

DON BOSCO INSTITUTE OF TECHNOLOGY



(Approved by AICTE-New Delhi, NBA, Accredited courses, affiliated to VTU, Belgaum)

BENGALURU-560074

Laboratory Certificate

This is to certify that Mr. / Ms. Lathaiah..... has

satisfactorily completed the course of experiments in practical

..... DataBase Management System..... Laboratory..... prescribed by the

..... DBIT..... VTU..... university

..... Computer Science & Engineering..... course in the

Laboratory of this college in the year 2021 - 2022

Signature of the Head of the Department

Date: 11/2/22

Signature of the Lecturer

In-charge of the batch

| Marks | |
|---------|----------|
| Maximum | Obtained |
| 40 | 38 |

Name of the Candidate: Lathaiah.....

USN: IDB2D.CS405.....

Examination Centre: DBIT.....

Date of Practical Examination:

INDEX

Name of the Student..... Pathanamitri..... Class: 5B.CSE..... Year: 2021-2022,

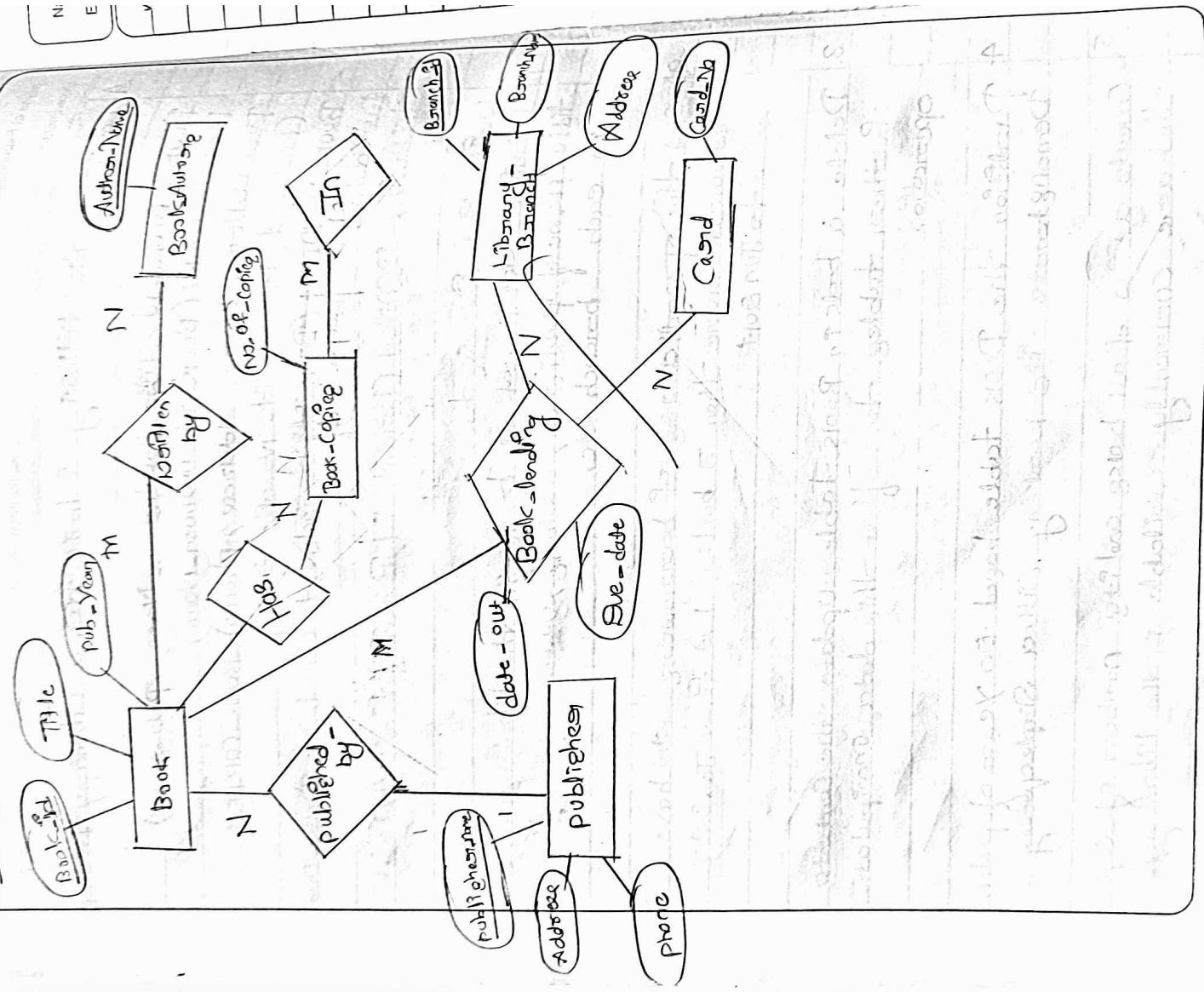
| Expt. No | Date | Title of the Experiment | Page No. | Date of submission | Remarks |
|----------|----------|--|----------|--------------------|---------------------|
| 1 | 10/11/21 | Library Database consisting of 7-tables namely Book, Book_Author, CARO, PUBLISHGR, Book_COPIES, Book_LENDING & LIBRARY, BRANCH and writing SQL Queries | 01-08 | 24/11/21 | S CB 24/11/21 |
| 2 | 24/11/21 | Order Database consisting of 3 tables namely Salesman, CUSTOMER, & ORDER and Executing SQL Queries | 08-14 | 2/12/21 | S CB 2/12/21 |
| 3 | 5/12/21 | Movies Database consisting of 5-tables namely, ACTOR, DIRECTOR, MOVIES, Movie_CAST & RATING | 15-18 | 9/12/21 | S CB 9/12/21 |
| 4 | 9/12/21 | Schema for College Database | 19-24 | 30/12/21 | S CB 30/12/21 |
| 5 | 23/2/21 | Schema for Company Database Employee, DEPT, DEPLOCATION, PROJECT, WORKS_ON | 28-35 | 11/1/22 | S CB 11/1/22 |

REMARKS



S
TEACHER
11/1/22

Entity - Relationship Diagram



Name of Experiment..... LIBRARY DB Date..... 10/11/21
Experiment No..... 01 Experiment Result.....

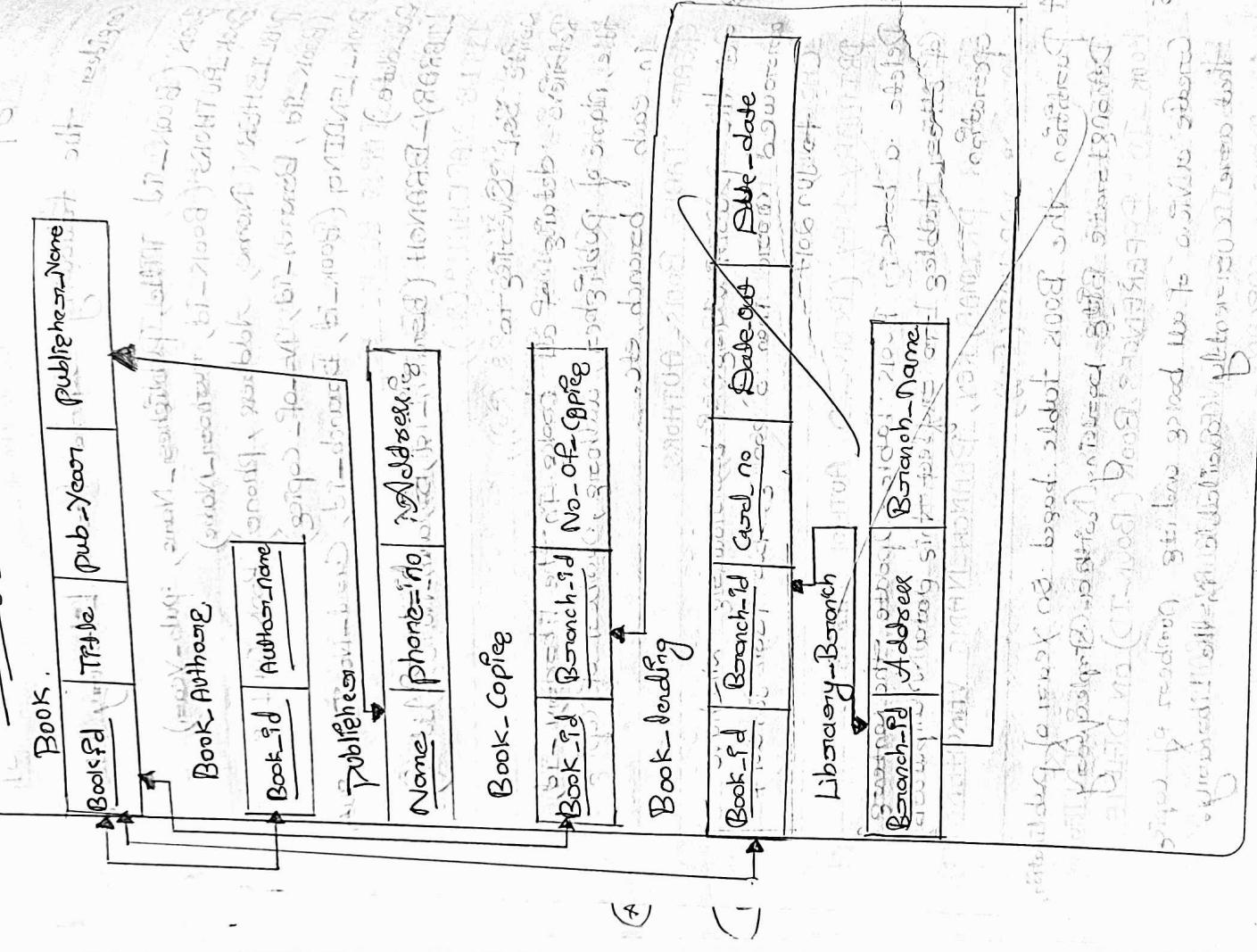
Page No. 01

A Consider the following Schema for a library database.

| | | |
|--|--|--|
| Book (Book_Id, Title, Publisher_Name, Pub_Year) | Author (Author_Id, Author_Name) | LIBRARY_BRANCH (Branch_Id, Branch_Name, Address) |
| Book_AUTHORS (Books_Id, Author_Id) | PUBLISHER (Name, Address, Phone) | Branch_Copies (Books_Id, Branch_Id, No_of_Copies) |
| (Books_Id, Branch_Id, No_of_Copies) | BOOK_LENDING (Books_Id, Branch_Id, Card_No, IssueDate, Due_date) | Branch_Copies (Branch_Id, Books_Id, No_of_Copies) |
| | | |
| 1. Retrieve all books in the library - by title, place of publication, author, number of copies in each branch, etc. | 2. Get the count of books in each branch who have borrowed more than 3 books but less than 4 to Jun 2017 | 3. Delete a book in Books Table. Update the count of other table to reflect the data manipulation operation |
| | | |
| | | 4. Partition the Books Table based on year of publication, Demographic. The partitioning with a Replicability. |
| | | 5. Create a view of all books and the number of copies that are currently available in the library. |

Schemed diagram

- EC (Employee)



| Table Creation | |
|---|--|
| CREATE TABLE PUBLISHER | (NAME VARCHAR(20) . PRIMARY KEY, PHONE INTEGER, ADDRESS VARCHAR(20)); |
| CREATE TABLE Books | (BOOK_ID INTEGER . PRIMARY KEY, TITLE VARCHAR(20), PUB_YEAR VARCHAR(20)), PUBLISHER_NAME REFERENCES PUBLISHER(NAME) ON DELETE CASCADE; |
| CREATE TABLE Books_Authors | (AUTHOR_NAME VARCHAR(20), Books_ID REFERENCES Books_ID) ON DELETE CASCADE, |
| PRIMARY KEY (Books_ID, AUTHOR_NAME); | |
| CREATE TABLE LIBRARY_BRANCH | (BRANCH_ID INTEGER PRIMARY KEY, BRANCH_NAME VARCHAR(50), ADDRESS VARCHAR(50)); |
| CREATE TABLE Books_COPIES (No_OF_COPIES INTEGER, Books_ID REFERENCES Books (Books_ID) ON DELETE CASCADE); | |
| BRANCH_ID REFERENCES LIBRARY_BRANCH . BRANCH_ID ON DELETE CASCADE; | |

PRIMARY KEY (Book-ID, Branch-ID);

CREATE TABLE CARD

(CARD-No INTEGER PRIMARY KEY);

CREATE TABLE Book_LENDING
(DATE_OUT DATE, DATE_IN DATE,
DUE_DATE DATE,
Book-ID REFERENCES Books (Book-ID) ON DELETE CASCADE,
Branch-ID REFERENCES LIBRARY_BRANCH (Branch-ID)
ON DELETE CASCADE,
CARD-No REFERENCES CARD (CARD-No) ON DELETE
CASCADE,
PRIMARY KEY (Book-ID, Branch-ID, CARD-No));

Table Book_Lending

DESC PUBLISHER;
SQL > desc PUBLISHER;
Name: _____
Null?: _____
Type: _____
Not Null: _____
VARCHAR(80)

Phone: _____
Address: _____
Title: _____
Name: _____
Null?: _____
Type: _____
Not Null: _____
VARCHAR(80)

DESC BOOKS;
SQL > desc BOOKS;
Name: _____
Book-ID REFERENCES CARD (CARD-No) NOT NULL,
TITLE VARCHAR(30),
PUB-YEAR INTEGER,
PUBLISHER-NAME VARCHAR(80)

Name of Experiment.....

.....

Experiment No.

.....

Date.....

.....

