## JADAVPUR UNIVERSITY

Database Management System

PROJECT & Lab AssignmentS

**Name:** SHRUTI PATHAK & ANNESA MONDAL **Roll No:** 002210503021 & 002210503022

**Dept:** Computer Science and Engineering

**Class:** MCA 1st Year 2nd Sem **Session:** 2022-2024

# *LIBRARY MANAGEMENT SYSTEM*

* **GENERAL DESCRIPTION :**

Library Management System is computerized system which helps staffs (librarian) to manage the library

daily activity in electronic format. It reduces the paperwork such as file lost, file damaged and time

consuming. It can help staff to manage the transaction or record more effectively and take less time .

* **PROBLEM STATEMENT :**

The problem occurred before having computerized system includes :

1. **File Lost:**

When computerized system is not implemented file is always lost because of human environment. Sometimes due to some human error there may be a loss of records.

1. **File Damaged:**

When computerized system is not there file is always lost due to some accident like spilling of water by some member on file accidentally. Besides some natural disaster like floods or fires may also damage files.

1. **Difficult To Search Records :**

When there is no computerized system there is always a difficulty in searching of records if the records are large in number.

1. **Space Consuming :**

After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

1. **Cost Consuming :**

As there is no computerized system to add, each record paper will be needed which will increase the cost of management of library.

* **SYSTEM ANALYSIS AND REQUIREMENTS:**

**>> NON FUNCTIONAL REQUIREMENTS**

**Product Requirements**

1. **EFFICIENCY REQUIREMENT :**

When a library Management system will be implemented librarian and user will easily access library as searching and book transaction will be very faster.

1. **RELIABILITY REQUIREMENT :**

The system should accurately performs member registration, member validation , report generation, book transaction and search.

1. **USABILITY REQUIREMENT :**

The system is designed for a user friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

**>> FUNCTIONAL REQUIREMENTS**

1. **NORMAL USER**
2. **USER LOGIN**

**>> Description of feature :**

This feature used by the user to login into the system. They are required to enter user id and Password before they are allowed to enter the system .Every member has different id and password.

The user id and password will be verified and if invalid id is there member is not allowed to enter the system.

**>> Functional requirements**

-user id is provided when they register

-The system must only allow user with valid id and password to enter the system

-The system performs authorization process which decides What user level can access to.

-The user must be able to logout after they finished using system.

1. **REGISTER NEW USER**

**>> Description of feature**

This feature can be performed by all users to register new user to Create account.

**>> Functional requirements**

-System must be able to verify information

-System must be able to delete information if information is wrong

1. **REGISTER NEW BOOK**

**>> Description of feature :**

This feature allows to add new books to the library

**>> Functional requirements**

-System must be able to verify information

-System must be able to enter number of copies into table.

- System must be able to not allow two books having same book id.

1. **SEARCH BOOK**

**>> Description of feature :**

This feature is found in book Maintenance part. We can search book based on book id, book name,

Publication or by author name,

**>> Functional requirements**

- System must be able to search the database based on select search type

- System must be able to filter book based on keyword enterd

- System must be able to show the filtered book in table view

-System should be able to add detailed information about events

* **SOFTWARE AND HARDWARE ANALYSYS AND REQUIREMENTS:**

**SOFTWARE :**

>> Any Operating system for we browse interface.

>> Database MYSQL-MYSQL is used as database as it easy to maintain and retrieve records by simple queries which are in English language which are easy to understand and easy to write.

**HARDWARE :**

>> Atleast 1GB storage on the server side.

>> Internet connection on both reader and server side.

* **DATA COLLECTION:**

1. **Books**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BookID** | **Title** | **Author** | **Book\_Status** | **Price** |
| B00001 | PROGRAMMING WITH JAVA | E BALAGURUSAMY | Available | 100 |
| B00002 | DATABASE MANAGEMENT SYSTEM | RAJIV CHOPRA | Available | 300 |
| B00003 | PYTHON PROGRAMMING FOR BEGINNERS | ANTHONY ADAMS | Available | 700 |
| B00004 | OPERATING SYSTEMS | WILLIAM STALLINGS | Not Available | 400 |
| B00005 | DATA STRUCTURES USING C | REEMA THAREJA | Available | 200 |

1. **Member**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **MembID** | **Name** | **Contact\_No.** | **Address** | **Memb\_Type** | **Memb\_Date** | **Expiry\_Date** |
| M00001 | Arjun Kapoor | 6778767564 | Alipore,Kolkata | Student | 2021-05-01 | 2026-04-30 |
| M00002 | Ishani Dutta | 9387827966 | Behala,Kolkata | Staff | 2020-08-01 | 2025-07-30 |
| M00003 | Kabir Singh | 8136976445 | Dhakuria,Kolkata | Teacher | 2018-10-01 | 2023-09-30 |
| M00004 | Ananya Pandey | 9855465765 | Jadavpur,Kolkata | Student | 2022-05-01 | 2027-04-30 |

