EXPERIMENT No. 01: Library Database

1.1 Experiment Details:

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)

Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, 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

1.2 Softwares Required:

Operating System: Windows 7

DBMS: Oracle 10g

1.3 Pre-Requisite:

Basics of DBMS, SQL Programming

1.4 Procedure:

TABLE CREATION:

```
CREATE TABLE PUBLISHER (
NAME CHAR (15),
ADDRESS VARCHAR (20),
PHONE VARCHAR (10),
PRIMARY KEY (NAME));

CREATE TABLE LIBRARY_BRANCH (
BRANCH_ID VARCHAR (5),
BRANCH_NAME CHAR (15),
ADDRESS VARCHAR (20),
PRIMARY KEY (BRANCH_ID));

CREATE TABLE BORROWERS (
 CARD_NO VARCHAR (10),
 NAME CHAR (15),
 ADDRESS VARCHAR (20),
 PHONE VARCHAR (10),
 PRIMARY KEY (CARD_NO));

CREATE TABLE BOOK (
BOOK_ID VARCHAR (5),
TITLE CHAR (20),
PUBLISHER_NAME CHAR (15),
PUB_YEAR INT,
PRIMARY KEY (BOOK_ID),
FOREIGN KEY (PUBLISHER_NAME) REFERENCES PUBLISHER (NAME)
ON DELETE CASCADE)
PARTITION BY RANGE (PUB_YEAR)
(PARTITION P VALUES LESS THAN (2019));
```

```
CREATE TABLE AUTHORS (
BOOK_ID VARCHAR (5),
AUTHOR_NAME CHAR (15),
PRIMARY KEY (BOOK_ID),
FOREIGN KEY (BOOK_ID) REFERENCES BOOK (BOOK_ID)
ON DELETE CASCADE);

CREATE TABLE BOOK_COPIES (
BOOK_ID VARCHAR (5),
BRANCH_ID VARCHAR (5),
NO_OF_COPIES INT,
PRIMARY KEY (BOOK_ID, BRANCH_ID),
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_LENDING (
BOOK_ID VARCHAR (5),
BRANCH_ID VARCHAR (5),
CARD_NO VARCHAR (10),
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,
FOREIGN KEY (CARD_NO) REFERENCES BORROWERS (CARD_NO)
ON DELETE CASCADE);

COMMIT;
Commit complete.
```

INSERTION OF VALUES:

```
SQL> INSERT INTO PUBLISHER VALUES ('&NAME','&ADDRESS','&PHONE');
Enter value for name: SUDHA MOORTHY
Enter value for address: BANGLORE
Enter value for phone: 8778960098
Old   1: INSERT INTO PUBLISHER VALUES ('&NAME','&ADDRESS','&PHONE')
New   1: INSERT INTO PUBLISHER VALUES ('SUDHA
MOORTHY','BANGLORE','8778960098')

SQL> SELECT * FROM PUBLISHER;

NAME                    ADDRESS              PHONE
---------------         ------------------   ----------
SUDHA MOORTHY           BANGLORE            8778960098
ELITE                   MANGLORE            7887809899
ZEN                     MUMBAI              9009876435
APTIMITHRA              CHENNAI             7665434261
EXPRESS                 BANGLORE             8776645321


SQL> INSERT INTO BOOK VALUES
('&BOOK_ID','&TITLE','&PUBLISHER_NAME','&PUB_YEAR');
Enter value for book_id: BK1
Enter value for title: HELLO
Enter value for publisher_name: ELITE
Enter value for pub_year: 2012
Old   1: INSERT INTO BOOK VALUES
('&BOOK_ID','&TITLE','&PUBLISHER_NAME','&PUB_YEAR')
New   1: INSERT INTO BOOK VALUES ('BK1','HELLO','ELITE','2012')
```

```
SQL> SELECT * FROM BOOK;

BOOK_ID     TITLE                    PUBLISHER_NAME   PUB_YEAR
-----       --------------------     ---------------  ----------
BK1         HELLO                    ELITE                2012
BK2         THE WINGS OF FIRE        ELITE                2013
BK3         YOUNG INDIA              ZEN                  2010
BK4         10 SECONDS               APTIMITHRA           2000
BK5         APTI MITHRA              APTIMITHRA           2009
BK6         OLIVE BOARD              APTIMITHRA           2018
BK7         PASSWORD                 EXPRESS              2010


SQL> INSERT INTO AUTHORS VALUES ('& BOOK_ID ','&AUTHOR_NAME');
Enter value for book_id: BK1
Enter value for author_name: RAMAN
Old   1: INSERT INTO AUTHORS VALUES ('& BOOK_ID ','&AUTHOR_NAME')
New   1: INSERT INTO AUTHORS VALUES ('BK1','RAMAN')

SQL> SELECT * FROM AUTHORS;

BOOK_ID     AUTHOR_NAME
-----       ---------------
BK1         RAMAN
BK2         ABDUL KALAM
BK3         CHETHAN
BK4         ABHISHEK
BK5         ABHISHEK
BK6         SAHANA
BK7         SUDHA MOORTHY
```

```
SQL> INSERT INTO LIBRARY_BRANCH VALUES
('&BRANCH_ID','&BRANCH_NAME','&ADDRESS');
Enter value for branch_id: B1
Enter value for branch_name: AKRITHI
Enter value for address: MANGLORE
Old   1: INSERT INTO LIBRARY_BRANCH VALUES
('&BRANCH_ID','&BRANCH_NAME','&ADDRESS')
New   1: INSERT INTO LIBRARY_BRANCH VALUES
('B1','AKRITHI','MANGLORE')

SQL> SELECT * FROM LIBRARY_BRANCH;

BRANCH_ID          BRANCH_NAME     ADDRESS
-----              ---------------     --------------------
B1                 AKRITHI             MANGLORE
B2                  SANKALP            BANGLORE
B3                 THE BEST            KOLKATTA
B4                 SAHYADRI            MANGLORE
B5                 AGNES               CHENNAI

SQL> SET LINESIZE 200 PAGESIZE 3000;

SQL> INSERT INTO BORROWERS VALUES
('&CARD_NO','&NAME','&ADDRESS','&PHONE');
Enter value for card_no: C1
Enter value for name: SHRITHI
Enter value for address: MERAMAJALU
Enter value for phone: 7338527841
Old   1: INSERT INTO BORROWERS VALUES
('&CARD_NO','&NAME','&ADDRESS','&PHONE')
New   1: INSERT INTO BORROWERS VALUES
('C1','SHRITHI','MERAMAJALU','7338527841')
```

```
SQL> SELECT * FROM BORROWERS;

CARD_NO    NAME             ADDRESS              PHONE
----------    ---------------    --------------------       ----------
C1              SHRITHI        MERAMAJALU       7338527841
C2              NAVYA          KONAJE               7665689076
C3              JOHN            MILAGRIS            9887888705
C4              KAVANA       VITTAL                6544332167
C5             VARSHINI       BANTWAL           8765432178

SQL> INSERT INTO BOOK_COPIES VALUES
('&BOOK_ID','&BRANCH_ID','&NO_OF_COPIES');
Enter value for book_id: BK1
Enter value for branch_id: B1
Enter value for no_of_copies: 5
Old   1: INSERT INTO BOOK_COPIES VALUES
('&BOOK_ID','&BRANCH_ID','&NO_OF_COPIES')
New   1: INSERT INTO BOOK_COPIES VALUES ('BK1','B1','5')

SQL> SELECT * FROM BOOK_COPIES;

BOOK_ ID      BRANCH_ID     NO_OF_COPIES
-----              -----                 ------------
BK1             B1                          5
BK2             B2                          4
BK2             B3                          6
BK3             B3                          2
BK4             B4                          6
BK5             B4                          3
BK6             B4                          3
BK7             B5                          2
```

```
SQL> INSERT INTO BOOK_LENDING VALUES
('&BOOK_ID','&BRANCH_ID','&CARD_NO','&DATE_OUT','&DUE_DATE');
Enter value for book_id: BK1
Enter value for branch_id: B1
Enter value for card_no: C1
Enter value for date_out: 4-JAN-2017
Enter value for due_date: 14-JAN-2017
Old   1: INSERT INTO BOOK_LENDING VALUES
('&BOOK_ID','&BRANCH_ID','&CARD_NO','&DATE_OUT','&DUE_DATE')
New   1: INSERT INTO BOOK_LENDING VALUES ('BK1','B1','C1','4-JAN-2017','14-
JAN-2017')

SQL> SELECT * FROM BOOK_LENDING;

BOOK__ID    BRANCH__ID       CARD_NO    DATE_OUT     DUE_DATE
-----        -----            ----------  ---------     ---------
BK1         B1               C1          04-JAN-17    14-JAN-17
BK4         B4               C1          05-MAY-17    20-MAY-17
BK5         B4               C1          03-FEB-17    18-FEB-17
BK6         B4               C1          22-JUN-17    29-JUN-17
BK2         B2               C2          12-JUN-17    22-JUN-17
BK3         B3               C3          23-MAY-18    30-MAY-18
BK7         B5               C4          23-JAN-19    11-FEB-19
BK7         B5               C5          12-JUN-19    22-JUN-19


SQL> COMMIT;

Commit complete.
```