Page No. 3

| | |
|-------------------------------------|----------------------------|
| DESC Book_Authors; | |
| Sql > DESC Book_Authors; | |
| Name | Null? |
| AUTHOR_NAME | NOTNULL |
| Book_ID | NOTNULL NUMBER(38) |
| Type - SQL | Type - SQL |
| DESC LIBRARY_BRANCH; | |
| Sql > DESC LIBRARY_BRANCH; | SQL > DESC LIBRARY_BRANCH; |
| Name | Null? |
| BRANCH_ID | NOTNULL NUMBER(38) |
| BRANCH_NAME | VARCHAR2(50) |
| ADDRESS | VARCHAR2(100) |
| DEPARTMENT | VARCHAR2(50) |
| DESC Books_COPIES; | DESC Books_COPIES; |
| Sql > DESC Books_COPIES; | SQL > DESC Books_COPIES; |
| Name | Null? |
| BRANCH_ID | NOTNULL NUMBER(38) |
| Book_ID | NOTNULL NUMBER(38) |
| No_OF_COPIES | NOTNULL NUMBER(38) |
| Type - SQL | Type - SQL |
| DESC CARD; | |
| Sql > DESC CARD; | SQL > DESC CARD; |
| Name | Null? |
| CARD_NO. | NOTNULL NUMBER(38) |
| Type - SQL | Type - SQL |
| SQL: 'SELECT * FROM CARD' | |
| Ans: (CARD_ID, CARD_NO., CARD_DATE) | |

DESC Books_LENDING;

Sql>desc books_lending;

Name-----+-----+-----+-----+-----+-----+-----+

DATE_OUT

DATE_IN

DUE_DATE

BOOK_ID

BRANCH_ID

CARD_NO.

Temporary of value to table

INSERT INTO PUBLISHER VALUES ('MCRAW-HILL',
6587, 'BANGALORE');

INSERT INTO PUBLISHER VALUES ('PERSON', 98890766,
'NEWDELI');

INSERT INTO PUBLISHER VALUES ('RANJDOM HOUSE',
9345, 'HYDRABAD');

INSERT INTO PUBLISHER VALUES ('HACHETTE LIBRARY',
8970568340, 'CHENNAI');

INSERT INTO PUBLISHER VALUES ('GROUPOPLANETA',
838, 'BANGALORE');

INSERT INTO Book VALUES (1, 'DBMS', 'JAN-80', 'MCRAW-
HILL');

INSERT INTO Book VALUES (2, 'ADAMS', 'JUN-80',
'MCRAW-HILL');

INSERT INTO Books VALUES (3, 'CN', "Sep-Sale", 'Person');

Name of Experiment.....

Date.....

Experiment No.....

Experiment Result.....

Page No. 4

```
INSERT INTO Books VALUES ('CCN', 'GEP-005', 'GRUPP  
PLANETA');  
INSERT INTO Books VALUES ('S', '08', 'MAY-001', 'PEARSON');  
  
INSERT INTO Books_Authors VALUES ('NAVATHE', '1');  
INSERT INTO Books_Authors VALUES ('DIAVATHA', '2');  
INSERT INTO Books_Authors VALUES ('TANENBAUM', '3');  
INSERT INTO Books_Authors VALUES ('EDWARD ANGEL', '4');  
INSERT INTO Books_Authors VALUES ('KALYAN', '5');  
  
INSERT INTO LIBRARY_BRANCH VALUES (19, 'RR NAGAR',  
'BANGALORE');  
INSERT INTO LIBRARY_BRANCH VALUES (11, 'RNSIT',  
'BANGALORE');  
INSERT INTO LIBRARY_BRANCH VALUES (12, 'RAJAJI  
NAGAR', 'BANGALORE');  
INSERT INTO LIBRARY_BRANCH (13, 'NITIE', 'MANASA  
LORE');  
INSERT INTO LIBRARY_BRANCH VALUES (14, 'MANIPAL  
INDIA');  
  
INSERT INTO Books_COPIES VALUES (10, 1, 10);  
INSERT INTO Books_COPIES VALUES (15, 1, 11);  
INSERT INTO Books_COPIES VALUES (20, 2, 12);  
INSERT INTO Books_COPIES VALUES (5, 2, 13);  
INSERT INTO Books_COPIES VALUES (7, 3, 14);  
INSERT INTO Books_COPIES VALUES (15, 10);  
INSERT INTO Books_COPIES VALUES (3, 4, 11);
```

INSERT INTO CARD VALUES (100);
INSERT INTO CARD VALUES (101);
INSERT INTO CARD VALUES (102);
INSERT INTO CARD VALUES (103);
INSERT INTO CARD VALUES (104);

INSERT INTO Books-LENDING VALUES ('61-JAN-17',
-17, 10, 101);
INSERT INTO Books-LENDING VALUES ('11-JAN-17',
17, 3, 14, 101);
INSERT INTO Books-LENDING VALUES ('81-FEB-17',
APR-17, 8, 13, 101);

INSERT INTO Book-LENDING VALUES ('15-MAR-17',
15-JUL-17, 4, 11, 101);

INSERT INTO Books-LENDING VALUES ('19-APR-17',
19-MAY-17, 11, 104);
INSERT INTO Books-LENDING VALUES ('19-APR-17',
19-MAY-17, 11, 104);

SELECT * FROM PUBLISHGR;
NAME - - - - -
MICORARIO-HILL
PERSON - - - - -
RAM DOM HOUSE
HACHETTE-LEVERZUHA
GROUP PLANETA
CITY - - - - -
NEWDELI
MADRAS
HYDRABAD
CHENNAI
BANGALORE
POODHUR
MUMBAI
KOLKATA
CALCUTTA
DELHI
PUNE
GOA
BOMBAY
CHENNAI
BANGALORE
FRESHL

Experiment Result.....

| | | | | | |
|--|-----------------------|--------------------|--------------------|------------------|-----------------------|
| <u>SELECT</u> * <u>FROM</u> Books; | <u>Books</u> | <u>Book ID</u> | <u>Title</u> | <u>PUB YEAR</u> | <u>PUBLISHER NAME</u> |
| 1. DRMS | JAN-2016 | MCGRAW-HILL | | | |
| 2. APBMS | JUN-2016 | MCGRAW-HILL | | | |
| 3. CN | SEP-2016 | PEARSON | | | |
| 4. CG | SEP-2015 | GRUPO PLANETA | | | |
| 5. DS | MAY-2016 | PERSON | | | |
| <u>SELECT</u> * <u>FROM</u> Books AUTHORS; | <u>Books</u> | <u>Author ID</u> | <u>Author Name</u> | <u>Book ID</u> | <u>Book Title</u> |
| <u>SQL</u> \rightarrow <u>SELECT</u> * <u>FROM</u> Books AUTHORS; | | | | | |
| <u>Author</u> | <u>Name</u> | <u>Book ID</u> | <u>Title</u> | <u>Author ID</u> | <u>Book Title</u> |
| NAVATHA | 1 | DRMS | JAN-2016 | 1 | DRMS |
| NAVATHA | 2 | MCGRAW-HILL | MCGRAW-HILL | 2 | MCGRAW-HILL |
| TANENDRAUM | 3 | MCGRAW-HILL | MCGRAW-HILL | 3 | MCGRAW-HILL |
| GDWARD ANGEL | 4 | PEARSON | PEARSON | 4 | PEARSON |
| GARVIN | 5 | GRUPO PLANETA | GRUPO PLANETA | 5 | GRUPO PLANETA |
| <u>SELECT</u> * <u>FROM</u> LIBRARY_BRANCH; | <u>LIBRARY_BRANCH</u> | <u>Branch ID</u> | <u>Branch Name</u> | <u>Address</u> | <u>City</u> |
| <u>SQL</u> \rightarrow <u>SELECT</u> * <u>FROM</u> LIBRARY_BRANCH; | | | | | |
| <u>Branch</u> | <u>ID</u> | <u>Branch Name</u> | <u>Address</u> | <u>City</u> | <u>Branch ID</u> |
| RENSAER | 1001 | VINYL | BANGALORE | BANGALORE | 1001 |
| TRANSIT | 1101 | 1-800 | BANGALORE | BANGALORE | 1101 |
| RAJTEL NASHAR | 1201 | 1994 | BANGALORE | BANGALORE | 1201 |
| NETTE | 1301 | 1991 | MANGALORE | MANGALORE | 1301 |
| MANIPAL | 1401 | 1991 | UDUPI | UDUPI | 1401 |

SELECT * FROM BOOK_COPIES;

SQL> SELECT * FROM BOOK_COPIES;

| No_of_Copies | Books_ID | Branch_ID | Count |
|--------------|----------|-----------|-------|
| 10 | 1 | 10 | 10 |
| 5 | 1 | 11 | 5 |
| 8 | 1 | 12 | 8 |
| 5 | 2 | 10 | 5 |
| 7 | 3 | 9 | 12 |
| 3 | 4 | 11 | 14 |
| 3 | 5 | 10 | 10 |
| 3 | 6 | 11 | 11 |
| 3 | 7 | 12 | 12 |
| 3 | 8 | 13 | 13 |
| 3 | 9 | 14 | 14 |
| 3 | 10 | 15 | 15 |
| 3 | 11 | 16 | 16 |
| 3 | 12 | 17 | 17 |
| 3 | 13 | 18 | 18 |
| 3 | 14 | 19 | 19 |
| 3 | 15 | 20 | 20 |
| 3 | 16 | 21 | 21 |
| 3 | 17 | 22 | 22 |
| 3 | 18 | 23 | 23 |
| 3 | 19 | 24 | 24 |
| 3 | 20 | 25 | 25 |
| 3 | 21 | 26 | 26 |
| 3 | 22 | 27 | 27 |
| 3 | 23 | 28 | 28 |
| 3 | 24 | 29 | 29 |
| 3 | 25 | 30 | 30 |
| 3 | 26 | 31 | 31 |
| 3 | 27 | 32 | 32 |
| 3 | 28 | 33 | 33 |
| 3 | 29 | 34 | 34 |
| 3 | 30 | 35 | 35 |
| 3 | 31 | 36 | 36 |
| 3 | 32 | 37 | 37 |
| 3 | 33 | 38 | 38 |
| 3 | 34 | 39 | 39 |
| 3 | 35 | 40 | 40 |
| 3 | 36 | 41 | 41 |
| 3 | 37 | 42 | 42 |
| 3 | 38 | 43 | 43 |
| 3 | 39 | 44 | 44 |
| 3 | 40 | 45 | 45 |
| 3 | 41 | 46 | 46 |
| 3 | 42 | 47 | 47 |
| 3 | 43 | 48 | 48 |
| 3 | 44 | 49 | 49 |
| 3 | 45 | 50 | 50 |
| 3 | 46 | 51 | 51 |
| 3 | 47 | 52 | 52 |
| 3 | 48 | 53 | 53 |
| 3 | 49 | 54 | 54 |
| 3 | 50 | 55 | 55 |
| 3 | 51 | 56 | 56 |
| 3 | 52 | 57 | 57 |
| 3 | 53 | 58 | 58 |
| 3 | 54 | 59 | 59 |
| 3 | 55 | 60 | 60 |
| 3 | 56 | 61 | 61 |
| 3 | 57 | 62 | 62 |
| 3 | 58 | 63 | 63 |
| 3 | 59 | 64 | 64 |
| 3 | 60 | 65 | 65 |
| 3 | 61 | 66 | 66 |
| 3 | 62 | 67 | 67 |
| 3 | 63 | 68 | 68 |
| 3 | 64 | 69 | 69 |
| 3 | 65 | 70 | 70 |
| 3 | 66 | 71 | 71 |
| 3 | 67 | 72 | 72 |
| 3 | 68 | 73 | 73 |
| 3 | 69 | 74 | 74 |
| 3 | 70 | 75 | 75 |
| 3 | 71 | 76 | 76 |
| 3 | 72 | 77 | 77 |
| 3 | 73 | 78 | 78 |
| 3 | 74 | 79 | 79 |
| 3 | 75 | 80 | 80 |
| 3 | 76 | 81 | 81 |
| 3 | 77 | 82 | 82 |
| 3 | 78 | 83 | 83 |
| 3 | 79 | 84 | 84 |
| 3 | 80 | 85 | 85 |
| 3 | 81 | 86 | 86 |
| 3 | 82 | 87 | 87 |
| 3 | 83 | 88 | 88 |
| 3 | 84 | 89 | 89 |
| 3 | 85 | 90 | 90 |
| 3 | 86 | 91 | 91 |
| 3 | 87 | 92 | 92 |
| 3 | 88 | 93 | 93 |
| 3 | 89 | 94 | 94 |
| 3 | 90 | 95 | 95 |
| 3 | 91 | 96 | 96 |
| 3 | 92 | 97 | 97 |
| 3 | 93 | 98 | 98 |
| 3 | 94 | 99 | 99 |
| 3 | 95 | 100 | 100 |
| 3 | 96 | 101 | 101 |
| 3 | 97 | 102 | 102 |
| 3 | 98 | 103 | 103 |
| 3 | 99 | 104 | 104 |

SELECT * FROM CARD; : EXCUTED - 2009-04-17 11:45:45 AM

SQL> SELECT * FROM CARD; : EXCUTED - 2009-04-17 11:45:45 AM

CARD_No

100
101
102
103
104.

DATE-IN

2009-04-17
2009-04-17
2009-04-17
2009-04-17
2009-04-17

DATE-OUT

2009-04-17
2009-04-17
2009-04-17
2009-04-17
2009-04-17

Books_ID

1
1
1
1
1

Branch_ID

10
10
10
10
10

CARD_No

101
101
101
101
101

DATE-IN

2009-04-17
2009-04-17
2009-04-17
2009-04-17
2009-04-17

DATE-OUT

2009-04-17
2009-04-17
2009-04-17
2009-04-17
2009-04-17

Books_ID

1
1
1
1
1

Branch_ID

10
10
10
10
10

CARD_No

101
101
101
101
101

DATE-IN

2009-04-17
2009-04-17
2009-04-17
2009-04-17
2009-04-17

DATE-OUT

2009-04-17
2009-04-17
2009-04-17
2009-04-17
2009-04-17

Name of Experiment..... Experiment No.....

Date..... 15-10-2019 Page No. 6

Ques:- Write a query to find details of all books in the library and their title, name of publisher, author, number of copies in each branch etc.

SELECT B.Book_ID, B.Title, B.Publisher_Name,
A.Author_Name, C.No_of_Copies, L.Branch_ID
FROM Books B, Books_Authors A, Books_Copies C,
Library_Branch L
WHERE Books_ID = A.Book_ID
AND B.Books_ID = C.Book_ID
AND L.Branch_ID = C.Branch_ID;

| Book_ID | Title | Publisher_Name | Author_Name | No_of_Copies | Branch |
|---------|-------|----------------|---------------|--------------|--------|
| 1 | DRMS | McGraw-Hill | NAVATH | 5 | 11 |
| 2 | DRMS | McGraw-Hill | NAVATH | 5 | 12 |
| 3 | ADAMS | McGraw-Hill | NAVATH | 5 | 13 |
| 4 | ADAMS | McGraw-Hill | NAVATH | 5 | 14 |
| 5 | CN | PEARSON | THAKUR | 7 | 15 |
| 6 | OS | PEARSON | GALVIN | 10 | 16 |
| 7 | 4 | LIT | GRUPO PLANETA | EDWARD ANGEL | 3 |
| 8 | CG | LIT | GRUPO PLANETA | EDWARD ANGEL | 4 |

1. Get the particulars of books which have more than 3 books, but from Jan 2017 onwards.

Ans:-
1. To find particular books which have more than 3 books, but from Jan 2017 onwards.

```
SELECT CARD-No FROM  
Book_LENDING
```

WHERE DATE_OUT BETWEEN '01-JAN-00' AND '01-JULY-00'
GROUP BY CARD-No

HAVING COUNT(*) > 3
AND CARD-No
= 101

⑤ Delete a book in Book Table. Update the contents of the
Table to reflect the data manipulation operation.