1. **Borrowed by**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MembID** | **BookID** | **Issue\_Date** | **Due\_Date** | **Return\_Date** |
| M00001 | B00004 | 2023-03-12 | 2023-04-10 | 2023-04-12 |
| M00003 | B00002 | 2023-03-15 | 2023-04-15 | 2023-04-05 |
| M00004 | B00001 | 2023-02-24 | 2023-03-24 | 2023-03-28 |

1. **Publisher**

|  |  |  |  |
| --- | --- | --- | --- |
| **PubID** | **Name** | **Contact\_No.** | **Address** |
| P00001 | Mc Graw Hill | 6869875744 | Gariahat, Kolkata |
| P00002 | Pearson | 7834512879 | Dumdum, Kolkata |
| P00003 | OXFORD | 7812434509 | Park Street, Kolkata |
| P00004 | WILEY | 9823451289 | Ballygunge, Kolkata |

1. **Published by**

|  |  |
| --- | --- |
| **BookID** | **PubID** |
| B00001 | P00001 |
| B00002 | P00003 |
| B00005 | P00002 |

1. **Staff**

|  |  |  |  |
| --- | --- | --- | --- |
| **StaffID** | **Name** | **Contact\_no** | **Address** |
| S00001 | Aayan Kundu | 7346015361 | Tollygunge, Kolkata |
| S00002 | Ishita Dutta | 9572534269 | New Alipore, Kolkata |
| S00003 | Ananya Pandey | 9855465765 | Behala, Kolkata |
| S00004 | Jay Mahato | 9823451289 | Ballygunge, Kolkata |