1.5 RESULTS

QUERY 1:

```
SQL> SELECT A.BOOK_ID, A.TITLE, A.PUBLISHER_NAME, B.AUTHOR_NAME,
C.BRANCH_ID, C.NO_OF_COPIES
FROM BOOK A, AUTHORS B, BOOK_COPIES C WHERE
A.BOOK_ID=B.BOOK_ID AND A.BOOK_ID=C.BOOK_ID;
```

```
BOOK_ID   BOOK_ TITLE           PUBLISHER_NAME    AUTHOR_NAME      BRANCH_ID   NO_OF_COPIES
-----     --------------------       ---------------     ---------------           -----         ------------
BK1       FAMOUS FIVE          SUDHA MOORTHY   SUDHA MOORTHY     B1           5
BK2       THE WINGS OF FIRE    ELITE            ABDUL KALAM       B2           4
BK2       THE WINGS OF FIRE    ELITE            ABDUL KALAM       B3           6
BK3       YOUNG INDIA          ZEN              CHETHAN           B3           2
BK4       10 SECONDS           APTIMITHRA       ABHISHEK          B4           6
BK5       APTI MITHRA          APTIMITHRA       ABHISHEK          B4           3
BK6       OLIVE BOARD          APTIMITHRA       SAHANA            B4           3
BK7       PASSWORD             EXPRESS          SUDHA MOORTHY     B5           2

8 rows selected.
```

QUERY 2:

```
SQL> SELECT CARD_NO, NAME, ADDRESS, PHONE FROM BORROWERS
WHERE CARD_NO IN
(SELECT DISTINCT CARD_NO FROM BOOK_LENDING
WHERE DATE_OUT BETWEEN '01-JAN-2017' AND '30-JUN-2017'
GROUP BY (CARD_NO) HAVING COUNT (DATE_OUT)>3);
```

```
CARD_NO    NAME           ADDRESS              PHONE
----------      ---------------   --------------------        ----------
C1             SHRITHI      MERAMAJALU      7338527841
```

QUERY 3:

```
SQL> SELECT * FROM BOOK;

BOOK_ID      TITLE                      PUBLISHER_NAME   PUB_YEAR
-----       --------------------         ---------------      ----------
BK1         HELLO                       ELITE                  2012
BK2         THE WINGS OF FIRE           ELITE                  2013
BK3         YOUNG INDIA                 ZEN                    2010
BK4         10 SECONDS                  APTIMITHRA             2000
BK5         APTI MITHRA                 APTIMITHRA             2009
BK6         OLIVE BOARD                 APTIMITHRA             2018
BK7         PASSWORD                    EXPRESS                2010

SQL> SELECT * FROM AUTHORS;

BOOK_ID     AUTHOR_NAME
-----       ---------------
BK1         RAMAN
BK2         ABDUL KALAM
BK3         CHETHAN
BK4         ABHISHEK
BK5         ABHISHEK
BK6         SAHANA
BK7         SUDHA MOORTHY

SQL> SELECT * FROM BOOK_COPIES;

BOOK_ ID     BRANCH_ID    NO_OF_COPIES
-----         -----          ------------
BK1         B1                5
BK2         B2                4
BK2         B3                6
```

```
BK3          B3                  2
BK4          B4                  6
BK5          B4                  3
BK6          B4                  3
BK7          B5                  2

SQL> SELECT * FROM BOOK_LENDING;

BOOK__ID     BRANCH__ID          CARD_NO    DATE_OUT     DUE_DATE
-----         -----               ----------  ---------    ---------
BK1          B1                  C1          04-JAN-17    14-JAN-17
BK4          B4                  C1          05-MAY-17    20-MAY-17
BK5          B4                  C1          03-FEB-17    18-FEB-17
BK6          B4                  C1          22-JUN-17    29-JUN-17
BK2          B2                  C2          12-JUN-17    22-JUN-17
BK3          B3                  C3          23-MAY-18    30-MAY-18
BK7          B5                  C4          23-JAN-19    11-FEB-19
BK7          B5                  C5          12-JUN-19    22-JUN-19
```

SQL> DELETE FROM BOOK WHERE BOOK_ID='BK1';

```
1 row deleted.

SQL> SELECT * FROM BOOK;

BOOK_ID       TITLE                     PUBLISHER_NAME   PUB_YEAR
-----        --------------------        ---------------       ----------
BK2          THE WINGS OF FIRE         ELITE                  2013
BK3          YOUNG INDIA               ZEN                    2010
BK4          10 SECONDS                APTIMITHRA             2000
BK5          APTI MITHRA               APTIMITHRA             2009
BK6          OLIVE BOARD               APTIMITHRA             2018
BK7          PASSWORD                  EXPRESS                2010
```

```
SQL> SELECT * FROM AUTHORS;

BOOK_ID      AUTHOR_NAME
-----        ---------------
BK2          ABDUL KALAM
BK3          CHETHAN
BK4          ABHISHEK
BK5          ABHISHEK
BK6          SAHANA
BK7          SUDHA MOORTHY

SQL> SELECT * FROM BOOK_COPIES;

BOOK_ ID     BRANCH_ID          NO_OF_COPIES
-----        -----              ------------
BK2          B2                 4
BK2          B3                 6
BK3          B3                 2
BK4          B4                 6
BK5          B4                 3
BK6          B4                 3
BK7          B5                 2

SQL> SELECT * FROM BOOK_LENDING;
BOOK_ID BRANCH_ID CARD_NO      DATE_OUT    DUE_DATE
-----        -----     ----------      ---------    ---------
BK4          B4           C1           05-MAY-17   20-MAY-17
BK5          B4           C1           03-FEB-17   18-FEB-17
BK6          B4           C1           22-JUN-17   29-JUN-17
BK2          B2           C2           12-JUN-17   22-JUN-17
BK3          B3           C3           23-MAY-18   30-MAY-18
BK7          B5           C4           23-JAN-19   11-FEB-19
BK7          B5           C5           12-JUN-19   22-JUN-19
```

QUERY 4:

```
SQL> INSERT INTO BOOK VALUES
('&BOOK_ID','&TITLE','&PUBLISHER_NAME','&PUB_YEAR');
Enter value for book_id: BK8
Enter value for title: HELLO
Enter value for publisher_name: CHAITHANYA
Enter value for pub_year: 2030
Old   1: INSERT INTO BOOK VALUES
('&BOOK_ID','&TITLE','&PUBLISHER_NAME','&PUB_YEAR')
New   1: INSERT INTO BOOK VALUES ('BK8','HELLO','CHAITHANYA','2030')

INSERT INTO BOOK VALUES ('BK8','HELLO','CHAITHANYA','2030')
         *
ERROR at line 1:
ORA-14400: inserted partition key does not map to any partition
```

QUERY 5:

```
SQL> CREATE VIEW BCL AS
SELECT A.BOOK_ID, A.TITLE, B.BRANCH_ID, B.NO_OF_COPIES
FROM BOOK A, BOOK_COPIES B WHERE A.BOOK_ID=B.BOOK_ID;

View created.

SQL> SELECT * FROM BCL;

BOOK_ID    TITLE                 BRANCH_ID NO_OF_COPIES
-----      --------------------  -----     ------------
BK2        THE WINGS OF FIRE     B2        4
BK2        THE WINGS OF FIRE     B3        6
BK3        YOUNG INDIA           B3        2
BK4        10 SECONDS            B4        6
```

BK5	APTI MITHRA	B4	3
BK6	OLIVE BOARD	B4	3
BK7	PASSWORD	B5	2

1.6 Pre – Experiment Questions:

1. What is database?
2. What is DBMS?
3. What is a Database system?
4. Advantages of DBMS?

1.7 Post – Experiment Questions:

1. Define the "integrity rules"
2. What is extension and intension?
3. What is Data Independence?
4. What is Data Model?

EXPERIMENT No. 02: Order Database

2.1 Experiment Details:

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 salesman who had more than one customer.
3. List all the salesman 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.