DELETE FROM Book
WHERE Book-ID = 3;
Book-ID = 3; Book-ID = 4; Book-ID = 5;

SQL> DELETE FROM Book WHERE Book-ID = 3;

& WHERE Book-ID = 3;
1> Book-ID = 3; Book-ID = 4; Book-ID = 5;
1> Book-ID = 3; Book-ID = 4; Book-ID = 5;
1> Book-ID = 3; Book-ID = 4; Book-ID = 5;

SQL> SELECT * FROM Book;

Book-ID TITLE PUBLISHER-NAME
4 JUNGLE BOOK MCGRAW-HILL
5 ADAMSSON JUNGLE BOOK MCGRAW-HILL
6 - SEP - 0015 GROUP PHANTOM

5 Sept 00 Group Phantasm - 0015 appears.
High cost meet the goal.

⑥ Partition the Book table based on year of publication
Demonstrate the use of partitioning with a simple query

Name of Experiment..... Experiment No.....

Date..... Experiment Result.....

Page No. 7

CREATE VIEW PUBLICATION AS SELECT

PUB_YEAR

From Books

PUB_YEAR

JAN - 2017

JUN - 2016

Sep - 2016

Sep - 2015

May - 2016

4] Create a view of all books & the number of copies
that are currently available in the library

CREATE VIEW Books AS

SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES

FROM Book B, Books_C, LIBRARY_BRANCH L

WHERE B.BOOK_ID = C.BOOK_ID

AND C.BRANCH_ID = L.BRANCH_ID;

Select * from Books;

Books_ID Title No_of_Copies

1 DBMS 10

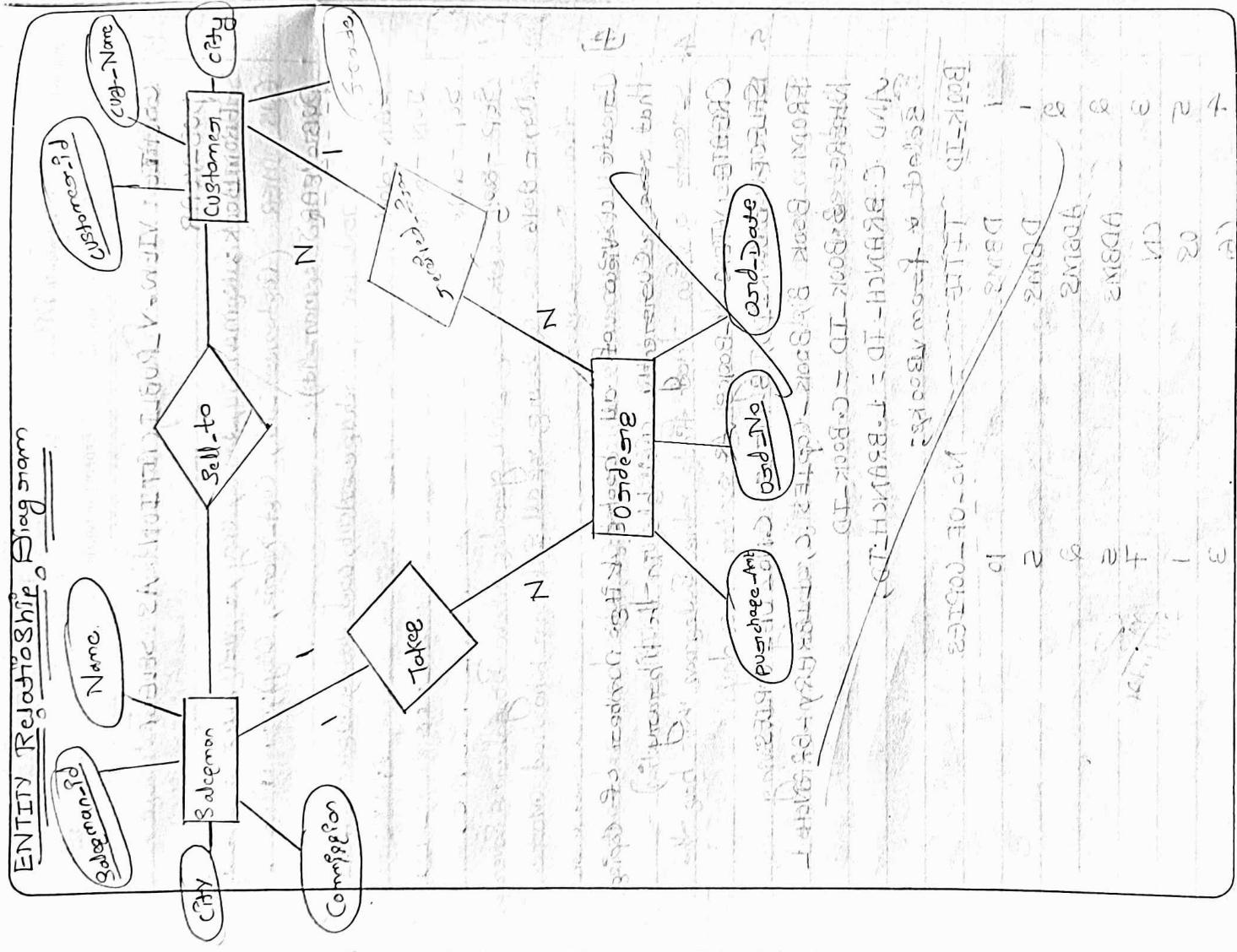
2 DBMS 5

3 DBMS 7

4 OS 1

4 CG

3



Name of Experiment... Q7.....DB.....
Experiment No....B.....8.....

Date.....8/11/11.....
Page No.8.....

B Consider the following Schema for OrderDBase :-

SALESMAN (Salesman_id, Name, City, Commision)
CUSTOMER (Customer_id, Cust_Name, City)
GRADE (Salesman_id)
ORDERS (ord_no, Purchaser_Amt, ord_date, Customer_id,
Salesman_id).
In this SQL Query to.

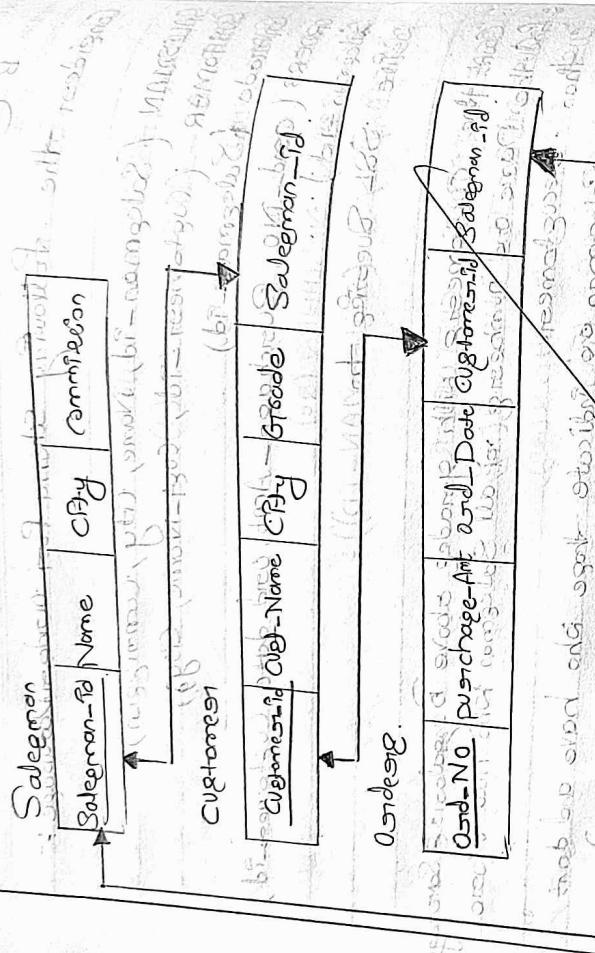
1. Count the customers with grade above Bangalore & Service
2. Find the Name and number of all Salesman who had more than 5 employees.
3. List all Salesman and indicate those who have and don't have Customer in their City (use UNION operation)
4. Calculate average that finds the Salesman who has the highest number of a day.
5. Deregister the DELETE operation by increasing Salesman with Id 1000. All his orders must also be deleted.

Solution
=====

Schema Diagramm B

Order-Header
Order-Detail

Customer
Salesman



ER

Table Creation

```

CREATE TABLE SALESMAN
(SALESMAN-ID NUMBER(4),
NAME VARCHAR2(20),
CITY VARCHAR2(20),
COMMISSION NUMBER(8,0),
PRIMARY KEY (SALESMAN-ID));
NOKRITI WED

CREATE TABLE CUSTOMER
(CUSTOMER-ID NUMBER(4),
CUS_NAME VARCHAR2(20),
CITY VARCHAR2(20),
GRADE NUMBER(3),
PRIMARY KEY (Customer-ID),
SALESMAN-ID REFERENCES SALESMAN(SALESMAN-ID))
ON DELETE SET NULL;
NOKRITI WED

CREATE TABLE ORDERS
(ORD-No NUMBER(5),
PURCHASE_AMT NUMBER(10,2),
ORD_DATE DATE,
PRIMARY KEY (ORD-No),
Customer-ID REFERENCES CUSTOMER(CUSTOMER-ID)
ON DELETE CASCADE,
SALESMAN-ID REFERENCES SALESMAN(SALESMAN-ID)
ON DELETE CASCADE);
NOKRITI WED

```

Table Description

SQL > DESC SALESMAN;

Customer Information

| Name | Type |
|-------------|---|
| SALESMAN_ID | (+) NUMBER(3,0) |
| NAME | NOT NULL VARCHAR(4) |
| CITY | (0)B(25) VARCHAR(15), CITY(SIZE) VARCHAR(15) |
| COMMISSION | :((1-MIN(0)) NUMBER(3,0)) |

SQL > DESC CUSTOMER;

| Name | Type |
|-------------|---|
| CUSTOMER_ID | (0) NUMBER(3,0) |
| CUST_NAME | NOT NULL VARCHAR(15) |
| CITY | (0)B(25) VARCHAR(15), CITY(SIZE) VARCHAR(15) |
| GRADE | (0) NUMBER(3,0) |
| SALESMAN_ID | (0) NUMBER(3,0) |

DESC ORDERS;

| Name | Type |
|--------------|--------------------|
| ORDINONID | NOT NULL NUMBER(5) |
| PURCHASE_AMT | NUMBER(10,2) |
| ORD_DATE | DATE |
| CUSTOMER_ID | NUMBER(4) |
| SALESMAN_ID | NUMBER(4) |

Name of Experiment..... Date..... Experiment Result.....

Page No. 10

Experiment No.....

Insertion of Values To Table

```
INSERT INTO SALESMAN VALUES (1000, 'JOHN', 'BANGALORE', '95%');
INSERT INTO SALESMAN VALUES (2000, 'RAVI', 'BANGALORE', '90%');
INSERT INTO SALESMAN VALUES (3000, 'KUMAR', 'MYSORE', '15%');
INSERT INTO SALESMAN VALUES (4000, 'SMITH', 'DELI', '30%');
INSERT INTO SALESMAN VALUES (5000, 'HARSHA', 'HYDRAVABAD', '15%');
INSERT INTO CUSTOMER VALUES (10, 'PREETHI', 'BANGALORE', '100, 100);
INSERT INTO CUSTOMER VALUES (11, 'VIVEK', 'MANGALORE', '300, 100);
INSERT INTO CUSTOMER VALUES (12, 'BHOGESWAR', 'CHENNAI', '400, 800);
INSERT INTO CUSTOMER VALUES (13, 'CHEETHAM', 'BANGALORE', '800, 800);
INSERT INTO CUSTOMER VALUES (14, 'MANATHA', 'BANGALORE', '400, 300);

INSERT INTO ORDERS VALUES (50, 5000, '04-MAY-17', '10,1000);
INSERT INTO ORDERS VALUES (51, 450, '02-JAN-17' '10, 8000);
INSERT INTO ORDERS VALUES (52, 1000, '04-FEB-17', '13,8000);
INSERT INTO ORDERS VALUES (53, 3500, '13-APR-17', '14, 3200);
INSERT INTO ORDERS VALUES (54, 3500, '09-MAR-17', '10, 8000);

SELECT * FROM SALESMAN;
```

| SALESMAN_ID | NAME | CITY | COMMISSION |
|-------------|--------|------------|------------|
| 1000 | JOHN | BANGALORE | 95% |
| 2000 | RAVI | BANGALORE | 90% |
| 3000 | KUMAR | MYSORE | 95% |
| 4000 | SMITH | HYDRAVABAD | 30% |
| 5000 | HARSHA | DELI | 15% |

| CUSTOMER-ID | CUST-NAME | CITY | GRADE | SALESMAN-ID |
|-------------|-----------|-----------|-------|-------------|
| 10 | PREETHI | BANGALORE | 100 | 1000 |
| 11 | VINEK | MANGALORE | 300 | 1000 |
| 12 | BHASKAR | CHENNAI | 400 | 800 |
| 13 | CHETHAN | BANGALORE | 800 | 800 |
| 14 | CHAMATHA | BANGALORE | 400 | 3000 |

SELECT * FROM ORDERS;

| ORDNO | PURCHASE-AMT | ORD-DATE | CUSTOMER-ID | SALESMAN-ID |
|-------|--------------|-----------|-------------|-------------|
| 50 | 5000 | 04-MAY-17 | 10 | 1000 |
| 51 | 4500 | 08-JAN-17 | 10 | 8000 |
| 52 | 10000 | 04-FEB-17 | 13 | 8000 |
| 53 | 3500 | 13-APR-17 | 14 | 3000 |
| 54 | 5500 | 09-MAR-17 | 10 | 8000 |