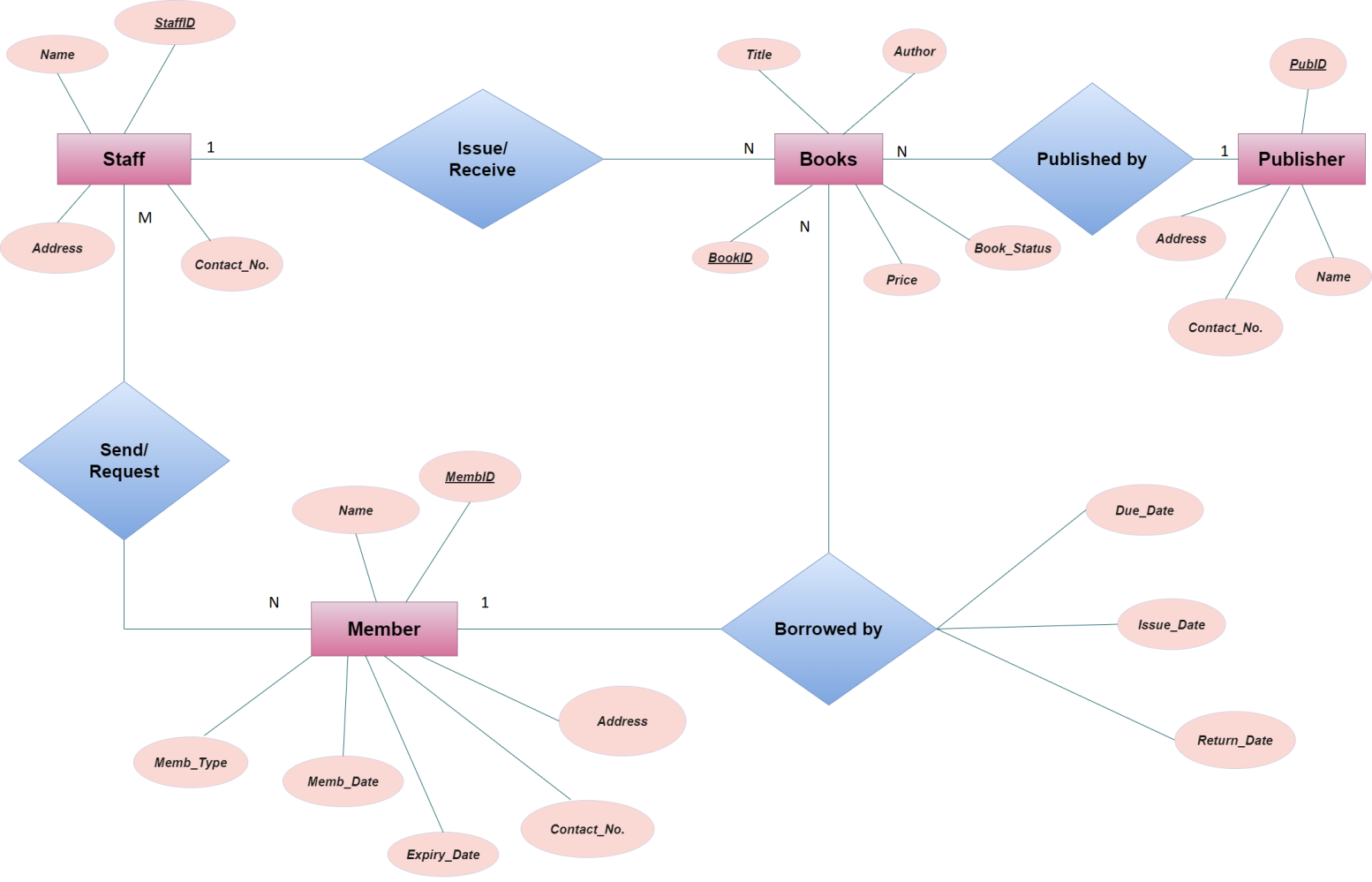
1. **Issue/Receive**

|  |  |
| --- | --- |
| **StaffID** | **BookID** |
| S00002 | B00004 |
| S00001 | B00002 |
| S00004 | B00001 |

1. **Send/Requests**

|  |  |
| --- | --- |
| **StaffID** | **MembID** |
| S00002 | M00001 |
| S00001 | M00003 |
| S00004 | M00004 |

* **E-R DIAGRAM:**



***ASSIGNMENT 1***

# *Question 1: Create the following tables:

**Table name: EMPLOYEE**

# Structure:

|  |  |
| --- | --- |
| EMP\_CODE | char(16) |
| EMP\_NAME | char(20) |
| DEPT\_CODE | char(16) |
| DESIG\_CODE | char(16) |
| SEX | char(1) |
| ADDRESS | char (25) |
| CITY | char (20) |
| STATE | char (20) |
| PIN | char (6) |
| BASIC | Number |
| JN\_DT | Date |

Primary key is EMP\_CODE

**Table name: DESIGNATION**

# Structure:

DESIG\_CODE char(16) DESIG\_DESC char(20)

Primary key is DESIG\_CODE

**Table name: DEPARTMENT**

# Structure:

DEPT\_CODE char(16)

DEPT\_NAME char(20)

Primary key is DEPT\_CODE.

# Solution:

# CREATE TABLE EMPLOYEE(

# EMP\_CODE varchar(16) NOT NULL PRIMARY KEY,

# EMP\_NAME varchar(20),

# DEPT\_CODE varchar(16),

# DESIG\_CODE varchar(16),

# SEX varchar(1),

# ADDRESS varchar(25),

# CITY varchar(20),

# STATE varchar(20),

# PIN varchar(6),

# BASIC int,

# JN\_DT Date);

0 row(s) affected

CREATE TABLE DESIGNATION(

# DESIG\_CODE varchar(16) NOT NULL PRIMARY KEY,

DESIG\_DESC varchar(20));

0 row(s) affected

CREATE TABLE DEPARTMENT(

DEPT\_CODE varchar(16) NOT NULL PRIMARY KEY,

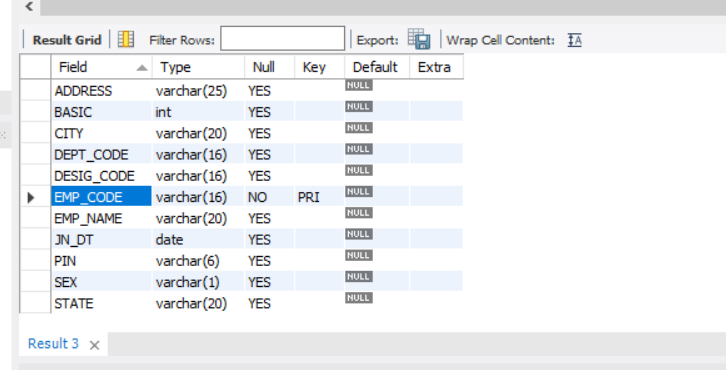
DEPT\_NAME varchar(20));

0 row(s) affected

# *Question 2: Display the structure of each table.

**Solution:**

DESC EMPLOYEE;



|  |  |
| --- | --- |
| DESC DESIGNATION; | |
| DESC DEPARTMENT; | |
|  |

# *Question 3. Insert few rows in each table.

**[While entering data in EMP table use DESIG\_CODE which exists in DESIGNATION table and DEPT\_CODE which is exists in DEPARTMENT table. In DESIGNATION table, assign code for Manager, Executive, officer, clerk and helper. In DEPARTMENT table, assign code for Personnel, Production, Purchase, Finance, Research departments]**