2.2 Softwares Required:

Operating System: Windows 7

DBMS: Oracle 10g

2.3 Pre-Requisite:

Basics of DBMS, SQL Programming

2.4 Procedure:

TABLE CREATION:

```
CREATE TABLE SALESMAN (
SALESMAN_ID VARCHAR (20),
NAME CHAR (15),
CITY CHAR (15),
COMMISSION INT,
PRIMARY KEY (SALESMAN_ID));

CREATE TABLE CUSTOMER (
CUSTOMER_ID VARCHAR (20),
CUST_NAME CHAR (15),
CITY CHAR (15),
GRADE FLOAT,
SALESMAN_ID VARCHAR (20),
PRIMARY KEY (CUSTOMER_ID),
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID)
ON DELETE CASCADE);

CREATE TABLE ORDERS (
ORD_NO VARCHAR (15),
PURCHASE_AMT INT,
ORD_DATE DATE,
CUSTOMER_ID VARCHAR (20),
SALESMAN_ID VARCHAR (20),
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);
```

INSERTION OF VALUES:

```
SQL> INSERT INTO SALESMAN VALUES
('&SALESMAN_ID','&NAME','&CITY','&COMMISSION');
Enter value for salesman_id: 1000
Enter value for name: NIKIL
Enter value for city: MANGALORE
Enter value for commission: 1500
Old   1: INSERT INTO SALESMAN VALUES
('&SALESMAN_ID','&NAME','&CITY','&COMMISSION')
New   1: INSERT INTO SALESMAN VALUES ('1000','NIKIL','MANGALORE','1500')

1 row created.

SQL> SELECT * FROM SALESMAN;

SALESMAN_ID       NAME          CITY              COMMISSION
--------------------    ---------------      ---------------           ----------
1000            NIKIL         MANGALORE           1500
1001            CHETHAN       GOA                  800
1002             RAVI         BANGALORE            750
1003             KIRAN        MYSORE               850
1004            AKASH         MUMBAI               950
1005            VARUN         PUNE                 1200

SQL> INSERT INTO CUSTOMER VALUES
('&CUSTOMER_ID','&CUST_NAME','&CITY','&GRADE','&SALESMAN_ID');
Enter value for customer_id: C1
Enter value for cust_name: SHREYA
Enter value for city: GOA
Enter value for grade: 3.5
Enter value for salesman_id: 1001
```

```
Old   1: INSERT INTO CUSTOMER VALUES
('&CUSTOMER_ID','&CUST_NAME','&CITY','&GRADE','&SALESMAN_ID')
New   1: INSERT INTO CUSTOMER VALUES ('C1','SHREYA','GOA','3.5','1001')

1 row created.

SQL> SELECT * FROM CUSTOMER;

CUSTOMER_ID         CUST_NAME          CITY         GRADE   SALESMAN_ID
--------------------  ---------------  -------------- ----------  --------------------
C1                    SHREYA             GOA             3.5          1001
C2                    SHRUTHI            MYSORE          4.5          1003
C3                     PRATISTA           PUNE            3           1000
C4                    VIDYA              MUMBAI          2.5          1000
C5                    CRISEL             BANGALORE        2           1004
C6                    GLENISHA           BANGALORE       2.5          1005
C7                    SHAMANTH            BANGALORE      1.7          1002
C8                    JOHN               MANGALORE        5.3         1001

SQL> INSERT INTO ORDERS VALUES
('&ORD_NO','&PURCHASE_AMT','&ORD_DATE','&CUSTOMER_ID','&SALESMAN_ID');

SQL> SELECT * FROM ORDERS;

ORD_NO       PURCHASE_AMT         ORD_DATE    CUSTOMER_ID       SALESMAN_ID
--------------- ------------          ---------       --------------------      -----------------
OR1              25000               25-JAN-18         C1                   1001
OR2              13000               16-MAY-18         C2                   1003
OR3               3500               20-AUG-19         C3                   1000
OR4               5000               27-APR-18         C4                   1000
OR5               6000               27-APR-18         C5                   1004
OR6              10000               08-JUN-18         C6                   1005
```

```
OR7             15000            27-OCT-18        C7               1002
OR8            150000            06-DEC-18        C8               1001

8 rows selected.
```

2.5 RESULTS

QUERY 1:

```
SQL> SELECT COUNT (CUSTOMER_ID) FROM CUSTOMER
WHERE GRADE> (SELECT AVG (GRADE) FROM CUSTOMER
WHERE CITY='BANGALORE');

COUNT (CUSTOMER_ID)
------------------
           6
```

QUERY 2:

```
SQL> SELECT SALESMAN_ID, NAME FROM SALESMAN
WHERE SALESMAN_ID IN
(SELECT SALESMAN_ID FROM CUSTOMER
GROUP BY SALESMAN_ID HAVING COUNT (SALESMAN_ID) >1);

SALESMAN_ID         NAME
--------------------         ---------------
     1000                    NIKIL
     1001                    CHETHAN
```

QUERY 3:

```
SQL> (SELECT DISTINCT A.SALESMAN_ID, A.NAME, A.CITY
FROM SALESMAN A, CUSTOMER B
WHERE A.SALESMAN_ID=B.SALESMAN_ID AND A.CITY=B.CITY)
UNION
(SELECT DISTINCT A.SALESMAN_ID, A.NAME, A.CITY
FROM SALESMAN A, CUSTOMER B
WHERE A.SALESMAN_ID=B.SALESMAN_ID AND A.CITY! =B.CITY);

SALESMAN_ID          NAME                CITY
--------------------  ---------------     ---------------
     1000            NIKIL               MANGALORE
     1001            CHETHAN             GOA
     1002            RAVI                BANGALORE
     1003            KIRAN               MYSORE
     1004            AKASH               MUMBAI
     1005            VARUN               PUNE

6 rows selected.
```

QUERY 4:

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

View created.
```

```
SQL> SELECT * FROM MAX_ORDER;

SALESMAN_ID      NAME        ORD_DATE
--------------------   ---------------      ---------
1001              CHETHAN      25-JAN-18
1003               KIRAN       16-MAY-18
1000              NIKIL        20-AUG-19
1004              AKASH        27-APR-18
1005              VARUN        08-JUN-18
1002               RAVI        27-OCT-18
1001              CHETHAN      06-DEC-18

7 rows selected.
```

QUERY 5:

```
SQL> SELECT * FROM SALESMAN;

SALESMAN_ID      NAME          CITY              COMMISSION
--------------------    ---------------      ---------------         ----------
1000              NIKIL        MANGALORE          1500
1001              CHETHAN      GOA                 800
1002               RAVI        BANGALORE           750
1003               KIRAN       MYSORE              850
1004              AKASH        MUMBAI              950
1005              VARUN        PUNE                1200
```

```
SQL> SELECT * FROM CUSTOMER;

CUSTOMER_ID        CUST_NAME           CITY         GRADE  SALESMAN_ID
--------------       ----------         --------- -------   ------------
C1                   SHREYA             GOA            3.5        1001
C2                   SHRUTHI            MYSORE         4.5        1003
C3                    PRATISTA           PUNE           3         1000
C4                   VIDYA              MUMBAI         2.5        1000
C5                   CRISEL             BANGALORE       2         1004
C6                   GLENISHA           BANGALORE      2.5        1005
C7                   SHAMANTH            BANGALORE     1.7        1002
C8                   JOHN               MANGALORE      5.3        1001

SQL> SELECT * FROM ORDERS;

ORD_NO       PURCHASE_AMT         ORD_DATE     CUSTOMER_ID       SALESMAN_ID
---------- --------                ---------    -------------      -----------
OR1              25000             25-JAN-18          C1                 1001
OR2              13000             16-MAY-18          C2                 1003
OR3               3500             20-AUG-19          C3                 1000
OR4               5000             27-APR-18          C4                 1000
OR5               6000             27-APR-18          C5                 1004
OR6              10000             08-JUN-18          C6                 1005
OR7              15000             27-OCT-18          C7                 1002
OR8             150000             06-DEC-18          C8                 1001

SQL> DELETE FROM SALESMAN WHERE SALESMAN_ID=1000;

1 row deleted.
```

```
SQL> SELECT * FROM SALESMAN;

SALESMAN_ID        NAME                  CITY            COMMISSION
--------------------       ---------------           ---------------          ----------
1001                 CHETHAN            GOA                 800
1002                 RAVI               BANGALORE           750
1003                 KIRAN              MYSORE              850
1004                 AKASH              MUMBAI              950
1005                 VARUN              PUNE                1200

SQL> SELECT* FROM CUSTOMER;