Ques: Count the number of customers having grade above Borgobore

1. Count the customers having grade above Borgobore

2. SELECT GRADE, COUNT(DISTINCT CUSTOMER-ID) AS COUNT
FROM CUSTOMER,

GROUP BY GRADE HAVING GRADE > ('S' & 'G')

FROM CUSTOMER

WHERE CITY = 'BANGALORE') AS COUNT

GRADE COUNT(DISTINCT CUSTOMER-ID)

| | | |
|-----|-------|---|
| 300 | MANOJ | Y |
| 400 | VEEKA | X |

| | | |
|-----|-------|---|
| 300 | MANOJ | Y |
| 400 | VEEKA | X |
| 400 | MANOJ | Y |
| 400 | VEEKA | X |
| 400 | MANOJ | Y |
| 400 | VEEKA | X |
| 400 | MANOJ | Y |
| 400 | VEEKA | X |
| 400 | MANOJ | Y |
| 400 | VEEKA | X |
| 400 | MANOJ | Y |
| 400 | VEEKA | X |
| 400 | MANOJ | Y |
| 400 | VEEKA | X |

MANOJ VEEKA MANOJ VEEKA MANOJ VEEKA MANOJ VEEKA

MANOJ VEEKA MANOJ VEEKA MANOJ VEEKA MANOJ VEEKA

2. Find the name and number of all Salesman 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);
```

| SALESMAN_ID | NAME |
|-------------|------|
| 1000 | JOHN |
| 8000 | RANT |

3. List all Salesmen and indicate those who have and don't have customers. Give (use UNION operation)

```
SELECT SALESMAN_ID, NAME, CUST_NAME,
       COUNT(CUST_ID) AS CNT
  FROM SALESMAN
 WHERE CNT >= 1
UNION
SELECT CUSTOMER_ID, CUSTOMER_NAME, CITY
  FROM CUSTOMER
 WHERE SALESMAN_ID = CUSTOMER_ID;
```

| SALESMAN_ID | NAME | CNT |
|-------------|------|-----|
| 1000 | JOHN | 2 |
| 8000 | RANT | 1 |

OOOG H - RANT - PO

| SALESMAN-ID | NAME | CUSTOMERNAME | Commission |
|-------------|--------|--------------|------------|
| 4000 | SMITH | No MATCH | 30% |
| 8000 | RAVI | CHETHAN | 30% |
| 8000 | RAVI | MAMATHA | 30% |
| 8000 | RAVI | PREETHI | 30% |
| 3000 | KUMAR | No MATCH | 15% |
| 1000 | JOHN | CHETHAN | 30% |
| 1000 | JOHN | MAMATHA | 30% |
| 1000 | JOHN | PREETHI | 30% |
| 5000 | HARSHA | No MATCH | 15% |

(a) Create a view that finds the Salesman who has the highest amount of order.

CREATE VIEW GLASSALESMAN AS
 SELECT B-ORD-DATE, A-SALESMAN-ID, A-NAME
 FROM SALESMA A, ORDERS B
 WHERE A-SALESMAN-ID = B-SALESMAN-ID
 AND B-PURCHASE-AMT = (SELECT MAX(PURCHASE-AMT)
 FROM ORDERS)

CREATE TABLE C-ORD-DATE = B-ORD-DATE,
 ORD-DATE(SALESMAN-ID NAME CT-MANAGE
 04-04-17 1000 JOHN MANAGER
 20-JAN-17 8000 RAVI SALESMAN
 24-FEB-17 2000 RAVI SALESMAN
 13-APR-17 3000 KUMAR SALESMAN
 09-MAR-17 2000 RAVI SALESMAN

Q) Demonstrate The DELETE Operations by cascading Salesman with id 1000. All the orders must also be deleted.

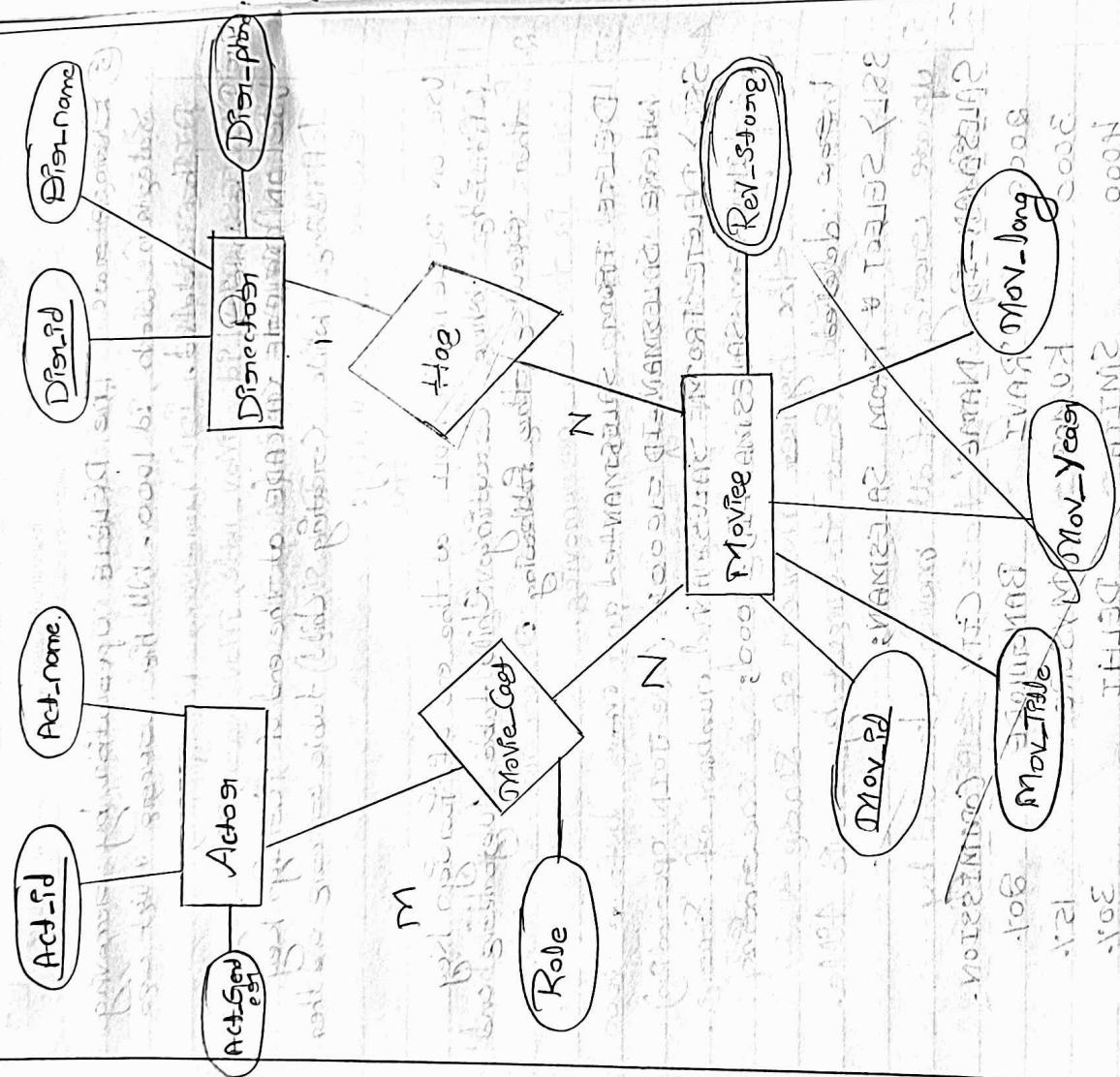
Use on DELETE CASCADE at the end of foreign key definition while creating Child table cascade and then execute the following:

Use on DELETE SET NULL at the end of foreign key definition while creating Child table cascading and then execute the following :

```
DELETE FROM SALESMAN
WHERE SALESMAN-ID = 1000;
SELECT * FROM SALESMAN
WHERE SALESMAN-ID = 1000;
1 row deleted
381) SELECT * FROM SALESMAN;
```

| CITY - COMMISSION - | | |
|---------------------|-------|-----------|
| SALESMAN-ID | Name | CITY |
| 2000 | RAVI | BANGALORE |
| 3000 | KUMAR | MYSORE |
| 4000 | SMITH | DELHI |
| 5000 | HARRY | HYDRABAD |

Date : 21-07-2012
Page No. : 12



2000 HYBRID HADEBUD 12.5
4000 SINTER DERHI 30.5
3000 12.5
2000 12.5

5/10/51

C Consider the Schema from MovieDatabase:

| |
|--|
| ACTOR (Act_Id, Act_Name, Act_Bienden) |
| DIRECTOR (Dir_Id, Dir_Name, Dir_Phone) |
| MOVIES (Mov_Id, Mov_Title, Mov_Year, Mov_Lang, Dir_Id, Movie_Cat (Act_Id, Mov_Id, Role)) |
| RATINGS (Mov_Id, Rev_Score) |

Write SQL Queries to:

1. Get the Title of all movies directed by 'Hitchcock'
2. Find the movie name whose one or more actress acted in two or more movies.
3. List all actress who acted in more than 8000 and also in a decade after 2015 (use JOIN operation)
4. Find the title of movie and number of shows for each movie that has at least one showing and find the highest number of shows that movie received & the count by movie title.

5. Update modify 'all movie directed by 'Steven Spielberg' to 5.
- ~~'Steven Spielberg' to 5.~~

Schema Diagram

Action

| Act-Id | Act-Name | Act-Gender |
|--------|-------------|-------------|
| Actor | Actor Name | Male/Female |
| Dancer | Dancer Name | Male/Female |

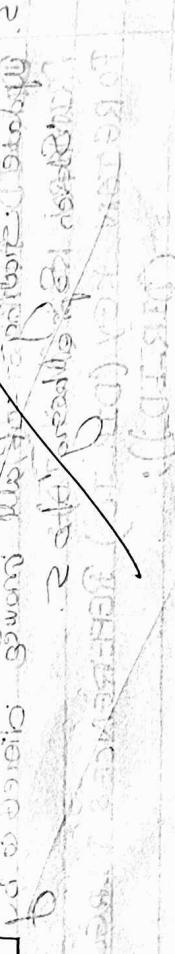
| Dir-Id | Dir-Name | Dir-Phone |
|----------|---------------|----------------|
| Director | Director Name | Director Phone |
| Producer | Producer Name | Producer Phone |

Movie.

| Mov-Id | Mov-Title | Mov-Year | Mov-Lang | Mov-Prod |
|--------|-------------|------------|------------|------------|
| Movie | Movie Title | Movie Year | Movie Lang | Movie Prod |
| Actor | Actor Name | Actor Year | Actor Lang | Actor Prod |

| Act-Id | Mov-Id | Role |
|--------|--------|-------------|
| Actor | Movie | Actor Role |
| Dancer | Movie | Dancer Role |

| Mov-Id | Ren-Stage |
|--------|-----------------|
| Movie | Movie Ren Stage |



Elements (11 total) from diagram:

1. Action
2. Director
3. Dancer
4. Movie
5. Actor
6. Producer
7. Role
8. Language
9. Year
10. Stage

Relationships:

1. Action → Director
2. Action → Dancer
3. Action → Movie
4. Director → Dancer
5. Director → Movie
6. Dancer → Movie
7. Role → Actor
8. Language → Movie
9. Year → Movie
10. Stage → Movie

Table Creation

```

CREATE TABLE ACTOR(
ACT_ID NUMBER(5),
ACT_NAME VARCHAR(80),
ACT_GENDER CHAR(1),
PRIMARY KEY (ACT_ID));
CREATE TABLE DIRECTOR(
DIR_ID NUMBER(3),
DIR_NAME VARCHAR(80),
DIR_PHONE NUMBER(10),
PRIMARY KEY (DIR_ID));
CREATE TABLE MOVIES(
MOV_ID NUMBER(4),
MOV_TITLE VARCHAR(25),
MOV_YEAR NUMBER(4),
MOV_LANG VARCHAR(12),
DIR_ID NUMBER(3),
PRIMARY KEY (MOV_ID),
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR);
CREATE TABLE MOVIE_CAST (ACT_ID NUMBER(3),
MOV_ID NUMBER(4),
ROLE VARCHAR(10),
PRIMARY KEY(FACT_ID, MOV_ID),
FOREIGN KEY(ACT_ID) REFERENCES ACTOR (ACT_ID),
FOREIGN KEY(MOV_ID) REFERENCES MOVIES (MOV_ID));

```

FOREIGN KEY

(MOV_ID) REFERENCES MOVIES (MOV_ID);

CREATE TABLE RATINGS (
 MOV_ID NUMBER (4),
 RER_STARS VARCHAR (85),
 PRIMARY KEY (MOV_ID),
 FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID);

Table : Reference

| Name | Act-ID | Dir-ID | Movie-ID | Rating |
|---------------------|---|---|--------------------------|---------------------------|
| DESC ACTOR; | (s1) ACT_NAME CHAR(80) (s2) ACT_GENDER CHAR(1) (s3) ACT_MIA CHAR(1) | (s1) DIR_NAME CHAR(80) (s2) DIR_PHONE CHAR(10) | (s1) MOVIE_ID NUMBER (3) | (s1) RER_STARS NUMBER (3) |
| SQL> DESC ACTOR; | | | | |
| SQL> DESC DIRECTOR; | | | | |
| SQL> DESC MOVIE; | | | | |
| SQL> DESC RATING; | | | | |

NOT NULL

CHAR(80)

CHAR(1)

CHAR(1)

NUMBER (3)

NUMBER (3)

CHAR(10)

Name of Experiment.....

Date.....

Experiment Result.....

Page No. 15

| | | |
|---------------------------------|---------------------------------|---------------------------------------|
| DESC MOVIES; | | |
| SQL> DESC MOVIES; | COL1 MOV_ID NUMBER(10) NOT NULL | COL2 MOV_TITLE VARCHAR2(80) |
| | | COL3 MOV_YEAR NUMBER(4) |
| | | COL4 MOV_LANG VARCHAR2(10) |
| | | COL5 DIR_ID NUMBER(3) |
| DESC MOVIE_CAST; | | |
| SQL> DESC MOVIE_CAST; | COL1 ACT_ID NUMBER(10) NOT NULL | COL2 ACT_NAME VARCHAR2(50) NOT NULL |
| | | COL3 MOV_ID NUMBER(10) NOT NULL |
| | | COL4 ACT_ROLE VARCHAR2(10) NOT NULL |
| | | COL5 REV_RATING NUMBER(2) NOT NULL |
| DESC RATINGS; | | |
| SQL> DESC RATINGS; | COL1 REV_ID NUMBER(10) NOT NULL | COL2 REV_TITLE VARCHAR2(100) NOT NULL |
| | | COL3 REV_YEAR NUMBER(4) NOT NULL |
| | | COL4 REV_STARS NUMBER(10) NOT NULL |
| | | COL5 REV_RATING NUMBER(2) NOT NULL |

Insertion of value to table