**Solution:**

INSERT INTO DEPARTMENT(DEPT\_CODE, DEPT\_NAME)

VALUES('PER','Personnel');

1 row(s) affected

INSERT INTO DEPARTMENT(DEPT\_CODE, DEPT\_NAME)

VALUES('PR','Production');

1 row(s) affected

INSERT INTO DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES('PUR','Purchase');

1 row(s) affected

INSERT INTO DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES('FI','Finance');

1 row(s) affeccted

INSERT INTO DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES('RE','Research');

1 row(s) affeccted

INSERT INTO DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES('MR','MARKETTIN);

1 row(s) affeccted

SELECT \* FROM DEPARTMENT;

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

INSERT INTO DESIGNATION(DESIG\_CODE, DESIG\_DESC) VALUES('MN','Manager');

1 row(s) affected

INSERT INTO DESIGNATION(DESIG\_CODE, DESIG\_DESC)

VALUES('EXE','Executive');

1 row(s) affected

INSERT INTO DESIGNATION(DESIG\_CODE, DESIG\_DESC) VALUES('OFF','Officer');

1 row(s) affected

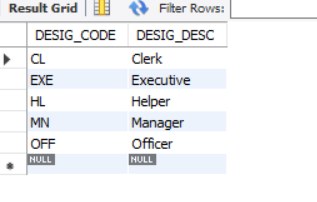
INSERT INTO DESIGNATION(DESIG\_CODE, DESIG\_DESC) VALUES('CL','Clerk');

1 row(s) affected

INSERT INTO DESIGNATION(DESIG\_CODE, DESIG\_DESC) VALUES('HL','Helper');

1 row(s) affected

SELECT \* FROM DESIGNATION;



INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E101','Shruti Pathak','PR','EXE','F','Purulia','Purulia','West Bengal','723102',45000,DATE '2015-05-01');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E102','Annesa Mondal','FI','OFF','F','Krishnagar','Nadia','West Bengal','741101',50000,DATE '2020-01-02')

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E103','Rahul Mahato','PUR','MN','M','Haldia','EastMidnapur','West Bengal','721108',60000,DATE '2020-05-01');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E104','Nilesh Kundu','RE','HL','M','Durgapur','West Burdwan','West Bengal','700014',90000,DATE '2018-02-01');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E105','Nitish Kumar Mondal','FI','CL','M','New Town','Kolkata','West Bengal','700026',25000,DATE '2022-01-02');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E106','Himangsu Bauri','PUR','EXE','M','Thakurpukur','Kolkata','West Bengal','400020',40000,DATE '2021-08-01');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E107','Ivy Dey','PR','MN','F','Garia','Kolkata','West Bengal','700009',

60000,DATE '2016-02-01');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E108','Saraswati Murmu','PER','OFF','F','Jadavpur','Kolkata','West Bengal','760038',55000,DATE '2020-08-01');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E109','Anjani Ahir','FI','CL','F','Naihati','North 24 Parbanas','West Bengal','710001',25000,DATE '2010-03-01');

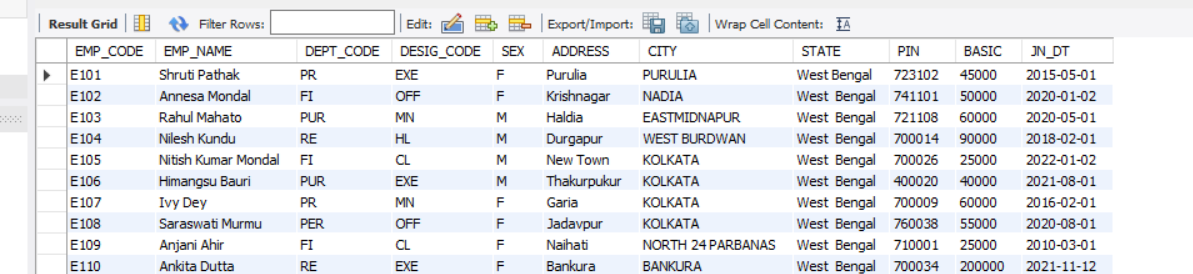
1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E110','Ankita Dutta','RE','EXE','F','Bankura','Bankura','West Bengal','700034',200000,DATE '2021-11-12');

1 row(s) affected

SELECT \* FROM EMPLOYEE;