CUSTOMER_ID     CUST_NAME     CITY             GRADE     SALESMAN_ID
--------------------      ----------------        ----------------          ----------       ------------------
C1               SHREYA        GOA                 3.5          1001
C2                SHRUTHI      MYSORE              4.5          1003
C5               CRISEL        BANGALORE           2           1004
C6               GLENISHA      BANGALORE           2.5          1005
C7               SHAMANTH      BANGALORE           1.7          1002
C8               JOHN          MANGALORE            5.3         1001

SQL> SELECT* FROM ORDERS;

ORD_NO     PURCHASE_AMT    ORD_DATE CUSTOMER_ID       SALESMAN_ID
--------------- ------------ --------- -------------------- --------------------
OR1            25000          25-JAN-18       C1                1001
OR2            13000         16-MAY-18        C2               1003
OR5             6000          27-APR-18       C5               1004
OR6            10000          08-JUN-18       C6               1005
OR7            15000          27-OCT-18       C7               1002
OR8           150000          06-DEC-18       C8               1001
```

2.6 Pre – Experiment Questions:

1. What is E-R model?
2. What is Object Oriented model?
3. What is an Entity?
4. What is an Entity type?
5. What is an Entity set?

2.7 Post – Experiment Questions:

1. What is an Extension of entity type?
2. What is an attribute?
3. What is a Relation Schema and a Relation?
4. What is degree of a Relation?
5. What is Relationship?

EXPERIMENT No. 03: Movie Database

3.1 Experiment Details:

Consider the schema for **Movie Database**:

ACTOR (Act_id, Act_Name, Act_Gender)

DIRECTOR (Dir_id, Dir_Name, Dir_Phone)

MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST (Act_id, Mov_id, Role)

RATING (Mov_id, Rev_Stars)

Write SQL queries to

1) List the titles of all movies directed by ‘Hitchcock’.
2) Find the movie names where one or more actors acted in two or more movies.
3) List all actors who acted in a movie before 2000 and also in a movie after 2015 (Use JOIN Operation).
4) Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
5) Update rating of all movies directed by ‘Steven Spielberg’ to 5.

3.2 Softwares Required:

Operating System: Windows 7

DBMS: Oracle 10g

3.3 Pre-Requisite:

Basics of DBMS, SQL Programming

3.4 Procedure:

TABLE CREATION:

```
CREATE TABLE ACTOR (
ACT_ID VARCHAR (10),
ACT_NAME CHAR (20),
ACT_GENDER CHAR (6),
PRIMARY KEY (ACT_ID));

CREATE TABLE DIRECTOR (
DIR_ID VARCHAR (10),
DIR_NAME CHAR (20),
DIR_PHONE VARCHAR (10),
PRIMARY KEY (DIR_ID));

CREATE TABLE MOVIES (
MOV_ID VARCHAR (10),
MOV_TITLE VARCHAR (20),
MOV_YEAR INT,
MOV_LANG CHAR (10),
DIR_ID VARCHAR (10),
PRIMARY KEY (MOV_ID),
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID)
ON DELETE CASCADE);

CREATE TABLE MOVIE_CAST (
ACT_ID VARCHAR (10),
MOV_ID VARCHAR (10),
ROLE CHAR (10),
PRIMARY KEY (ACT_ID, MOV_ID),
FOREIGN KEY (ACT_ID) REFERENCES ACTOR (ACT_ID)
ON DELETE CASCADE,
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID)
ON DELETE CASCADE);

 CREATE TABLE RATING (
 MOV_ID VARCHAR (10),
 REV_STARS INT,
 MOV_REVIEWER VARCHAR (20),
 PRIMARY KEY (MOV_ID, REV_STARS, MOV_REVIEWER),
 FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID)
 ON DELETE CASCADE);

SQL> DESC ACTOR;
 Name                          Null?              Type
 -----------------------------  ------------------  -------------
 ACT_ID                         NOT NULL          VARCHAR2 (10)
 ACT_NAME                                         CHAR (20)
 ACT_GENDER                                       CHAR (6)

SQL> DESC DIRECTOR;
 Name                      Null?           Type
 ------------------------------  ----------------  ---------------------
 DIR_ID                     NOT NULL       VARCHAR2 (10)
 DIR_NAME                                  CHAR (20)
 DIR_PHONE                                 VARCHAR2 (10)

SQL> DESC MOVIES;
 Name                          Null?       Type
 ----------------------------  ------------------  ------------------
 MOV_ID                      NOT NULL      VARCHAR2 (10)
 MOV_TITLE                                 VARCHAR2 (20)
 MOV_YEAR                                   NUMBER (38)
 MOV_LANG                                  CHAR (10)
 DIR_ID                                    VARCHAR2 (10)
```

```
SQL> DESC MOVIE_CAST;
 Name                           Null?           Type
 --------------------------------  --------------     ----------------------------
 ACT_ID                        NOT NULL        VARCHAR2 (10)
 MOV_ID                        NOT NULL        VARCHAR2 (10)
 ROLE                                          CHAR (10)

SQL> DESC RATING;
 Name                           Null?           Type
 ----------------------------       -----------------    -------------------
 MOV_ID                        NOT NULL        VARCHAR2 (10)
 REV_STARS                     NOT NULL        NUMBER (38)
 MOV_REVIEWER                  NOT NULL        VARCHAR2 (20)



SQL> COMMIT;
Commit complete.
```

INSERTION OF VALUES:

```
SQL> INSERT INTO ACTOR VALUES
('&ACT_ID','&ACT_NAME','&ACT_GENDER');
Enter value for act_id: A01
Enter value for act_name: ELLE
Enter value for act_gender: FEMALE
Old   1: INSERT INTO ACTOR VALUES
('&ACT_ID','&ACT_NAME','&ACT_GENDER')
New   1: INSERT INTO ACTOR VALUES ('A01',' ELLE ',' FEMALE ')
```

```
SQL> SELECT * FROM ACTOR;

ACT_ID          ACT_NAME            ACT_GENDER
----------      --------------------  ------------------
A01             ELLE                    FEMALE
A02             TOM                     MALE
A03             JENNIFER                FEMALE
A04             AUSTIN                  MALE
A05             SANDRA                  FEMALE
A06             GRAYSON                 MALE

SQL> INSERT INTO DIRECTOR VALUES
('&DIR_ID','&DIR_NAME','&DIR_PHONE');
Enter value for dir_id: D01
Enter value for dir_name: HITCHCOCK
Enter value for dir_phone: 9987654321
Old   1: INSERT INTO DIRECTOR VALUES
('&DIR_ID','&DIR_NAME','&DIR_PHONE')
New   1: INSERT INTO DIRECTOR VALUES ('D01','HITCHCOCK','9987654321')

SQL> SELECT * FROM DIRECTOR;

DIR_ID    DIR_NAME              DIR_PHONE
---------- --------------------           ----------
D01       HITCHCOCK             9987654321
D02       STEVEN SPIELBERG      9878685848
D03       LANDON                9998887654
D04       BEATRICE              9787675747
D05       CARL PHILIP           9989796959
D06       ADAM BUSBY            9181716151
```

```
SQL> INSERT INTO MOVIES VALUES
('&MOV_ID','&MOV_TITLE','&MOV_YEAR','&MOV_LANG','&DIR_ID');
Enter value for mov_id: M01
Enter value for mov_title: PSYCO
Enter value for mov_year: 1999
Enter value for mov_lang: ENGLISH
Enter value for dir_id: D01
Old   1: INSERT INTO MOVIES VALUES
('&MOV_ID','&MOV_TITLE','&MOV_YEAR','&MOV_LANG','&DIR_ID')
New   1: INSERT INTO MOVIES VALUES ('M01','PSYCO','1999','ENGLISH','D01')

SQL> SELECT * FROM MOVIES;

MOV_ID     MOV_TITLE            MOV_YEAR   MOV_LANG        DIR_ID
---------- --------------------  ----------  ----------      ----------
M01        PSYCO                 1999       ENGLISH         D01
M02        PACIFIER              1988       ENGLISH         D02
M03        JOKER                 2016       ENGLISH         D01
M04        SILENT NIGHT          1990       ENGLISH         D02
M05        ACE                   2017       ENGLISH         D03
M06        WINDSOR CASTLE        1997       ENGLISH         D04
M07        SWEDISH ROYALS        2016       SWEDISH         D05
M08        OUTDAUGHTERED         2017       ENGLISH         D06


SQL> INSERT INTO MOVIE_CAST VALUES ('&ACT_ID','&MOV_ID','&ROLE');
Enter value for act_id: A01
Enter value for mov_id: M05
Enter value for role: DAUGHTER
Old   1: INSERT INTO MOVIE_CAST VALUES ('&ACT_ID','&MOV_ID','&ROLE')
New   1: INSERT INTO MOVIE_CAST VALUES ('A01','M05','DAUGHTER')
```