```

INSERT INTO FACTOR VALUES (301, 'ANUSHKA','F');
INSERT INTO ACTOR VALUES (302, 'PRABHAS', 'M');
INSERT INTO ACTOR VALUES (303, 'PUNITH', 'M');
INSERT INTO ACTOR VALUES (304, 'TERRY', 'M');

```

```

INSERT INTO DIRECTOR VALUES (60, 'RAJMOULI', 845169100);
INSERT INTO DIRECTOR VALUES (61, 'HITCHCOCK', 746613891);
INSERT INTO DIRECTOR VALUES (62, 'FARAN', 998644653);
INSERT INTO DIRECTOR VALUES (63, 'STEVEN', 'SPIELBERG');
8989746530;

```

```

INSERT INTO MOVIES VALUES (1001, 'BAHUBALI-2', 2017, 'TELEVISION', 60);
INSERT INTO MOVIES VALUES (1002, 'BAHUBALI-1', 2015, 'TELEVISION', 61);
INSERT INTO MOVIES VALUES (1003, 'AKASH', 2008, 'KANNADA', 61);
INSERT INTO MOVIES VALUES (1004, 'WAR HORSE', 2011, 'ENGLISH', 63);

```

```

INSERT INTO MOVIE_CAST VALUES (301, 100, 'HEROINE');
INSERT INTO MOVIE_CAST VALUES (302, 100, 'HEROINE');
INSERT INTO MOVIE_CAST VALUES (303, 100, 'HERO');
INSERT INTO MOVIE_CAST VALUES (303, 100, 'GUEST');
INSERT INTO MOVIE_CAST VALUES (304, 100, 'HERO');

```

Name

Expri

Name of Experiment.....

Page No.

Date.....

Experiment No.....

Page No.

.....

| | | | | | |
|-------------------------------------|--------|-------------------|------------|------------|--------|
| INSERT INTO RATING VALUES (1001,4); | 1001 | 4 | | | |
| INSERT INTO RATING VALUES (1002,2); | 1002 | 2 | | | |
| INSERT INTO RATING VALUES (1003,5); | 1003 | 5 | | | |
| INSERT INTO RATING VALUES (1004,4); | 1004 | 4 | | | |
| SELECT * FROM ACTOR; | ACT_ID | ACT_NAME | ACT_GENDER | | |
| 301 ANUSHKA SHARMA | 301 | ANUSHKA SHARMA | F | | |
| 302 PRADEEP KUMAR | 302 | PRADEEP KUMAR | M | | |
| 303 PUNEETH RAJAMOULI | 303 | PUNEETH RAJAMOULI | M | | |
| 304 TERMY | 304 | TERMY | M | | |
| SQL> SELECT * FROM DIRECTOR; | DIR_ID | DIR_NAME | DIR_PHONE | | |
| 60 RAJAMOULI | 60 | RAJAMOULI | 8751611001 | | |
| 61 HIT-TECH | 61 | HIT-TECH | 7786138911 | | |
| 62 TEARAN | 62 | TEARAN | 9986446531 | | |
| 63 STEVEN SPIELBERG | 63 | STEVEN SPIELBERG | 8989776530 | | |
| SQL> SELECT * FROM MOVIES; | MOV_ID | MOV_TITLE | MOV_YEAR | MOV_LANG | DIR_ID |
| 1001 BAHDALI | 1001 | BAHDALI | 2012 | BOLLYWOOD | 60 |
| 1002 BAHDALI | 1002 | BAHDALI | 2015 | TELEVISION | 60 |
| 1003 AFASHI | 1003 | AFASHI | 2008 | KANNADA | 61 |
| 1004 WAR HORSE | 1004 | WAR HORSE | 2011 | ENGLISH | 63 |

SELECT * FROM MOVIE_CAST;

| ACT_ID | MOV_ID | ROLE |
|--------|--------|---------------|
| 301 | 1002 | HEROINE |
| 301 | 1001 | HEROINE |
| 303 | 1003 | HERO |
| 303 | 1002 | GUEST |
| 304 | 1004 | HERO, HEROINE |

SELECT * FROM RATINGS;

| MOV_ID | REV_STARS |
|--------|-----------|
| 1001 | 4 |
| 1002 | 8 |
| 1003 | 5 |
| 1004 | 4.5 |
| 1005 | 8 |

Question: List all movies directed by Hitchcock.

Let's list all movies directed by Hitchcock.

SELECT MOV_TITLE, DIRECTOR_ID
FROM MOVIES
WHERE DIRECTOR_ID IN (SELECT DIR_ID
FROM DIRECTORS
WHERE DIR_NAME = 'Hitchcock');

MOV_TITLE -

AnsweR

2. Find the movie name whose one or more actors in two movies.

```

SELECT .MOV_TITLE.
FROM MOVIES M,MOVIE_CAST MC
WHERE M.MOV_ID = MC.MOV_ID AND ACT_ID IN
(SELECT ACT_ID
FROM MOVIE_CAST GROUP BY ACT_ID HAVING
COUNT(FACT_ID) > 1)
GROUP BY MOV_TITLE HAVING
COUNT(*) > 1;

```

MOV_TITLE -

BRAHMA

3. List all actors who acted in movie before 2000 and also in a movie after 2015 (use JOIN operation)

```

SELECT ACT_NAME,MOV_TITLE,MOV_YEAR
FROM ACTOR A JOIN
MOVIE_CAST
ON A.ACT_ID = ACT_ID
JOIN MOVIES M
ON C.MOV_ID = M.MOV_ID
WHERE M.MOV_YEAR < 2000 AND M.MOV_YEAR > 2015;
```

OR.

SELECT :A.ACT_NAME, A.ACT_NAME, C.MOV_TITLE, C.MOV_YEAR
FROM ACTOR A,MOVIE_CAST B, MOVIES C

WHERE A.ACT_ID = B.ACT_ID

AND B.MOV_ID = C.MOV_ID.

AND C.MOV_YEAR NOT BETWEEN 8000 AND 8050;

ACINAME

MOV_TITLE

BAHUBALI-2

4. Find the title of movie and number of stars from each movie that has at least one star and find the highest number of stars that movie received. Sort the result by movie title.

SELECT MOV_TITLE, MAX(REV_STARS) FROM MOVIES

INNER JOIN RATING USING(MOV_ID) GROUP BY MOV_TITLE

HAVING MAX(REV_STARS) > 0 ORDER BY MOV_TITLE

MOV_TITLE

BAHUBALI-1

MOV_TITLE

BAHUBALI-2

AIRASH
4 WAR HORSE
4 WAR HORSE

WINTER'S BANE
WINTER'S BANE

Name of Experiment.....
Experiment No.....

Date.....
Experiment Result.....
Page No. 18

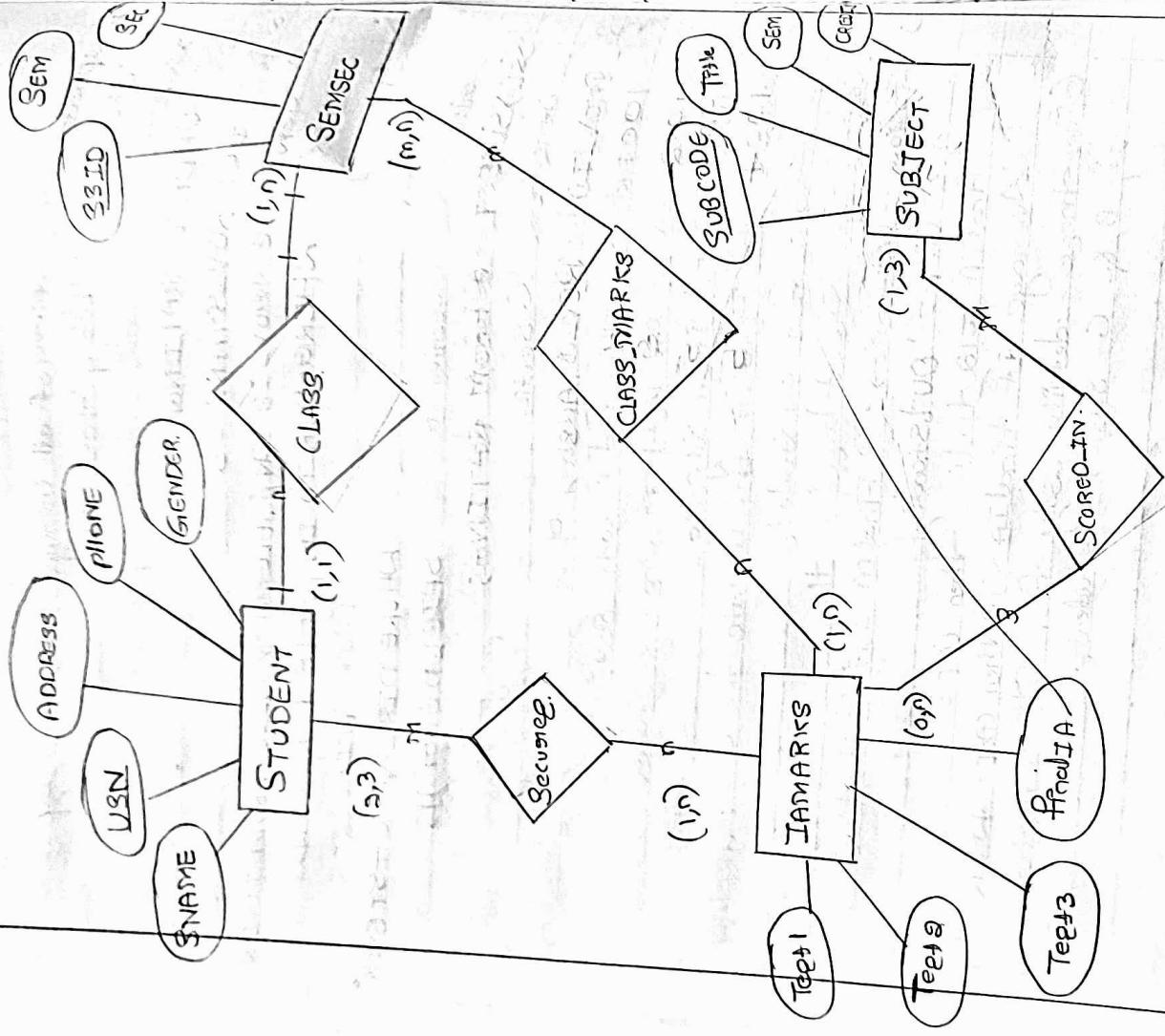
5. Updating rating of all Movie directed by
'Steven Spielberg' to 5.0

```
UPDATE RATING
SET REV_STARS = 5
WHERE MOV_ID IN (SELECT MOV_ID FROM MOVIES
                   WHERE DIR_ID IN (SELECT DIR_ID
                                      FROM DIRECTOR
                                     WHERE DIR_NAME = 'STEVEN
                                         SPIELBERG'));
SELECT * FROM RATING;
```

| MOV_ID | REV_STARS |
|--------|-----------|
| 1001 | 4 |
| 1002 | 8 |
| 1003 | 5 |
| 1004 | 5 |

Q. 13)
Ans:

Entity Relationship Program



Name of Experiment..... Experiment No.D

Date..... Experiment Result.....

Page No. 19

D. Create the Schema for College Database

STUDENT (USN, Name, Address, Phone) (Gender)
SEMESTER (SSID, Sem, SEC)
CLASS (USN SSID)
SUBJECT (Subcode, Title, Sem, Credit)
MARKS (USN, Subject, SSID, Test1, Test2, Test3, FinalTA) ↳

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 '1B11SCE101' in all Subjects.
4. Calculate the Final TA. And update the corresponding table for all students.
5. Category wise Student based on the following conditions : If FinalTA = 17 to 20 then CAT = 'Outstanding'
~~If FinalTA = 12 to 16 then CAT = 'Average'~~
~~If FinalTA < 12 then CAT = 'Needs Improvement'~~
Given three details : Only from 8th Semester
A, B, & C Section Students

Schema Diagram

Student.

| USN | Sname | Address | Phone | Grode |
|-----|-------|---------|-------|-------|
|-----|-------|---------|-------|-------|

Semsec.

| SSID | Sem | Sec |
|------|------|-----|
| USN | SSID | |

Class

| USN | SSID |
|-----|------|
| USN | SSID |

Subject

| Subcode | Title | Sem | Credit |
|---------|-------|-------|--------|
| USN | SSID | Teeth | Teeth |

Tamang.

| USN | Subcode | SSID | Teeth | Teeth | Fmid |
|-----|---------|-------|-------|-------|------|
| USN | SSID | Teeth | Teeth | Teeth | Fmid |

= TA) ~~student~~ student = TFA

CREATE TABLE MARKS