# *Question 4: In EMP table insert few rows without DEPT\_CODE and BASIC.

**Solution**:

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E111','Priyanka Ghosh',NULL,'EXE','F','Dumdum','Kolkata','West Bengal','700034',NULL,DATE '2021-11-12');

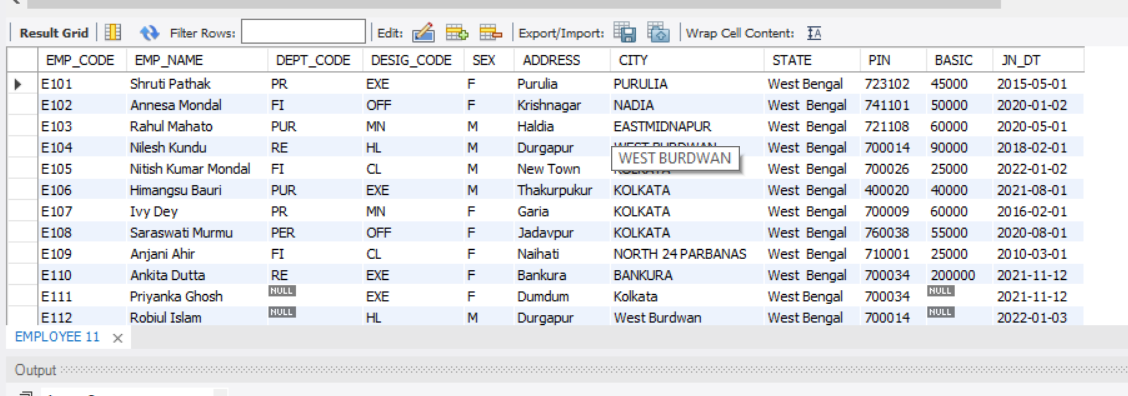
1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E112','Robiul Islam',NULL,'HL','M','Durgapur','West Burdwan','West Bengal','700014',NULL,DATE '2022-01-03');

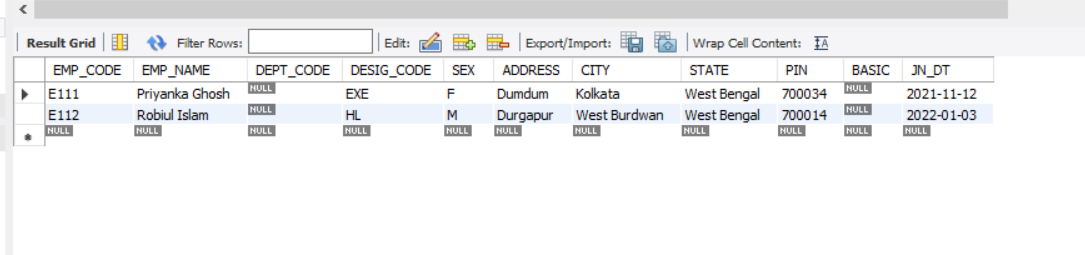
1 row(s) affected

SELECT \* FROM EMPLOYEE;



# *Question 5: Find the rows with unassigned DEPT\_CODE. Solution.

SELECT \* FROM EMPLOYEE WHERE DEPT\_CODE IS NULL;



# *Question 6: Find the rows with BASIC equal to 0.

**Solution**:

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

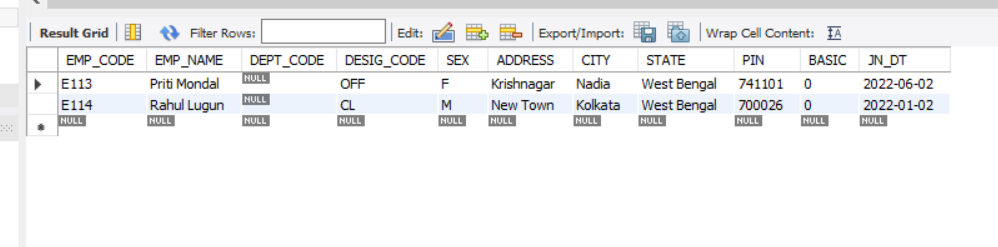
('E113','Priti Mondal',NULL,'OFF','F','Krishnagar','Nadia','West Bengal','741101',0,DATE '2022-06-02');