```
SQL> SELECT * FROM MOVIE_CAST;

ACT_ID          MOV_ID     ROLE
----------      ----------  ----------
A01             M05         DAUGHTER
A03             M06         HEROINE
A03             M05         MOTHER
A01             M06         SISTER
A06             M04         HERO
A04             M05         FATHER

SQL> INSERT INTO RATING VALUES
('&MOV_ID','&REV_STARS','&MOV_REVIEWER');
Enter value for mov_id: M01
Enter value for rev_stars: 5
Enter value for mov_reviewer: INDIA TIMES
Old   1:  INSERT INTO RATING VALUES
('&MOV_ID','&REV_STARS','&MOV_REVIEWER')
New   1:  INSERT INTO RATING VALUES ('M01','5','INDIA TIMES')

SQL> SELECT * FROM RATING;

MOV_ID     REV_STARS   MOV_REVIEWER
----------    ----------     --------------------
M01          5            INDIA TIMES
M02          2            THE HINDU
M03          3            THE HINDU
M04          1            INDIA TIMES
M05          4            YOUTUBERS
M08          4            TLC
```

3.5 RESULTS:

QUERY 1:

```
SQL> SELECT MOV_TITLE FROM MOVIES, DIRECTOR WHERE
MOVIES.DIR_ID=DIRECTOR.DIR_ID AND DIR_NAME='HITCHCOCK';

MOV_TITLE
--------------------
PSYCO
JOKER
```

QUERY 2:

```
SQL> SELECT DISTINCT MOV_TITLE FROM MOVIES, MOVIE_CAST WHERE
MOVIES.MOV_ID=MOVIE_CAST.MOV_ID AND MOVIE_CAST.ACT_ID
IN
(SELECT ACT_ID FROM MOVIE_CAST GROUP BY ACT_ID HAVING
COUNT (*)>=2);

MOV_TITLE
--------------------
ACE
WINDSOR CASTLE
```

QUERY 3:

(WITHOUT JOIN OOPERATION)

```
SQL> (SELECT DISTINCT ACT_NAME FROM ACTOR, MOVIES, MOVIE_CAST
WHERE MOVIES.MOV_YEAR<2000 AND
ACTOR.ACT_ID= MOVIE_CAST.ACT_ID AND
MOVIES.MOV_ID=MOVIE_CAST.MOV_ID)
INTERSECT
(SELECT DISTINCT ACT_NAME FROM ACTOR, MOVIES, MOVIE_CAST
WHERE MOVIES.MOV_YEAR>2015 AND
ACTOR.ACT_ID=MOVIE_CAST.ACT_ID AND
MOVIES.MOV_ID=MOVIE_CAST.MOV_ID);

ACT_NAME
--------------------
ELLE
JENNIFER

SQL> SELECT DISTINCT ACT_NAME FROM ACTOR, MOVIES, MOVIE_CAST
WHERE MOVIES.MOV_YEAR<2000 AND
ACTOR.ACT_ID=MOVIE_CAST.ACT_ID AND
MOVIES.MOV_ID=MOVIE_CAST.MOV_ID;

ACT_NAME
--------------------
GRAYSON
JENNIFER
ELLE
```

```
SQL> SELECT DISTINCT ACT_NAME FROM ACTOR, MOVIES, MOVIE_CAST
WHERE MOVIES.MOV_YEAR>2015 AND
ACTOR.ACT_ID=MOVIE_CAST.ACT_ID AND
MOVIES.MOV_ID=MOVIE_CAST.MOV_ID;

ACT_NAME
--------------------
ELLE
JENNIFER
AUSTIN
```

(WITH JOIN OPERATION)

```
SQL> (SELECT DISTINCT A.ACT_NAME FROM ACTOR A
  JOIN MOVIE_CAST B ON A.ACT_ID=B.ACT_ID
  JOIN MOVIES C ON B.MOV_ID=C.MOV_ID WHERE C.MOV_YEAR<2000)
  INTERSECT
  (SELECT DISTINCT A.ACT_NAME FROM ACTOR A
  JOIN MOVIE_CAST B ON A.ACT_ID=B.ACT_ID
  JOIN MOVIES C ON B.MOV_ID=C.MOV_ID WHERE C.MOV_YEAR>2015);

ACT_NAME
--------------------
ELLE
JENNIFER
```

QUERY 4:

```
SQL> SELECT A.MOV_TITLE, MAX (B.REV_STARS) FROM
MOVIES A, RATING B WHERE A.MOV_ID=B.MOV_ID
GROUP BY A.MOV_TITLE ORDER BY A.MOV_TITLE;

MOV_TITLE           MAX (B.REV_STARS)
-------------------- ----------------
ACE                        4
JOKER                      3
OUTDAUGHTERED              4
PACIFIER                   2
PSYCO                      5
SILENT NIGHT               1

SQL> SELECT A.MOV_TITLE, MAX (B.REV_STARS) FROM
MOVIES A, RATING B WHERE A.MOV_ID=B.MOV_ID
GROUP BY A.MOV_TITLE;

MOV_TITLE           MAX (B.REV_STARS)
--------------------        ----------------
PSYCO                           5
JOKER                           3
ACE                             4
OUTDAUGHTERED                   4
PACIFIER                        2
SILENT NIGHT                    1
```

QUERY 5:

```
SQL> SELECT * FROM DIRECTOR;

DIR_ID    DIR_NAME                DIR_PHONE
---------- --------------------    ----------
D01       HITCHCOCK               9987654321
D02       STEVEN SPIELBERG        9878685848
D03       LANDON                  9998887654
D04       BEATRICE                9787675747
D05       CARL PHILIP             9989796959
D06       ADAM BUSBY              9181716151

SQL> SELECT * FROM MOVIES;

MOV_ID    MOV_TITLE           MOV_YEAR   MOV_LANG          DIR_ID
---------- --------------------   ----------  ----------        ----------
M01       PSYCO                1999       ENGLISH           D01
M02       PACIFIER             1988       ENGLISH           D02
M03       JOKER                2016       ENGLISH           D01
M04       SILENT NIGHT         1990       ENGLISH           D02
M05       ACE                  2017       ENGLISH           D03
M06       WINDSOR CASTLE       1997       ENGLISH           D04
M07       SWEDISH ROYALS       2016       SWEDISH           D05
M08       OUTDAUGHTERED        2017       ENGLISH           D06

SQL> SELECT * FROM RATING;

MOV_ID     REV_STARS   MOV_REVIEWER
----------   ----------    --------------------
M01            5          INDIA TIMES
M02            2          THE HINDU
M03            3          THE HINDU
```

```
M04           1           INDIA TIMES
M05           4           YOUTUBERS
M08           4           TLC

SQL> UPDATE RATING SET REV_STARS=5 WHERE MOV_ID IN
(SELECT MOV_ID FROM MOVIES A, DIRECTOR B
WHERE A.DIR_ID=B.DIR_ID AND B.DIR_NAME='STEVEN SPIELBERG');

2 rows updated.

SQL> SELECT * FROM RATING;

MOV_ID     REV_STARS   MOV_REVIEWER
----------   ----------      --------------------
M01           5           INDIA TIMES
M02           5           THE HINDU
M03           3           THE HINDU
M04           5           INDIA TIMES
M05           4           YOUTUBERS
M08           4           TLC
```

3.6 Pre – Experiment Questions:

1. What is Relationship set?
2. What is Relationship type?
3. What is degree of Relationship type?
4. What is DDL (Data Definition Language)?

3.7 Post – Experiment Questions:

1. What is Relational Algebra?
2. What is Relational Calculus?
3. What is normalization?
4. What is Functional Dependency?

EXPERIMENT No. 04: College Database

4.1 Experiment Details:

Consider the schema for **College Database:**

STUDENT (USN, SName, Address, Phone, Gender)
SEMSEC (SSID, Sem, Sec)
CLASS (USN, SSID)
SUBJECT (Subcode, Title, Sem, Credits)
IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to

1. List all the student details studying in fourth semester 'C' section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN '1BI17CS101' in all subjects.
4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:

   If FinalIA = 17 to 20 then CAT = 'Outstanding'
   If FinalIA = 12 to 16 then CAT = 'Average'
   If FinalIA< 12 then CAT = 'Weak'
   Give these details only for 8th semester A, B, and C section students.

4.2 Softwares Required:

Operating System: Windows 7
DBMS: Oracle 10g

4.3 Pre-Requisite:

Basics of DBMS, SQL Programming

4.4 Procedure:

TABLE CREATION:

```
 CREATE TABLE STUDENT (
 USN VARCHAR (20),
 SNAME CHAR (15),
 ADDRESS VARCHAR (20),
 PHONE NUMBER (10),
 GENDER CHAR (6),
 PRIMARY KEY (USN));