```

USN VARCHAR(10),
SUBCODE VARCHAR(8),
SSID VARCHAR(5),
TEST1 NUMBER(10),
TEST2 NUMBER(10),
TEST3 NUMBER(10),
FINALIA NUMBER(8),
PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE),
FOREIGN KEY (SSID) REFERENCES SEMESTER (SSID));

```

Table Description

DESC STUDENT;

| Name | Length | Type | Constraint |
|---------|--------|----------|-------------|
| USN | 10 | CHAR(10) | Primary Key |
| SNAME | 20 | CHAR(20) | Foreign Key |
| ADDRESS | 20 | CHAR(20) | |
| PHONE | 10 | CHAR(10) | |
| GENDER | 1 | CHAR(1) | |

DESC SEMESTER;

| Name | Length | Type | Constraint |
|--------|--------|---------|-------------|
| SEMID | 5 | CHAR(5) | Primary Key |
| SEMSEC | 5 | CHAR(5) | Foreign Key |

Name of Experiment.....Experiment No.....

Page No. 21

Page No. १

Date..... Experiment Result.....

Name of Experiment..... Experiment No.....

| | |
|------------------------------|--|
| DESC CLASS; | |
| SAL>DESC CLASS; | |
| Name: | |
| USN | |
| SSID | |
| DESC SUBJECT; | |
| SAL>DESC SUBJECT; | |
| Name | |
| SubCode | |
| Title | |
| SGM | |
| Credits | |
| DESC FAMARKS; | |
| SAL>DESC FAMARKS; | |
| Name | |
| USN | |
| SubCode | |
| SSID | |
| TEST1 | |
| TEST2 | |
| FINAL1 | |
| EN | |
| Insertion of Values to Table | |
| EN | |

INSERT INTO STUDENT VALUES ('IRN13CS062', 'SANDHYA',
'BENGALURU', 7728889918, 'F');

INSERT INTO STUDENT VALUES ('IRN13CS091', 'TEESHA',
'BENGALURU', 7712318518, 'F');

INSERT INTO STUDENT VALUES ('IRN13CS066', 'SUPRIYA',
'MANGALURU', 8874561182, 'F');

INSERT INTO STUDENT VALUES ('IRN14CS010', 'ABHAY',
'BENGALURU', 9900811801, 'M');

INSERT INTO STUDENT VALUES ('IRN14CS032', 'BHASKAR',
'BENGALURU', 9923411099, 'M');

INSERT INTO STUDENT VALUES ('IRN14CS085', 'ASMI',
'BENGALURU', 7894737377, 'F');

INSERT INTO STUDENT VALUES ('IRN15CS011', 'AJAY',
'Tumkur', 9845091341, 'M');

INSERT INTO STUDENT VALUES ('IRN15CS089', 'CHITRA',
'DAVANGERE', 7896478181, 'F');

INSERT INTO STUDENT VALUES ('IRN15CS045', 'TEENA',
'BELLARY', 9944850181, 'M');

INSERT INTO STUDENT VALUES ('IRN15CS095', "SANJOSH",
'MANALORE', 8812338801, 'M');

INSERT INTO STUDENT VALUES ('IRN16CS045', 'ISMAIL',
'KALABURGEE', 990023801, 'M');

INSERT INTO STUDENT VALUES ('IRN16CS088', 'SAMEEKA',
'SHIMOGA', 9905548818, 'F');

INSERT INTO STUDENT VALUES ('IRN16CS088', 'VINA',
'CHRIS AMATRANIA', 8800880011, 'M');

```

INSERT INTO SEMSEC VALUES ('CSEBA', 8, 'A');
INSERT INTO SEMSEC VALUES ('CSECB', 8, 'B');
INSERT INTO SEMSEC VALUES ('CSECA', 8, 'C');
INSERT INTO SEMSEC VALUES ('CSEFA', 7, 'A');
INSERT INTO SEMSEC VALUES ('CSEFB', 7, 'B');
INSERT INTO SEMSEC VALUES ('CSEFC', 7, 'C');
INSERT INTO SEMSEC VALUES ('CSEGA', 6, 'A');
INSERT INTO SEMSEC VALUES ('CSEGBC', 6, 'B');
INSERT INTO SEMSEC VALUES ('CSEGAC', 6, 'C');
INSERT INTO SEMSEC VALUES ('CSEGAI', 5, 'A');
INSERT INTO SEMSEC VALUES ('CSEGBI', 5, 'B');
INSERT INTO SEMSEC VALUES ('CSEGCI', 5, 'C');

```

(CSEGC)

```

INSERT INTO SEMSEC VALUES ('CSEGDA', 4, 'A');
INSERT INTO SEMSEC VALUES ('CSEGDB', 4, 'B');
INSERT INTO SEMSEC VALUES ('CSEGAC', 4, 'C');
INSERT INTO SEMSEC VALUES ('CSEGDA', 3, 'A');
INSERT INTO SEMSEC VALUES ('CSEGDB', 3, 'B');
INSERT INTO SEMSEC VALUES ('CSEGDC', 3, 'C');
INSERT INTO SEMSEC VALUES ('CSEGAA', 2, 'A');
INSERT INTO SEMSEC VALUES ('CSEGAB', 2, 'B');
INSERT INTO SEMSEC VALUES ('CSEGAC', 2, 'C');

```

(IRNIBCS091, CSEGC)

```

INSERT INTO CLASS VALUES ('IRNIBCS091', 'CSEGC');
INSERT INTO CLASS VALUES ('IRNIBCS092', 'CSEGA');
INSERT INTO CLASS VALUES ('IRNIBCS093', 'CSEGB');

```

(CSEGC)

~~INSERT INTO CLASS VALUES ('IRN14C8010', 'CEGA');~~
~~INSERT INTO CLASS VALUES ('IRN14C8005', 'CEGA');~~
INSERT INTO CLASS VALUES ('IRN15C8011', 'CEGA');
INSERT INTO CLASS VALUES ('IRN15C8009', 'CEGA');
INSERT INTO CLASS VALUES ('IRN16C8004', 'CEGA');
INSERT INTO CLASS VALUES ('IRN16C8001', 'CEGA');

~~INSERT INTO CLASS VALUES ('IRN16C8045', 'CEGA');~~
~~INSERT INTO CLASS VALUES ('IRN16C8088', 'CEGA');~~
INSERT INTO CLASS VALUES ('IRN16C8000', 'CEGA');

~~INSERT INTO SUBJECT VALUES ('10CS81', 'ACF', 8, 4);~~
~~INSERT INTO SUBJECT VALUES ('10CS80', 'SSM', 8, 4);~~
INSERT INTO SUBJECT VALUES ('10CS83', 'NMI', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS84', 'UC', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS85', 'PN', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS84', 'OAO', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS82', 'EGT', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS83', 'PTW', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS84', 'DNDM', 7, 4);

~~INSERT INTO SUBJECT VALUES ('10CS75', 'JAVA', 7, 4);~~
~~INSERT INTO SUBJECT VALUES ('10CS76', 'SAN', 7, 4);~~
INSERT INTO SUBJECT VALUES ('15CS51', 'ME', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS50', 'CN', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS53', 'DBMS', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS54', 'ATC', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS55', 'TVA', 5, 4);

```

INSERT INTO SUBJECT VALUES ('SCS36', 'AI', 5, 3);
INSERT INTO SUBJECT VALUES ('SCS41', 'DA', 4, 4);
INSERT INTO SUBJECT VALUES ('SCS42', 'SE', 4, 4);
INSERT INTO SUBJECT VALUES ('SCS43', 'DB', 4, 4);
INSERT INTO SUBJECT VALUES ('SCS44', 'MPMC', 4, 4);
INSERT INTO SUBJECT VALUES ('SCS45', 'OBC', 4, 3);
INSERT INTO SUBJECT VALUES ('SCS46', 'DC', 4, 3);
INSERT INTO SUBJECT VALUES ('SCS31', 'M3', 3, 4);
INSERT INTO SUBJECT VALUES ('SCS32', 'ADE', 3, 4);
INSERT INTO SUBJECT VALUES ('SCS33', 'DSA', 3, 4);
INSERT INTO SUBJECT VALUES ('SCS34', 'OS', 3, 4);
INSERT INTO SUBJECT VALUES ('SCS35', 'VSP', 3, 3);
INSERT INTO SUBJECT VALUES ('SCS36', 'DMS', 3, 3);

INSERT INTO MARKS (USN, SUBCODE, SSID, TEST1, TEST2,
TEST3) VALUES ('IRN13CS091', 'KSE8C', 15, 16, 18);
INSERT INTO MARKS (USN, SUBCODE, SSID, TEST1,
TEST2, TEST3) VALUES ('IRN13CS091', 'KSE8C', 10, 19, 14);
INSERT INTO MARKS (USN, SUBCODE, SSID, TEST1, TEST2,
TEST3) VALUES ('IRN13CS091', '10CS83', 'KSE8C', 19, 15, 19);
INSERT INTO MARKS (USN, SUBCODE, SSID, TEST1, TEST2,
TEST3) VALUES ('IRN13CS091', '10CS84', 'KSE8C', 20, 16, 19);
INSERT INTO MARKS (USN, SUBCODE, SSID, TEST1, TEST2,
TEST3) VALUES ('IRN13CS091', '10CS85', 'KSE8C', 15, 15, 19);

SELECT * FROM STUDENT;
SQL SELECT * FROM STUDENT;

```

| UIN | S Name | Address | Phone |
|--|-----------|-------------|---------------|
| IRN13CS080 | AKSHAY | BELAGAVI | 8844881982 |
| IRN13CS062 | SANDHYA | BENGALURU | 77888889912 F |
| IRN13CS091 | TEEGHA | BENGALURU | 4412312312 F |
| IRN13CS066 | SUPRIYA | MANGALURU | 8844881982 |
| IRN14CS010 | ABHAY | MANGALURU | 9900811801 F |
| IRN14CS032 | BHASKAR | BENGALURU | 9983811099 M |
| IRN15CS041 | ASTAY | TUMKUR | 9845091341 M |
| IRN15CS029 | CHITRA | DAVANAHALLI | 7696448181 F |
| IRN15CS045 | JEGGA | BELLARY | 9944650101 M |
| IRN15CS091 | SANTOSH | MANGALURU | 8818332801 M |
| IRN16CS045 | ISMAIL | KALABURGI | 9900838201 M |
| IRN16CS088 | SAMI GERA | SHIMOGGA | 9905548812 F |
| IRN16CS188 | VINAYAKA | CHIKmagalur | 8800880011 M |
| IRN16CS045 | ASMI | BANGALURU | 7894434344 F |
| SELECT * FROM SEMSEC; <i>(Entire page)</i> | | | |
| SQL SELECT * FROM SEMSEC; <i>(Entire page)</i> | | | |
| SSTID | SSEM | SSEM | SSEM |
| CSE8AT | 8122 | 8122 | 8122 |
| CSE8B | 8122 | 8122 | 8122 |
| CSE8C | 8122 | 8122 | 8122 |
| CSE7A | 7202 | 7202 | 7202 |
| CSE7B | 7202 | 7202 | 7202 |
| CSE7C | 7202 | 7202 | 7202 |
| CSE6A | 6 | 6 | 6 |
| CSE6B | B | B | B |
| CSE6C | 6 | C | C |
| CSE5A | A | A | A |
| CSE5B | 5 | C | C |
| CSE5C | 5 | C | C |

Name of Experiment.....

Date.....

Page No. **24**

Experiment No.....

Experiment Result.....

| | | | | |
|---------------------------|------|-------|---|------|
| CSE 4A | 4 | A | 1 | 100% |
| CSE 4B | 4 | B | 1 | 100% |
| CSE 4C | 4 | C | 1 | 100% |
| CSE 3A | 3 | A | 1 | 100% |
| CSE 3B | 3 | B | 1 | 100% |
| CSE 3C | 3 | C | 1 | 100% |
| CSE 2A | 2 | A | 1 | 100% |
| CSE 2B | 2 | B | 1 | 100% |
| CSE 2C | 2 | C | 1 | 100% |
| CSE 1A | 1 | A | 1 | 100% |
| CSE 1B | 1 | B | 1 | 100% |
| CSE 1C | 1 | C | 1 | 100% |
| SELECT * FROM CLASS; | | | | |
| SQL>SELECT * FROM .CLASS; | | | | |
| USN + SSID | 2160 | 42921 | | |
| IRN13CS060 - CSE8B | 76 | 2160 | | |
| IRN13CS062 - CSE8A | 76 | 2160 | | |
| IRN13CS066 - CSE8B | 76 | 2160 | | |
| IRN13CS091 CSE 8C | 76 | 2160 | | |
| IRN14CS010 CSE10 | 76 | 2160 | | |
| IRN14CS025 CSE7A | 76 | 2160 | | |
| IRN14CS039 CSE7A | 76 | 2160 | | |
| IRN15CS001 CSE4A | 76 | 208 | | |
| IRN15CS029 CSE4A | 76 | 208 | | |
| IRN15CS045 CSE4B | 76 | 208 | | |
| IRN15CS091 CSE4C | 76 | 144 | | |
| IRN15CS045 CSE3A | 76 | 144 | | |
| IRN16CS088 CSE3B | 76 | 84 | | |
| IRN16CS198 CSE3C | 76 | 84 | | |

14 course Selected.

SELECT & FROM SUBJECT;

| SUBCODE | TITLE | SEM | CREDITS |
|---------|-------|-----|---------|
| 10CS81 | ACA | 8 | 4 |
| 10CS82 | SSM | 8 | 4 |
| 10CS83 | NM | 8 | 4 |
| 10CS84 | CC | 8 | 4 |
| 10CS85 | PAW | 8 | 4 |
| 10CS41 | DOAD | 7 | 4 |
| 10CS42 | ECS | 7 | 4 |
| 10CS43 | PTA | 7 | 4 |
| 10CS44 | DINOM | 7 | 4 |
| 10CS45 | JAVA | 7 | 4 |
| 10CS46 | SAN | 7 | 4 |
| 15CS51 | ME. | 5 | 4 |
| 15CS52 | CN | 5 | 4 |
| 15CS53 | DBMS | 5 | 4 |
| 15CS54 | ATC | 5 | 4 |
| 15CS55 | JAVA | 5 | 4 |
| 15CS56 | AI | 5 | 4 |
| 15CS41 | ML4 | 4 | 3 |
| 15CS42 | SE | 4 | 3 |
| 15CS43 | DATA | 4 | 3 |
| 15CS44 | PPM | 4 | 3 |
| 15CS45 | PPMC | 4 | 3 |
| 15CS46 | DOC | 4 | 3 |
| 15CS31 | DC | 3 | 3 |
| 15CS32 | MIS | 3 | 3 |
| 15CS33 | ADA | 3 | 3 |
| 15CS34 | DSA | 3 | 3 |
| 15CS35 | CO | 3 | 3 |
| 15CS36 | UFS | 3 | 3 |

3 Seminal

CS320

3

```

SELECT & FROM TADARIES
SQL> SELECT & FROM TADARIES;
      USN    SUBCODE    SSID    TEST1    TEST2    TEST3    MEDALIA
1RN13CS091 10CS81  CSE8C   15     16     17     18
1RN13CS091 10CS82  CSE8C   12     19     14     14
1RN13CS091 10CS83  CSE8C   19     15     16     16
1RN13CS091 10CS84  CSE8C   20     16     17     19
1RN13CS091 10CS85  CSE8C   15     15     16     12.

```

Question:-

1 List all the Student details Studying in Fourth Semester 'C' Section.