1 row(s) affected

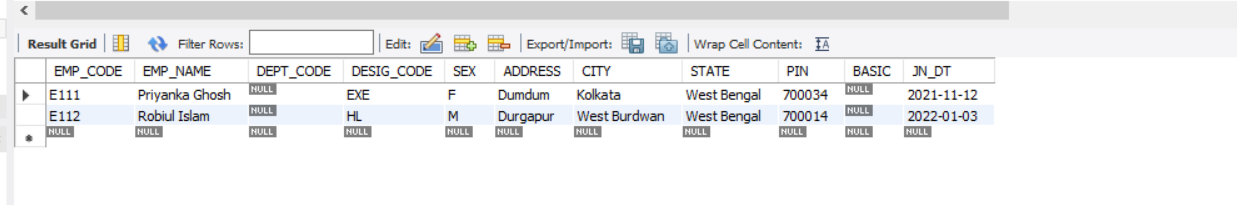
INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES

('E114','Rahul Lugun',NULL,'CL','M','New Town','Kolkata','West Bengal','700026',0,DATE '2022-01-02');

1 row(s) affected

SELECT \* FROM EMPLOYEE WHERE BASIC = 0;

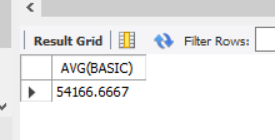
# *Question 7: Find the rows with unassigned Basic [note down the difference between the results obtained in Q.6 and Q.7] Solution:

 SELECT \* FROM EMPLOYEE WHERE BASIC IS NULL;

# *Question 8: Find the average BASIC of the employees.

**Solution:**

SELECT AVG(BASIC) FROM EMPLOYEE;



# *Question 9: Replace the BASIC with 0 for the rows with unassigned BASIC.

**Solution:**

UPDATE EMPLOYEE SET BASIC = 0

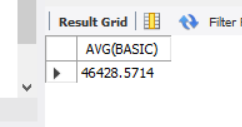
WHERE BASIC IS NULL;

2 row(s) affected Rows matched: 2 Changed: 2 Warnings:

# *Question 10. Find the average BASIC. (Note the difference of result obtained in Q.8 & Q.10.)

**Solution:**

SELECT AVG(BASIC) FROM EMPLOYEE;



# *Question 11. Delete the rows with unassigned DEPT\_CODE. Solution:

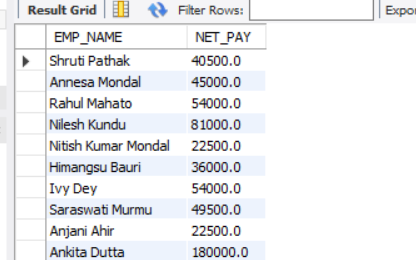
DELETE FROM EMPLOYEE WHERE DEPT\_CODE IS NULL;

4 row(s) affected

# *Question 12: Say, Net pay of an employee= Basic+ HRA+ DA where HRA is 50% of the Basic & DA is 40% of Basic. Show the employee name & Net pay for all employees.

**Solution:**

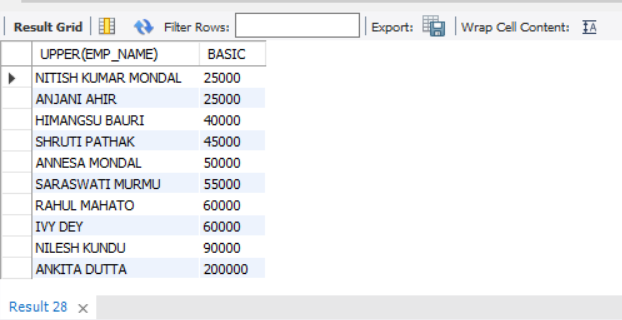
SELECT EMP\_NAME, BASIC \* 0.4 + BASIC \* 0.5 AS NET\_PAY FROM EMPLOYEE;



# *Question 13: Show the EMP\_NAME & BASIC in the ascending order of DEPT\_CODE. The employee name must appear in uppercase.

**Solution:**

SELECT UPPER(EMP\_NAME), BASIC

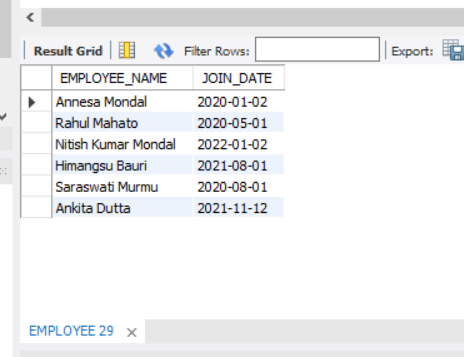
FROM EMPLOYEE

ORDER BY BASIC;

# *Question 14: Find the employees who have joined after 1st January 2010

**Solution:**

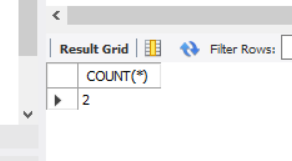
SELECT EMP\_NAME AS EMPLOYEE\_NAME, JN\_DT AS JOIN\_DATE FROM EMPLOYEE WHERE JN\_DT > DATE '2020-01-01';