CREATE TABLE SEMSEC (
 SSID VARCHAR (20),
 SEM NUMBER (3),
 SEC CHAR (3),
 PRIMARY KEY (SSID));

CREATE TABLE CLASS (
USN VARCHAR (20),
SSID VARCHAR (20),
PRIMARY KEY (USN, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN)
ON DELETE CASCADE,
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID)
ON DELETE CASCADE);

CREATE TABLE SUBJECT (
SUBCODE VARCHAR (15),
TITLE CHAR (20),
SEM NUMBER (3),
CREDITS INT,
PRIMARY KEY (SUBCODE));
```

```
CREATE TABLE IAMARKS (
USN VARCHAR (20),
SUBCODE VARCHAR (15),
SSID VARCHAR (20),
TEST1 NUMBER (2),
TEST2 NUMBER (2),
TEST3 NUMBER (2),
FINALIA NUMBER (2),
PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN)
ON DELETE CASCADE,
FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE)
ON DELETE CASCADE,
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID)
ON DELETE CASCADE);
```

INSERTION OF VALUES:

```
SQL> INSERT INTO STUDENT VALUES
('&USN','&SNAME','&ADDRESS','&PHONE','&GENDER');
Enter value for usn: 4SF17IS089
Enter value for sname: SHRITHI
Enter value for address: MERAMAJALU
Enter value for phone: 7338527841
Enter value for gender: FEMALE
Old   1: INSERT INTO STUDENTS VALUES
('&USN','&SNAME','&ADDRESS','&PHONE','&GENDER')
New   1: INSERT INTO STUDENTS VALUES ('4SF17IS089',' SHRITHI ', '
MERAMAJALU','7338527841','FEMALE')
```

```
SQL> SELECT * FROM STUDENT;

USN            SNAME          ADDRESS              PHONE        GENDER
-------------- -------------- -------------------- ----------   ------
4SF17IS089     SHRITHI        MERAMAJALU           7338527841   FEMALE
4SF17IS109     VARSHINI       BANTWAL              8999876543   FEMALE
4SF17IS001     SACHIN         VITTAL               7895858580   MALE
4SF17IS098     NAVISH         FARANGIPETE          9876543210   MALE
4SF17IS099     SWATHI         MANCHI               6878798765   FEMALE
4SF17IS108     VARSHA         KADRI                7896543210   FEMALE
4SF17IS088     BHARATH        PADIL                8765432100   MALE
1BI17CS101     SAHANA         MANGALORE            8765432190   FEMALE
4SF17CS088     KAVYA          ALAPE                9887665432   FEMALE
4SF17CS077     AYUSH          KALLADKA             9008765467   MALE

SQL> INSERT INTO SEMSEC VALUES ('&SSID','&SEM','&SEC');
Enter value for ssid: S1
Enter value for sem: 4
Enter value for sec: A
Old   1: INSERT INTO SEMSEC VALUES ('&SSID','&SEM','&SEC')
New   1: INSERT INTO SEMSEC VALUES ('S1','4','A')

SQL> SELECT * FROM SEMSEC;

SSID    SEM     SEC
------- ------  ---
S1      4       A
S2      4       C
S3      8       A
S4      8       B
S5      8       C
S6      5       A
```

```
SQL> INSERT INTO CLASS VALUES ('&USN','&SSID');
Enter value for usn: 4SF17IS089
Enter value for ssid: S6
Old   1: INSERT INTO CLASS VALUES ('&USN','&SSID')
New   1: INSERT INTO CLASS VALUES ('4SF17IS089','S6')

SQL> SELECT * FROM CLASS;

USN                SSID
-----------------  -------
1BI17CS101         S1
4SF17CS077         S2
4SF17CS088         S2
4SF17IS001         S3
4SF17IS088         S5
4SF17IS089         S6
4SF17IS098         S4
4SF17IS099         S4
4SF17IS108         S5
4SF17IS109         S3

SQL> INSERT INTO SUBJECT VALUES
('&SUBCODE','&TITLE','&SEM','&CREDITS');
Enter value for subcode: 17CS31
Enter value for title: DATA STRUCTURE
Enter value for sem: 4
Enter value for credits: 4
Old   1: INSERT INTO SUBJECT VALUES
('&SUBCODE','&TITLE','&SEM','&CREDITS')
New   1: INSERT INTO SUBJECT VALUES ('17CS31','DATA STRUCTURE','4','4')
```

```
SQL> SELECT * FROM SUBJECT;

SUBCODE       TITLE                  SEM          CREDITS
-------------- -------------------               ----------      ----------
17CS31        DATA STRUCTURE          4            4
17CS32        UNIX                    4            3
17CS33        DBMS                    5            4
17CS34        DOTNET                  5            3
17CS35        FLAT                    8            4
17CS36        JAVA                    8            3
17CS37        FS                      8            4

SQL> INSERT INTO IAMARKS VALUES
('&USN','&SUBCODE','&SSID','&TEST1','&TEST2','&TEST3','&FINALIA');
Enter value for usn: 1BI17CS101
Enter value for subcode: 17CS31
Enter value for ssid: S1
Enter value for test1: 12
Enter value for test2: 16
Enter value for test3: 13
Enter value for finalia:
Old   1: INSERT INTO IAMARKS VALUES
('&USN','&SUBCODE','&SSID','&TEST1','&TEST2','&TEST3','&FINALIA')
New   1: INSERT INTO IAMARKS VALUES ('1BI17CS101','17CS31','S1','12','16','13','')
```

```
SQL> SELECT * FROM IAMARKS;

USN              SUBCODE    SSID      TEST1      TEST2      TEST3     FINALIA
-----------------  -------------  ---------  ----------  ----------  ----------  ---------
1BI17CS101       17CS31      S1         12         16         13
1BI17CS101       17CS32      S1          8         19         16
4SF17CS077       17CS31      S2          7         12          8
4SF17CS077       17CS32      S2         16         18         19
4SF17CS088       17CS31      S2          9         12         10
4SF17CS088       17CS32      S2         18         16         17
4SF17IS001       17CS35      S3         12         15          9
4SF17IS001       17CS36      S3         16         16         17
4SF17IS001       17CS37      S3         19         18         17
4SF17IS088       17CS35      S5         19         20         17
4SF17IS088       17CS36      S5         13         14         15
4SF17IS088       17CS37      S5         15         13         17
4SF17IS089       17CS33      S6         19         20         20
4SF17IS089       17CS34      S6         19         20         19
4SF17IS098       17CS35      S4         13         15         16
4SF17IS098       17CS36      S4         13         15         16
4SF17IS098       17CS37      S4         17         15         16
4SF17IS099       17CS35      S4         19         20         20
4SF17IS099       17CS36      S4         19         20         19
4SF17IS099       17CS37      S4         19         20         20
4SF17IS108       17CS35      S5         16         16         16
4SF17IS108       17CS36      S5         18         17         19
4SF17IS108       17CS37      S5         19         16         20
4SF17IS109       17CS35      S3         18         18         18
4SF17IS109       17CS36      S3         19         16         19
4SF17IS109       17CS37      S3         16         17         18
```

4.5 RESULTS

QUERY 1:

```
SQL> SELECT A.*, B.SEM, B.SEC
     FROM STUDENTS A, SEMSEC B, CLASS C
     WHERE A.USN=C.USN AND B.SSID=C.SSID AND B.SEM=4 AND B.SEC='C';

USN            SNAME          ADDRESS        PHONE          GENDER   SEM SEC
----------------  ------------  ------------------  ---------------  ----------  ---
4SF17CS088     KAVYA          ALAPE          9887665432     FEMALE     4   C
4SF17CS077     AYUSH          KALLADKA       9008765467     MALE       4   C
```

QUERY 2:

```
SQL> SELECT B.SEM, B.SEC,
COUNT (CASE WHEN A.GENDER='MALE' THEN B.SSID END) AS MALECOUNT,
COUNT (CASE WHEN A.GENDER='FEMALE' THEN B.SSID END) AS
FEMALECOUNT
FROM STUDENT A, SEMSEC B, CLASS C
WHERE A.USN=C.USN AND B.SSID=C.SSID
GROUP BY B.SEM, B.SEC;

    SEM          SEC    MALECOUNT    FEMALECOUNT
    --------     ---    ----------       -----------
     4           A        0              1
     8           C        1              1
     8           B        1              1
     8           A        1              1
     5           A        0              1
     4           C        1              1
```

QUERY 3:

```
SQL> CREATE VIEW TEST1_MARKS AS
SELECT SUBCODE, TEST1 FROM IAMARKS
WHERE USN='1BI17CS101';

View created.