```

SELECT S.& SS.SEM  SS.SEC
FROM STUDENTS S, SEMESTERS, CLASS C
WHERE S.USN = C.USN AND
S.SSID = C.SSID AND
SS.SEM = 4 AND
SS.SEC = 4

```

```

USN  SNAME  ADDRESS  PHNO  SS.SEM
1RN13CS091  SANTOSH  MANGALURU  8818332201  04

```

2. Compute The Total Number of male and female student in each Semester and in each Section

```

SELECT SS.SEM, SS.SEC, S.GENDER, COUNT (S.GENDER)
AS . COUNT FROM
STUDENT S, SEMESTERS, CLASS C
WHERE USN = C.USN AND

```

SS-SID = C-SID.

GROUP BY SS-SEM , SS-SEC, S, GENDER

ORDER BY SEM ;

| SEM | S | G | COUNT |
|-----|---|---|-------|
| 3 | A | M | 1 |
| 3 | B | F | 1 |
| 3 | C | M | 1 |
| 4 | A | F | 1 |
| 4 | A | M | 1 |
| 4 | B | M | 1 |
| 4 | C | F | 1 |
| 7 | A | M | 9 |
| 8 | A | F | 1 |
| 8 | A | M | 1 |
| 8 | B | F | 1 |
| 8 | C | F | 1 |

3. Create a view of Test marks for all subjects
 'BTECS101'. In all subjects mark is same as
 CREATE VIEW STU-TEST1 - MARKS-MARKS AS
 SELECT TEST1, SUBCODE
 FROM 'TAKERS'
 WHEREG-USN = 'JRN13C09';

| TEST1 | SUBCODE | USN | MARKS |
|-------|---------|----------|-------|
| 15 | 10CS81 | JRN13C09 | 92 |
| 16 | 10CS82 | JRN13C09 | 92 |
| 19 | 10CS83 | JRN13C09 | 92 |
| 20 | 10CS84 | JRN13C09 | 92 |
| 15 | 10CS85 | JRN13C09 | 92 |

4 Calculate The Final TA (average of best two tests marks) and update the corresponding table for all students.

CREATE OR REPLACE PROCEDURE ALGMARIE IS

CURSOR_TAMARIS IS

SELECT GREATEST(TEST1,TEST2) AS A,GREATEST

(TEST1,TEST3) AS B,

GREATEST(TEST3,TEST2) AS C

FROM TAMARIS

WHERE FINITA IS NULL

FOR UPDATE;

C-ANUMBER; F1 08322 E8391 PROG LANG

C-BNUMBER; F2 08322 E8391 PROG LANG

C-CNUMBER; F3 08322 E8391 PROG LANG

C-SNUMBER;

BEGIN

OPEN C-TAMARKS;

LOOP

IF C-TAMARKS INTO C-A, C-B, C-C;

EXIT WHEN C-TAMARKS := NOT FOUND;

-- DBMS_OUTPUT.PUT_LINE ('C-A'||'||'||C-B||'||'

C-C);

IF C-S01 := C-A+C-B;

ELSIF

C-S01 := C-A+C-B;

END IF;

```

C_AV := C_SMI / 2;
-- DBMS_OUTPUT.PUT_LINE('sum = '||C_SMI);
-- DBMS_OUTPUT.PUT_LINE('AVE RAJE = '||C_AV);
UPDATE TAMARIS SET FINALIA = C_AV * NHGRG;
CURRENT OF C_TAMARKS;

```

END LOOP;

CLOSE C_TAMARKS; V_CLOSE(TAMARIS);
SELECT COUNT(*) FROM TAMARKS;
DBMS_OUTPUT.PUT_LINE(COUNT);
END;

| USN | SUBCODE | SSID | TEST1 | TEST2 | TEST3 | TEST4 |
|------------|---------|-------|-------|-------|-------|-------|
| IRN13CS091 | 10CS81 | CSE8C | 15 | 16 | 17 | 17 |
| IRNB13S091 | 10CS82 | CSE8C | 18 | 19 | 17 | 17 |
| IRN13CS091 | 10CS83 | CSE8C | 19 | 18 | 19 | 19 |
| IRN13CS091 | 10CS84 | CSE8C | 20 | 18 | 19 | 19 |
| IRN13CS091 | 10CS85 | CSE8C | 15 | 15 | 15 | 15 |

Category Student based on the following criterion: IF FinalIA = 17 & otherCAT = 'Outstanding',
 If FinalIA = 18 to 16 Then CAT = 'Average', If FinalIA < 16 Then CAT = 'Fail'.
 'Weak' will be output if FinalIA = 20 to 25.

Give three desirable output for section A, B, AC
 section Student.

Output : NO2.C

File

Output : NO2.C

Name of Experiment.....

Date.....

Experiment Result.....

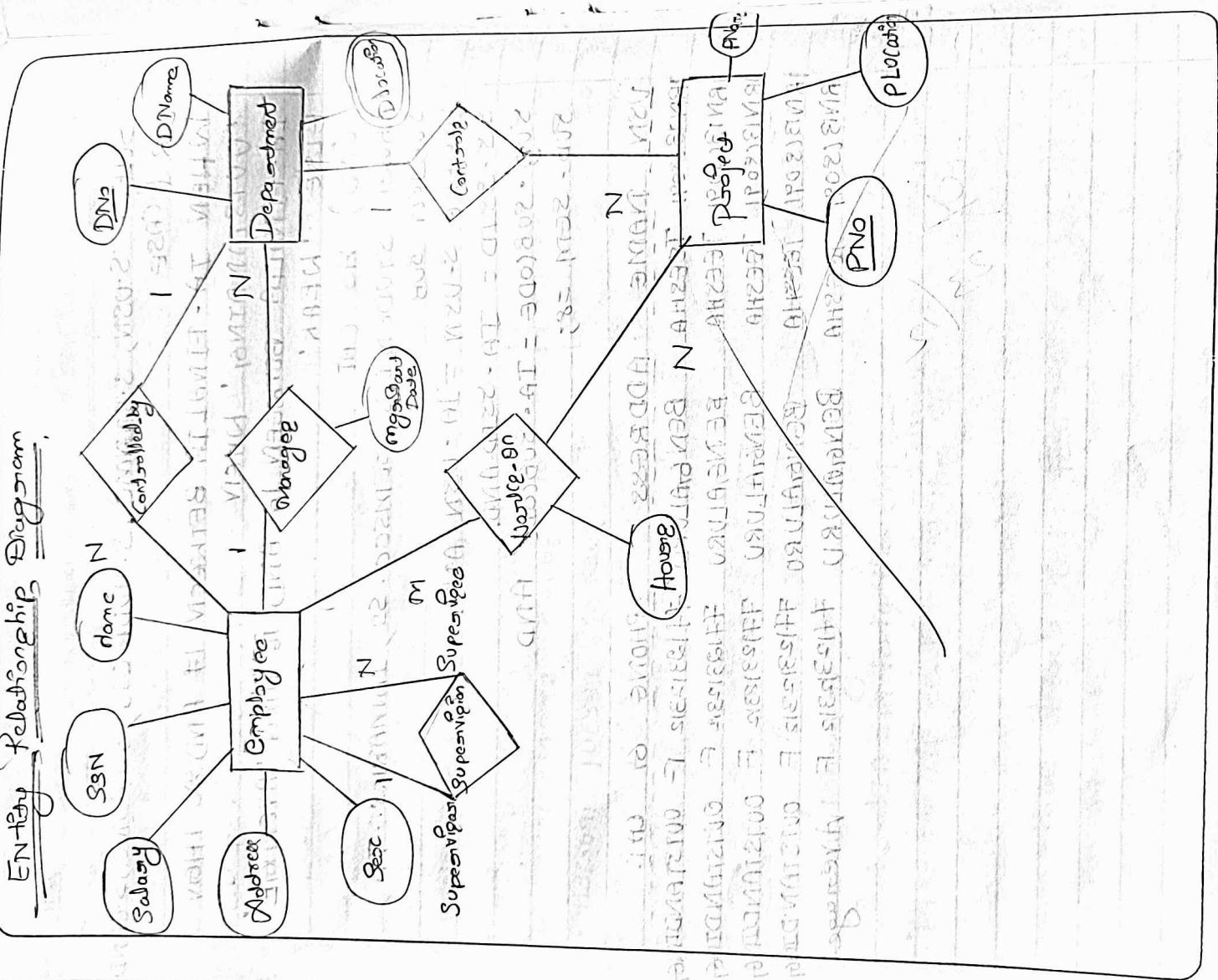
Page No. 87

```
SELECT S.USN, S.SNAME, S.ADDRESS, S.PHONE, S.GEND
      ER (CASE
      WHEN TA - FINALIA BETWEEN 17 AND 20 THEN
        'OUTSTANDING'
      WHEN TA - FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
      ELSE 'WEAK'
      END) AS CAT
  FROM STUDENT S, SEMESTER SS, IITMARIKS . TA,
  SUBJECT SUB
 WHERE S.USN = TA . USN AND
 SS . STD = TA . STD AND
 SUB . SUBCODE = TA . SUBCODE AND
 SUB . SEMESTER = 2;
```

| USN | NAME | ADDRESS | PHONE | CAT |
|-----------|--------|-----------|------------|-----------------|
| IRN135091 | TEESHA | BENGALURU | 9812312312 | F - OUTSTANDING |
| IRN135091 | TEESHA | BENGALURU | 9812312312 | F - OUTSTANDING |
| IRN135091 | TEESHA | BENGALURU | 7712312312 | F - OUTSTANDING |
| IRN135091 | TEESHA | BENGALURU | 9812312312 | F - OUTSTANDING |
| IRN135091 | TEESHA | BENGALURU | 7412312312 | F - AVERAGE |

✓ ✓ ✓ ✓ ✓ ✓

Entity Relationship Diagram



E) Concepts - The Schema From Company Database:

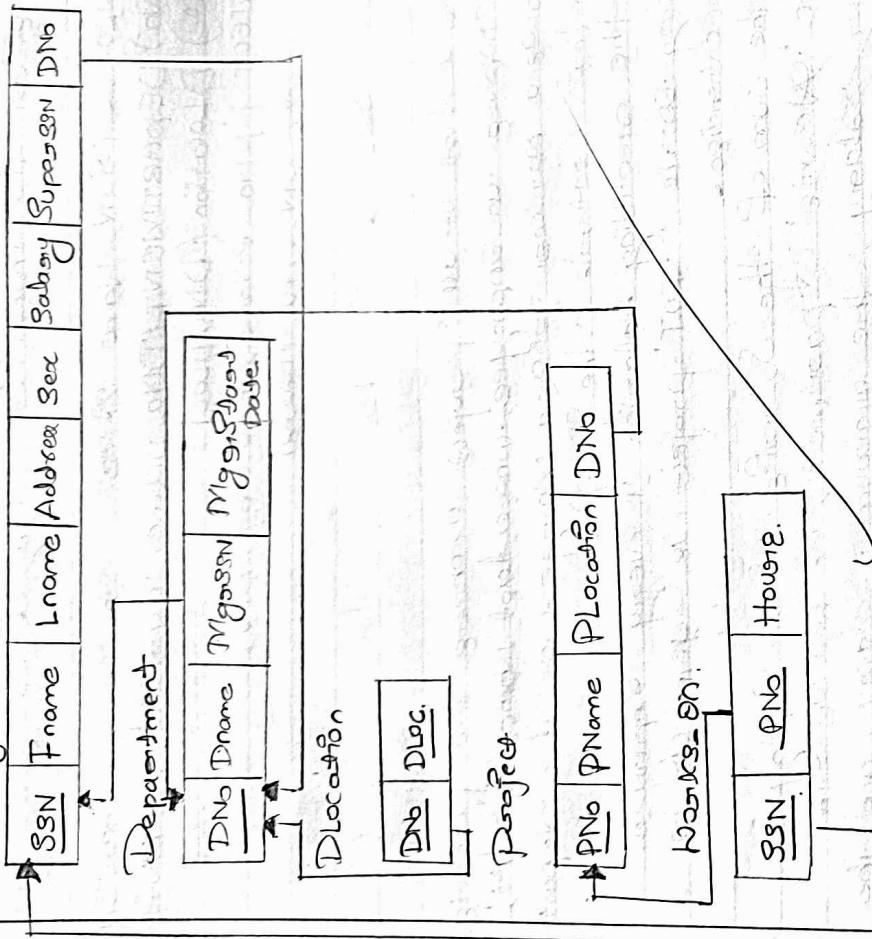
EMPLOYEE (SSN, Name, Address, Sex, Salary, Supervisor
 DNo) DEPARTMENT (DNo, DName, Manager, Project)
 DEPT LOCATION (DNo, DLoc)
 PROJECT (PNo, PName, PLocNo, DNo)
 WORKS_ON (SSN, PNo, Hours)

Write SQL Querying To.

- 1) Print a list of all Projects number for Project that involve an Employee whose last name is Scott either as a member or as Manager of the department that contains the Project.
- 2) Show the executing balance of every employee working in the 'TEST' Project is given as 10 percent of his/her current balance.
- 3) Find the sum of the Salaries of all Employee of the 'Accounts' Department, as well as the maximum Salary, the minimum Salary, and the average Salary in the department.
- 4) Retrieve the name of each Employee who has the smallest Project controlled by department number 5 (i.e., not Employee of department 5) from each department that has more than five Employees, otherwise the department number and the number of the Employee who does not make more than \$6,000.

Schema Diagram

Employee.



Name of Experiment..... Date.....
Experiment No..... Experiment Result.....

Page No. 29

Table Description

```
CREATE TABLE DEPARTMENT  
(DNo VARCHAR(20) PRIMARY KEY,  
DName VARCHAR(20), REGISTRATION  
DATE);
```

```
CREATE TABLE EMPLOYEE  
(SSN VARCHAR(20) PRIMARY KEY,  
ENAME VARCHAR(20),  
LNAME VARCHAR(20),  
ADDRESS VARCHAR(20),  
SEX CHAR(1), SALARY  
INTEGER,  
SUPERSSN REFERENCES EMPLOYEE(SSN),  
DNo REFERENCES DEPARTMENT(DNo);
```

NOTE: Once Department and Employee Table are
Created we Must Create department Table to add
Foreign Constraint SSN Using SQL Command

```
ALTER TABLE DEPARTMENT  
ADD SSN REFERENCES EMPLOYEE(SSN);
```

```
CREATE TABLE DISCOTTION  
(DLoc VARCHAR(20)),  
DNo REFERENCES DEPARTMENT(DNo),  
PRIMARY KEY (DNo, DLoc);
```

CREATE TABLE PROJECT (PNO

