:

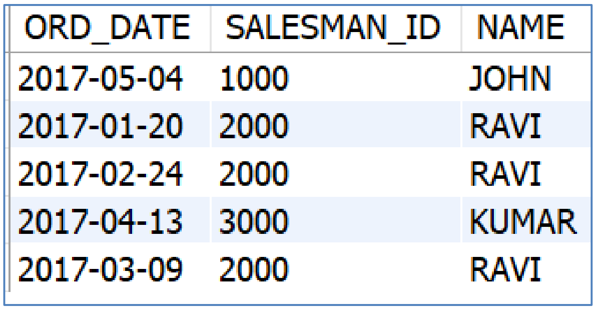


SALESMAN_ID	NAME	CUST_NAME
4000	SMITH	NO MATCH
2000	RAVI	PREETHI
2000	RAVI	CHETHAN
2000	RAVI	MAMATHA
3000	KUMAR	NO MATCH
1000	JOHN	PREETHI
1000	JOHN	CHETHAN
1000	JOHN	MAMATHA
5000	HARSHA	NO MATCH

-- 4. Create a view that finds the salesman who has the customer with the highest order of a day.

```
CREATE VIEW ELITESALESMAN AS
SELECT B.ORD_DATE, A.SALESMAN_ID, A.NAME
FROM SALESMAN A, ORDERS B
WHERE A.SALESMAN_ID = B.SALESMAN_ID AND B.PURCHASE_AMT=(SELECT
MAX(PURCHASE_AMT) FROM ORDERS C WHERE C.ORD_DATE = B.ORD_DATE);


select * from ELITESALESMAN;
```

Output:

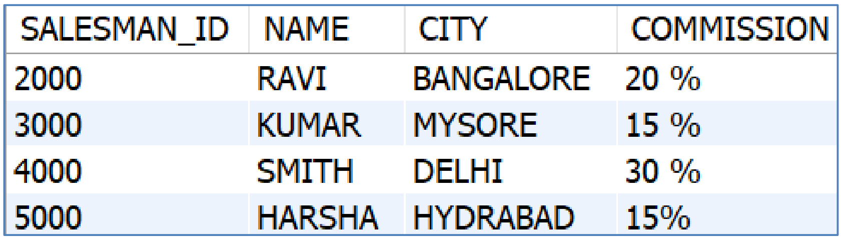


ORD_DATE	SALESMAN_ID	NAME
2017-05-04	1000	JOHN
2017-01-20	2000	RAVI
2017-02-24	2000	RAVI
2017-04-13	3000	KUMAR
2017-03-09	2000	RAVI

/* 5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted. */

```
DELETE FROM SALESMAN WHERE SALESMAN_ID=1000;

select * from SALESMAN;
```

Output:



SALESMAN_ID	NAME	CITY	COMMISSION
2000	RAVI	BANGALORE	20 %
3000	KUMAR	MYSORE	15 %
4000	SMITH	DELHI	30 %
5000	HARSHA	HYDRABAD	15%

****