SQL> SELECT * FROM TEST1_MARKS;

SUBCODE          TEST1
---------------   --------
17CS31            12
17CS32             8
```

QUERY 4:

```
SQL> UPDATE IAMARKS SET FINALIA=GREATEST (
(TEST1+TEST2)/2,
(TEST2+TEST3)/2,
(TEST3+TEST1)/2);

26 rows updated.

SQL> SELECT * FROM IAMARKS;
```

```
USN             SUBCODE         SSID        TEST1   TEST2   TEST3   FINALIA
--------------   ------         ----------  ---------- ---------- --------   ---------
1BI17CS101      17CS31          S1           12      16      13      15
1BI17CS101      17CS32          S1           8       19      16      18
4SF17CS077      17CS31          S2           7       12      8       10
4SF17CS077      17CS32          S2           16      18      19      19
4SF17CS088      17CS31          S2           9       12      10      11
4SF17CS088      17CS32          S2           18      16      17      18
4SF17IS001      17CS35          S3           12      15      9       14
4SF17IS001      17CS36          S3           16      16      17      17
4SF17IS001      17CS37          S3           19      18      17      19
4SF17IS088      17CS35          S5           19      20      17      20
4SF17IS088      17CS36          S5           13      14      15      15
4SF17IS088      17CS37          S5           15      13      17      16
4SF17IS089      17CS33          S6           19      20      20      20
4SF17IS089      17CS34          S6           19      20      19      20
4SF17IS098      17CS35          S4           13      15      16      16
4SF17IS098      17CS36          S4           13      15      16      16
4SF17IS098      17CS37          S4           17      15      16      17
4SF17IS099      17CS35          S4           19      20      20      20
4SF17IS099      17CS36          S4           19      20      19      20
4SF17IS099      17CS37          S4           19      20      20      20
4SF17IS108      17CS35          S5           16      16      16      16
4SF17IS108      17CS36          S5           18      17      19      19
4SF17IS108      17CS37          S5           19      16      20      20
4SF17IS109      17CS35          S3           18      18      18      18
4SF17IS109      17CS36          S3           19      16      19      19
4SF17IS109      17CS37          S3           16      17      18      18
```

QUERY 5:

```
SQL> SELECT S.*, IA.FINALIA,
(CASE
WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING'
WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
ELSE 'WEAK' END) AS CAT FROM STUDENT S, SEMSEC SS, IAMARKS IA
WHERE S.USN=IA.USN AND SS.SSID=IA.SSID AND SS.SEM=8;

USN            SNAME        ADDRESS         PHONE        GENDER    FINALIA CAT
---------------  -------------   -------------------  -------------     --------------- ----------  --------
4SF17IS109   VARSHINI  BANTWAL       8999876543   FEMALE    18     OUTSTANDING
4SF17IS109   VARSHINI  BANTWAL       8999876543   FEMALE    19     OUTSTANDING
4SF17IS109   VARSHINI  BANTWAL       8999876543   FEMALE    18     OUTSTANDING
4SF17IS001   SACHIN    VITTAL        7895858580   MALE      19     OUTSTANDING
4SF17IS001   SACHIN    VITTAL        7895858580   MALE      17     OUTSTANDING
4SF17IS001   SACHIN    VITTAL        7895858580   MALE      14     AVERAGE
4SF17IS098   NAVISH    FARANGIPETE   9876543210   MALE      17     OUTSTANDING
4SF17IS098   NAVISH    FARANGIPETE   9876543210   MALE      16     AVERAGE
4SF17IS098   NAVISH    FARANGIPETE   9876543210   MALE      16     AVERAGE
4SF17IS099   SWATHI    MANCHI        6878798765   FEMALE    20     OUTSTANDING
4SF17IS099   SWATHI    MANCHI        6878798765   FEMALE    20     OUTSTANDING
4SF17IS099   SWATHI    MANCHI        6878798765   FEMALE    20     OUTSTANDING
4SF17IS108   VARSHA    KADRI         7896543210   FEMALE    20     OUTSTANDING
4SF17IS108   VARSHA    KADRI         7896543210   FEMALE    19     OUTSTANDING
4SF17IS108   VARSHA    KADRI         7896543210   FEMALE    16     AVERAGE
4SF17IS088   BHARATH   PADIL         8765432100   MALE      16     AVERAGE
4SF17IS088   BHARATH   PADIL         8765432100   MALE      15     AVERAGE
4SF17IS088   BHARATH   PADIL         8765432100   MALE      20     OUTSTANDING
```

4.6 Pre – Experiment Questions:

1. When is a functional dependency F said to be minimal?
2. What is multivalued dependency?
3. What is 1 NF (Normal Form), 2NF, 3NF?
4. What is Lossless join property?
5. What is Fully Functional dependency?

4.7 Post – Experiment Questions:

1. What is BCNF (Boyce-Codd Normal Form)?
2. What is 4NF?
3. What is 5NF?
4. What is Domain-Key Normal Form?
5. What are partial, alternate, artificial, compound and natural key?

EXPERIMENT No. 05: Company Database

5.1 Experiment Details:

Consider the schema for **Company Database**:

EMPLOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, DNo)
DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate)
DLOCATION (DNo,DLoc)
PROJECT (PNo, PName, PLocation, DNo)
WORKS_ON (SSN, PNo, Hours)

Write SQL queries to

1. Make a list of all project numbers for projects that involve an employee whose name is ‘Scott’, either as a worker or as a manager of the department that controls the project.
2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.
3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department
4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).
5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6, 00,000.

5.2 Softwares Required:

Operating System: Windows 7

DBMS: Oracle 10g

5.3 Pre-Requisite:

Basics of DBMS, SQL Programming

5.4 Procedure:

TABLE CREATION:

```
CREATE TABLE DEPARTMENT (
DNO INT,
DNAME VARCHAR (30),
MGR_SSN VARCHAR (10),
MGR_START_DATE DATE,
PRIMARY KEY (DNO));

CREATE TABLE EMPLOYEE (
SSN VARCHAR (10),
NAME VARCHAR (20),
ADDRESS VARCHAR (20),
SEX CHAR (1),
SALARY DECIMAL (10, 3),
SUPER_SSN VARCHAR (10),
DNO INT,
PRIMARY KEY (SSN),
FOREIGN KEY (SUPER_SSN) REFERENCES EMPLOYEE (SSN)
ON DELETE CASCADE,
FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNO)
ON DELETE CASCADE);

CREATE TABLE DLOCATION (
DNO INT,
DLOC VARCHAR (20),
PRIMARY KEY (DNO, DLOC),
FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNO)
ON DELETE CASCADE);
```

```
CREATE TABLE PROJECT (
PNO INT,
PNAME VARCHAR (20),
PLOCATION VARCHAR (20),
DNO INT,
PRIMARY KEY (PNO),
FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNO)
ON DELETE CASCADE);

CREATE TABLE WORKS_ON (
SSN VARCHAR (20),
PNO INT,
HOURS INT,
PRIMARY KEY (SSN, PNO),
FOREIGN KEY (SSN) REFERENCES EMPLOYEE (SSN)
ON DELETE CASCADE,
FOREIGN KEY (PNO) REFERENCES PROJECT (PNO)
ON DELETE CASCADE);

SQL> ALTER TABLE DEPARTMENT
ADD CONSTRAINT C_MSSN
FOREIGN KEY (MGR_SSN) REFERENCES EMPLOYEE
ON DELETE CASCADE;

Table altered.
```

INSERTION OF VALUES:

```
SQL> INSERT INTO DEPARTMENT VALUES
('&DNO','&DNAME','&MGRSSN','&MGRSTARTDATE');
Enter value for dno: 1
Enter value for dname: ACCOUNTS
Enter value for mgrssn:
Enter value for mgrstartdate: 25-MAR-2018
Old   1: INSERT INTO DEPARTMENT VALUES
('&DNO','&DNAME','&MGRSSN','&MGRSTARTDATE')
New   1: INSERT INTO DEPARTMENT VALUES ('1','ACCOUNTS',",'25-MAR-2018')

SQL> SELECT * FROM DEPARTMENT;

    DNO          DNAME                  MGR_SSN    MGR_START
    -------      ---------------------- ---------- ---------
     1           ACCOUNTS                          25-MAR-18
     2           DEVELOPER                         25-JUL-17
     3           ESTABLISHMENT                     16-DEC-15
     4           FINANCE                           08-JAN-11
     5           MAMNAGEMENT                       11-NOV-09