# *Question 15: Find, how many employees have joined in the month of January?

**Solution:**

SELECT COUNT(\*) FROM EMPLOYEE WHERE SUBSTR(JN\_DT,6,2) = '01';

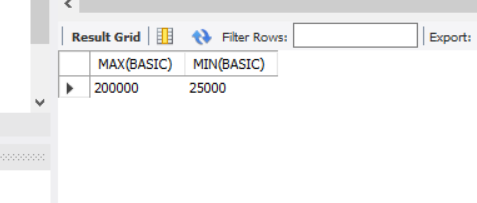


# *Question 16: Find the maximum & minimum Basic.

# 

# Solution:

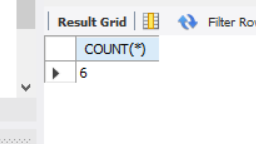
SELECT MAX(BASIC), MIN(BASIC) FROM EMPLOYEE;



# *Question 17: Find how many Female employees are there?

**Solution:**

SELECT COUNT(\*) FROM EMPLOYEE WHERE SEX = 'F'; COUNT(\*)



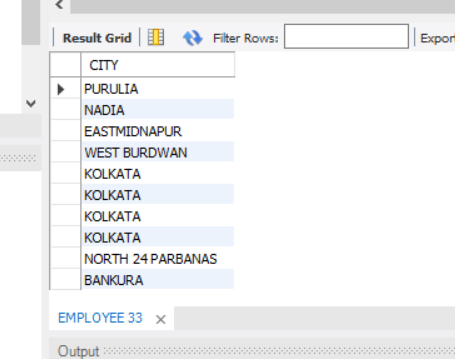
# *Question 18: Replace CITY with existing value converted into uppercase for all rows.

**Solution:**

UPDATE EMPLOYEE SET CITY = UPPER(CITY);

10 row(s) affected Rows matched: 10 Changed: 10 Warnings: 0

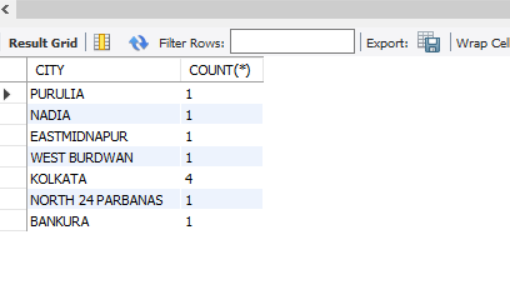
SELECT CITY FROM EMPLOYEE;



# *Question 19: Find in how many different cities various employees are residing?

**Solution:**

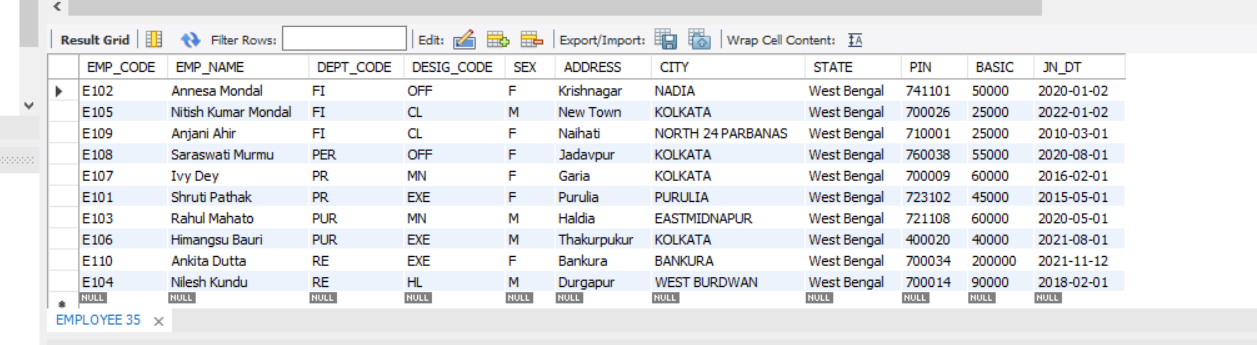
SELECT CITY, COUNT(\*) FROM EMPLOYEE GROUP BY CITY;



# *Question 20: Display the employee information in the ascending order of DEPT\_CODE and within a Department, it should be in the descending order of BASIC.