```

    INTEGER PRIMARYKEY , PNAME
    VARCHAR(80),
    LOCATION VARCHAR(80),
    DNo REFERENCES DEPARTMENT(DNo));

```

CREATE TABBLE WORKS ON

(HOURS. NUMBER (8))

SSN REFERENCES . EMPLOYEE (SSN)

PNO REFERENCES PROJECT (PNo),

PRIMARY KEY (SSN, PNo);

Table Description

DESC. - EMPLOYEE;

SOL > DESS EMPLOYEE:

Name:

837

FNAME

LITERATURE

三

ADDRESS

568

SALEBY

二〇〇〇年

308258

Name of Experiment..... Experiment No.....

Date..... Experiment Result.....

Page No. **30**

| DESC DEPARTMENT; | |
|------------------|------------------|
| SQL > | DESC DEPARTMENT; |
| Name | - - - - - |
| PNo | - - - - - |
| DNAME | |
| SQL > | DNAME; |
| INSTRSTARTDATE | - - - - - |
| INSTRENDDATE | - - - - - |
| DESC LOCATION; | |
| SQL > | DESC LOCATION; |
| Name | - - - - - |
| Loc | - - - - - |
| PNo | - - - - - |
| DESC PROTECT; | |
| SQL > | DESC PROTECT; |
| Name | - - - - - |
| PNo | - - - - - |
| PINADAE | - - - - - |
| LOCATION | - - - - - |
| PNo | - - - - - |
| DESC WORKS_ON; | |
| SQL > | DESC WORKS_ON; |
| Name | - - - - - |
| Hours | - - - - - |
| SEN | - - - - - |
| PNo | - - - - - |

Insertion of values to table.

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC01', 'JOHN', 'SCOTT', 'BANGA-
'M', 45000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC01', 'TAMMY', 'SMITH',
'F', 'BANGALORE', 'M', 50000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC02', 'HERAN', 'BAKER', 'BANGA-
'M', 70000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC03', 'EDWARD', 'SCOTT',
'M', 'MYSORE', 'M', 50000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC04', 'PAUL', 'HENDER-
'M', 'MANGALORE', 'M', 65000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC05', 'GIRISH', 'MALYA', 'MYSORE',
'M', 45000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC06', 'NEHA', 'SN', 'BANGALORE',
'F', 80000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC01', 'ATHARIA', 'K', 'MANGALORE',
'F', 35000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS,
SEX, SALARY) VALUES ('RNSEC02', 'ANTHOSH', 'KUMAR', 'MANGALORE',
'M', 40000);

Name of
Expert

| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Name of Experiment..... Experiment No.....

Date..... Experiment Result.....

Page No. 31

```
'01', '30 0000);  
INSERT INTO 'EMPLOYEE' (SSN, FNAME, LNAME, ADDRESS,  
SEX, SALARY) VALUES ('RNS1SE01', 'KEENA', 'M', 'MYSORE',  
'M', 60000);  
INSERT INTO 'EMPLOYEE' (SSN, FNAME, ADDRESS,  
SEX, SALARY) VALUES ('RNS1SE01', 'NAGESH', 'HR', 'BANGALORE',  
'M', 500000);  
  
INSERT INTO 'DEPARTMENT' VALUES ('1', 'ACCOUNT', '01-JAN  
-01', 'RNSACCO01');  
INSERT INTO 'DEPARTMENT' VALUES ('2', 'IT', '01-AUG-  
'01', 'RNSTIT01');  
INSERT INTO 'DEPARTMENT' VALUES ('3', 'ECE', '01-JUN-  
'01', 'RNSECE01');  
INSERT INTO 'DEPARTMENT' VALUES ('4', 'ISE', '01-Aug-  
'05', 'RNIS01');  
INSERT INTO 'DEPARTMENT' VALUES ('5', 'CSE', '01-JUN-  
'08', 'RNCS05');  
  
Note: Update Employee table to fill  
missing fields: SUPERSSN and DNO.  
  
UPDATE 'EMPLOYEE' SET  
SUPERSSN = NULL, DNO = 3  
WHERE SSN = 'RNS1SE01'  
  
UPDATE 'EMPLOYEE' SET  
SUPERSSN = 'RNSCS05', DNO = 5  
WHERE SSN = 'RNS1SE01';
```

UPDATE EMPLOYEE SET
SUPERSSN = 'RNSCGE03', DNo = '5'
WHERE SSN = 'RNCSCE00';

UPDATE EMPLOYEE SET
SUPERSSN = 'RNCSCE04', DNo = '5'
WHERE SSN = 'RNCSCE03';

UPDATE EMPLOYEE SET DNo = '5',
SUPERSSN = 'RNCSCE05',
HIERESSN = 'RNCSCE04';

UPDATE EMPLOYEE SET DNo = '5',
SUPERSSN = 'RNCSCE06',
HIERESSN = 'RNCSCE05';

UPDATE EMPLOYEE SET :
DNo = '5', SUPERSSN = NULL
HIERESSN = 'RNCSCE06';

UPDATE EMPLOYEE SET DNo = '
SUPERSSN = 'RNCSAC00'.
HIERESSN = 'RNCSAC01';

UPDATE EMPLOYEE SET
DNo = '1', SUPERSSN = NULL
WHERE SSN = 'RNCSAC02';

UPDATE EMPLOYEE SET
DNo = '4', SUPERSSN = NULL
HIERESSN = 'RNCSAC01';

Name of Experiment.....

Date..... Experiment Result.....

Experiment No.....

Page No 32

```
UPDATE EMPLOYEE SET  
DNO = '2', SUPERSEN = NULL  
WHERE SSN = 'RNSTT01';  
  
INSERT INTO LOCATION VALUES ('BANGALORE', '1');  
INSERT INTO LOCATION VALUES ('BANGALORE', '2');  
INSERT INTO LOCATION VALUES ('BANGALORE', '3');  
INSERT INTO LOCATION VALUES ('MANGALORE', '4');  
INSERT INTO LOCATION VALUES ('MANGALORE', '5');  
  
INSERT INTO PROJECT VALUES (100, 'IOT', 'BANGALORE',  
'5');  
INSERT INTO PROJECT VALUES (101, 'CLOUD', 'BANGALORE',  
'5');  
INSERT INTO PROJECT VALUES (102, 'BIGDATA', 'BANGALORE',  
'5');  
INSERT INTO PROJECT VALUES (103, 'SENSORS', 'BANGALORE',  
'3');  
INSERT INTO PROJECT VALUES (104, 'BANUS MANAGEMENT',  
'BANGALORE', '1');  
INSERT INTO PROJECT VALUES (105, 'SALARAY MANAGEMENT',  
'BANGALORE', '1');  
INSERT INTO PROJECT VALUES (106, 'OPENSTACK', 'BANGALORE',  
'4');  
INSERT INTO PROJECT VALUES (107, 'SMARTCITY',  
'BANGALORE', '2');  
  
GOALS:  
1. To Implement the Project Management System  
2. To Implement the Sensor Management System  
3. To Implement the Cloud Computing System  
4. To Implement the Big Data System  
5. To Implement the IOT System  
6. To Implement the Salary Management System  
7. To Implement the OpenStack System  
8. To Implement the Smart City System
```

```

INSERT INTO WORKS_ON VALUES (5, 'RNSCSE06', 100);
INSERT INTO WORKS_ON VALUES (6, 'RNSCSE03', 100);
INSERT INTO WORKS_ON VALUES (7, 'RNSCSE01', 100);
INSERT INTO WORKS_ON VALUES (5, 'RNSACCO1', 100);
INSERT INTO WORKS_ON VALUES (6, 'RNSACCO1', 100);
INSERT INTO WORKS_ON VALUES (4, 'RNSIT01', 100);
INSERT INTO WORKS_ON VALUES (10, 'RNSIT01', 100);

```

SELECT * FROM EMPLOYEE;

| SSN | FName | LName | ADDRESS | SALARY |
|----------|---------|--------|-------------|--------|
| RNSCSE01 | JOHN | SCOTT | BANGALORE M | 450000 |
| RNSCSE01 | JAMES | SMITH | BANGALORE M | 500000 |
| RNSCSE02 | FEARN | BAKER | BANGALORE M | 400000 |
| RNSCSE03 | EDWARD | SCOTT | MYSORE M | 500000 |
| RNSCSE04 | PAVAN | HEADER | MANGALORE M | 650000 |
| RNSCSE05 | GIRISH | MALYA | MYSORE M | 450000 |
| RNSCSE06 | NEHA | SONI | BANGALORE F | 800000 |
| RNSACCO1 | ATHANA | KUMAR | MANGALORE F | 350000 |
| RNSIT01 | SANTOSH | KUMAR | MANGALORE M | 300000 |
| RNSIT01 | VEENA | M | MYSORE M | 600000 |
| RNSIT01 | NAGESH | HR. | BANGALORE M | 500000 |

SELECT * FROM DEPARTMENT;

SQL> SELECT * FROM DEPARTMENT;

| DNo. | DNAME | INSTRSTARTDATE | INSTRENDDATE |
|------|----------|----------------|--------------|
| 1 | ACCOUNTS | 01-JAN-01 | RNS ACCO2 |
| 2 | IT | 01-Aug-16 | RNS IT01 |

Name of Experiment.....
Experiment No.....

Date.....
Experiment Result.....
Page No 33

| | | | |
|------------------------------------|-------------------|-----------|---------|
| 3 | EE | 01-JUN-08 | RNSCE01 |
| 4 | TSE | 01-AUG-15 | RNSCE01 |
| 5 | CSE | 01-JUN-08 | RNSCE05 |
| SELECT & FROM LOCATION; | | | |
| 100 | DNo | 1 | 1 |
| | BANGALORE | | |
| | BANGALORE | 2 | |
| | BANGALORE | 3 | |
| | MANGALORE | 4 | |
| | MANGALORE | 5 | |
| SELECT & FROM PROJECT; | | | |
| 100 | PNo | 1 | 1 |
| | PNAME | BANGALORE | |
| 101 | TDI | 5 | |
| | Cloud | BANGALORE | 5 |
| 102 | BIOPDATA | BANGALORE | 5 |
| 103 | SENSORS | BANGALORE | 3 |
| 104 | BANK MANAGEMENT | BANGALORE | 1 |
| 105 | SALARY MANAGEMENT | BANGALORE | 1 |
| 106 | OPENSTACK | BANGALORE | 4 |
| 107 | SIMNET CITY | BANGALORE | 2 |
| SELECT & FROM WORKS-ON; | | | |
| 108 | HOURS | SSN | PNo |
| | | RNSCE01 | 100 |
| 6 | RNSCE01 | 100 | 101 |
| 8 | RNSCE01 | 102 | 102 |
| 10 | RNSCE08 | 100 | 100 |

| | | |
|----|-----------|-----|
| 3 | RNSCS8E04 | 100 |
| 4 | RNSCS8E05 | 101 |
| 5 | RNSCS8E06 | 102 |
| 6 | RNSCS8E03 | 102 |
| 7 | RNSCS8E01 | 103 |
| 5 | RNSACCO1 | 104 |
| 6 | RNSACCO2 | 105 |
| 4 | RNSTSE01 | 106 |
| 10 | RNSIT01 | 107 |

Questions

- =====

 1. Make a list of all project numbers from Project that involve an employee whose last name is Scott, either as a monitor or as a manager of the department that handles the Project.

(SELECT DISTINCT #P, NO)

FROM PROJECT P, DEPARTMENT D, EMPLOYEE E

AND D-MGRSSN=E-SSN

AND E. WADDELL SCOTT
UNION

(S)ELECT DIRECTIVE: 01/08

~~FROM PROJECT PI WORKS-ON N, EMPLOYES E1~~

HERE $P_1 \cdot P_{NO} \neq H \cdot P_{NO}$

AND ~~El. 88N = 10-88 N~~ ~~El. 88N = 10-88 N~~

AND E1.LNAME = 'SCOTT');

Name of Experiment.....
Experiment No.....

Date.....
Experiment Result.....
Page No. 37

| PNo | 100 |
|-----|-----|
| | 101 |
| | 102 |
| | 103 |
| | 104 |
| | 105 |
| | 106 |
| | 107 |

5] Show the Resulting Balance of Every Employee Presenting on the Total Perfect ie Given also Percentage.

SELECT E.ENAME, E.INAME, E.SALARY AS

TNCR_SAL FROM

EMPLOYEE E, PROJCT P

WHERE E.SSN = P.SSN

AND H.PNO = P.PNO

AND P.PNAME = 'TOT'

| ENAME | INAME | TNCR_SAL |
|-------|-------|----------|
| JAMES | SMITH | 55000 |
| HELEN | BAKER | 74000 |
| PAVAN | HEGDE | 71500 |

3] Find the Sum of all Employee of the Account Department as well as the maximum Salary, the minimum Salary, and the average Salary is this.

department.

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

| sum(E.SALARY) | MAX(E.SALARY) | MIN(E.SALARY) | Avg(E.SALARY) |
|---------------|---------------|---------------|---------------|
| 650000 | 350000 | 300000 | 305000 |

4] Retrieve the name of each Employee who works
in the Project & contained by department number S
(See NOT EXISTS operator).

```
SELECT E.FNAME, E.LNAME
  FROM EMPLOYEE E, DEPARTMENT D
 WHERE NOT EXISTS (SELECT PNO FROM PROJECT
      WHERE DNO = 'S')
  MINUS (SELECT PNO
  FROM WORKS
 WHERE E.SSN = PSSN);
```

| F NAME | L NAME |
|--------|--------|
| JAMES | SMITH |

5] For each Department that has more than five
Employee, retrieves the department number and
number of the employee who have working more than
RS 6,00000.

Name of Experiment.....

Date.....

Experiment Result.....

Page No. 35

```
SELECT D.DNo, COUNT(*)  
FROM DEPARTMENT D, EMPLOYEE E  
WHERE D.DNo = E.DNo  
AND E.Salary > 60000  
AND D.DNo IN (SELECT E1.DNo  
FROM EMPLOYEE E1 GROUP  
BY E1.DNo HAVING COUNT(*) > 5)  
GROUP BY D.DNo;  
DNo - - - COUNT(*) - - -  
5 - - - - -  
3 - - - - -  
O. J. V.  
23/12/2019
```