SQL> INSERT INTO EMPLOYEE VALUES
('&SSN','&NAME','&ADDRESS','&SEX','&SALARY','&SUPERSSN','&DNO');
Enter value for ssn: E1
Enter value for name: SCOTT
Enter value for address: SAN FRANSISCO
Enter value for sex: M
Enter value for salary: 700000
Enter value for superssn:
Enter value for dno: 1
```

```
Old   1: INSERT INTO EMPLOYEE VALUES
('&SSN','&NAME','&ADDRESS','&SEX','&SALARY','&SUPERSSN','&DNO')
New   1: INSERT INTO EMPLOYEE VALUES
('E1','SCOTT','SAN FRANSISCO','M','700000','','1')

SQL> SELECT * FROM EMPLOYEE;

SSN    NAME             ADDRESS               SEX     SALARY   SUPER_SSN      DNO
------ ---------------- --------------------- ------- ---------- --------------- -------
E1     SCOTT            SAN FRANSISCO          M      700000                     1
E2     ALBERT           SAN ANDREAS            M      800000      E1             1
E3     JULIET           ENGLAND                F      900000      E2             1
E4     ROBERT           NEW YORK               M      850000      E1             1
E5     BAIRSTOW         NEW ZEALAND            M      750000      E4             1
E6     SMITH            AUSTRALIA              M      800000      E4             1
E7     WARNER           TURKEY                 M      500000      E3             2
E8     LUCY             AFRICA                 F      400000      E6             3
E9     ROSE             NETHERLANDS            F      300000      E8             4
E10    ALIS             NEIGERIA               M      750000      E2             5
E11    JAMES            CHINA                  M      900000      E8             5
E12    ANDERSON         KOREA                  M      500000      E9             4
E13    DISHA            MUMBAI                 F      400000      E12            3
E14    SARA             DELHI                  F      250000      E14            4
E15    JOYCE            CALIFORNIA             M      60000       E13            5


SQL> UPDATE EMPLOYEE SET SUPER_SSN='E15' WHERE SSN='E1';
SQL> SELECT * FROM EMPLOYEE;

SSN     NAME            ADDRESS          SEX    SALARY   SUPER_SSN    DNO
------  ----------------  ----------------------  -------  ----------  ---------------  -------
E1      SCOTT           SAN FRANSISCO    M     700000      E15         1
E2      ALBERT          SAN ANDREAS      M     800000      E1          1
E3      JULIET          ENGLAND          F     900000      E2          1
E4      ROBERT          NEW YORK         M     850000      E1          1
E5      BAIRSTOW        NEW ZEALAND      M     750000      E4          1
E6      SMITH           AUSTRALIA        M     800000      E4          1
E7      WARNER          TURKEY           M     500000      E3          2
E8      LUCY            AFRICA           F     400000      E6          3
E9      ROSE            NETHERLANDS      F     300000      E8          4
E10     ALIS            NEIGERIA         M     750000      E2          5
E11     JAMES           CHINA            M     900000      E8          5
E12     ANDERSON        KOREA            M     500000      E9          4
E13     DISHA           MUMBAI           F     400000      E12         3
E14     SARA            DELHI            F     250000      E14         4
E15     JOYCE           CALIFORNIA       M     60000       E13         5

SQL> UPDATE DEPARTMENT SET MGR_SSN='E4' WHERE DNO='1';

SQL> UPDATE DEPARTMENT SET MGR_SSN='E2' WHERE DNO='2';

SQL> UPDATE DEPARTMENT SET MGR_SSN='E1' WHERE DNO='3';

SQL> UPDATE DEPARTMENT SET MGR_SSN='E5' WHERE DNO='4';

SQL> UPDATE DEPARTMENT SET MGR_SSN='E3' WHERE DNO='5';
SQL> SELECT * FROM DEPARTMENT;

     DNO            DNAME                    MGR_SSN   MGR_START
     -------        ------------------------  ----------    ---------
      1             ACCOUNTS                  E4          25-MAR-18
      2             DEVELOPER                 E2          25-JUL-17
      3             ESTABLISHMENT             E1          16-DEC-15
      4             FINANCE                   E5          08-JAN-11
      5             MAMNAGEMENT               E3          11-NOV-09

SQL> INSERT INTO DLOCATION VALUES ('&DNO','&DLOC');
Enter value for dno: 1
Enter value for dloc: RUSSIA
Old   1: INSERT INTO DLOCATION VALUES ('&DNO','&DLOC')
New   1: INSERT INTO DLOCATION VALUES ('1','RUSSIA')

SQL> SELECT * FROM DLOCATION;

     DNO            DLOC
     ----------     -------------------
      1             RUSSIA
      2             CHINA
      3             LOS VEGAS
      4             MUMBAI
      5             CALIFORNIA

SQL> INSERT INTO PROJECT VALUES
('&PNO','&PNAME','&PLOACTION','&DNO');
Enter value for pno: 1
Enter value for pname: IOT
Enter value for ploaction: MUMBAI
Enter value for dno: 1
```

```
Old   1: INSERT INTO PROJECT VALUES
('&PNO','&PNAME','&PLOACTION','&DNO')
New   1: INSERT INTO PROJECT VALUES ('1','IOT','MUMBAI','1')

SQL> SELECT * FROM PROJECT;

     PNO           PNAME                          PLOCATION            DNO
     ------        ------------------------------ -----------------    -------
       1           IOT                            MUMBAI                  1
       2           BANK SOFTWARE                  BANGALORE               2
       3           EMBEDDED SYSTEMS               HONG KONG               3
       4           FOREST DEPARTMENT              DELHI                   4
       5           GOVERNMENT APP                 GANDHINAGAR             5

SQL> INSERT INTO WORKS_ON VALUES ('&SSN','&PNO','&HOURS');
Enter value for ssn: E1
Enter value for pno: 1
Enter value for hours: 8
Old   1: INSERT INTO WORKS_ON VALUES ('&SSN','&PNO','&HOURS')
New   1: INSERT INTO WORKS_ON VALUES ('E1','1','8')

SQL> SELECT * FROM WORKS_ON;

SSN     PNO     HOURS
------  ------- ----------
E1      1           8
E2      2           9
E3      3           6
E4      4           5
E5      5           10
```

5.5 RESULTS:

QUERY 1:

```
(SELECT DISTINCT P.PNO FROM PROJECT P, EMPLOYEE E, DEPARTMENT D
WHERE E.SSN=D.MGR_SSN AND D.DNO=P.DNO AND E.NAME='SCOTT')
UNION
(SELECT DISTINCT P.PNO FROM PROJECT P, EMPLOYEE E, WORKS_ON W
WHERE E.SSN=W.SSN AND P.PNO=W.PNO AND E.NAME='SCOTT');
```

```
   PNO
  -------
     1
     3
```

QUERY 2:

```
SELECT E.NAME, 1.1*SALARY AS INCREASED_SALARY FROM EMPLOYEE E,
PROJECT P, WORKS_ON W WHERE E.SSN=W.SSN AND P.PNO=W.PNO AND
P.PNAME='IOT';
```

```
NAME          INCREASED_SALARY
---------      ----------------
SCOTT          770000
```

QUERY 3:

```
SELECT SUM (SALARY), MAX (SALARY), MIN (SALARY), AVG (SALARY) FROM
EMPLOYEE E, DEPARTMENT D WHERE
E.DNO=D.DNO AND D.DNAME='ACCOUNTS';
```

```
SUM (SALARY)     MAX (SALARY)  MIN (SALARY)       AVG (SALARY)
-----------      -----------   -----------        -----------
4800000          900000        700000             800000
```

QUERY 4:

```
SELECT E.NAME FROM EMPLOYEE E WHERE    NOT EXISTS
((SELECT P.PNO FROM PROJECT P WHERE P.DNO=5)
MINUS
(SELECT W.PNO FROM WORKS_ON W WHERE E.SSN=W.SSN));
```

```
NAME
----------------
BAIRSTOW
```

QUERY 5:

```
SELECT E.DNO, COUNT (*) FROM EMPLOYEE E WHERE SALARY > 600000 AND
E.DNO IN
(SELECT DNO FROM EMPLOYEE GROUP BY DNO HAVING COUNT (*) > 5)
GROUP BY E.DNO;
```

```
     DNO   COUNT (*)
     -------  ----------
       1        6
```

5.6 Pre – Experiment Questions:

1. What is indexing and what are the different kinds of indexing?
2. What is system catalog or catalog relation? How is better known as?
3. What is meant by query optimization?
4. What is join dependency and inclusion dependency?

5.7 Post – Experiment Questions:

1. What is durability in DBMS?
2. What do you mean by atomicity and aggregation?
3. What is a checkpoint and when does it occur?
4. What are the different phases of transaction?
5. What is "transparent DBMS"?