**Solution:**

SELECT \* FROM EMPLOYEE ORDER BY DEPT\_CODE, BASIC DESC;

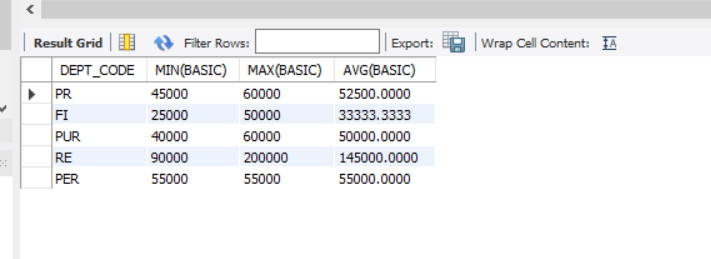


***ASSIGNMENT 2***

# Question 1: From the EMP table show the minimum, maximum and

# average basic for each department (show dept. Code). Solution:

SELECT DEPT\_CODE, MIN(BASIC), MAX(BASIC), AVG(BASIC) FROM EMPLOYEE GROUP BY DEPT\_CODE;

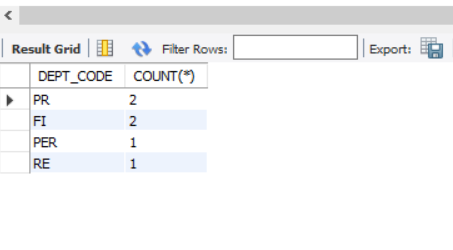


# Question 2: Find the number of female employees in each department (show dept. Code).

**Solution:**

SELECT DEPT\_CODE , COUNT(\*) FROM EMPLOYEE WHERE SEX = 'F'

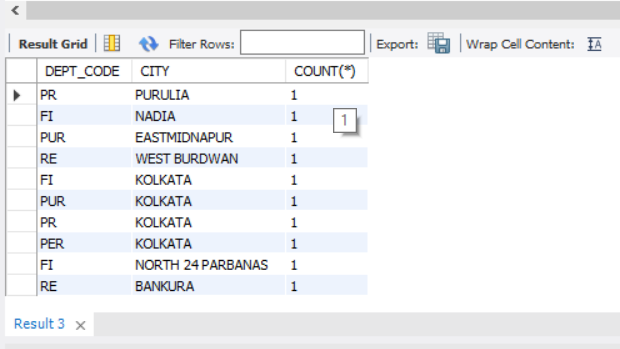
GROUP BY DEPT\_CODE;



# Question 3: Find the city wise no. of employees for each department (show dept. Code).

**Solution:**

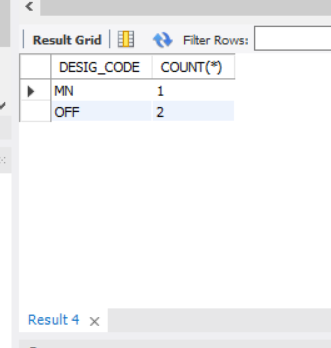
SELECT DEPT\_CODE, CITY, COUNT(\*) FROM EMPLOYEE GROUP BY DEPT\_CODE, CITY;



# Question 4: Show the designation wise no of employees who have joined in the year 2020 in each department. The listing should appear in the ascending order of no. of employees.

**Solution:**

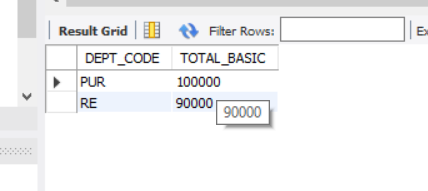
SELECT DESIG\_CODE , COUNT(\*) FROM EMPLOYEE WHERE SUBSTR(JN\_DT,8,2) = '20' GROUP BY DESIG\_CODE ORDER BY COUNT(\*);



# Question 5: Find the department code wise total basic of male employees only for the departments for which such total is more than 50,000 and the listing should appear in the descending order of total basic.

# Solution:

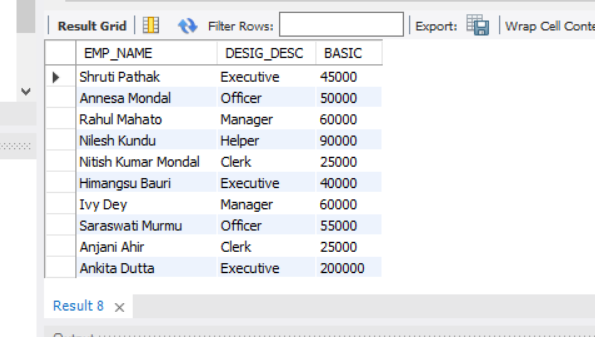
SELECT DEPT\_CODE , SUM(BASIC) AS TOTAL\_BASIC FROM EMPLOYEE WHERE SEX = 'M' GROUP BY DEPT\_CODE HAVING SUM(BASIC) > 50000 ORDER BY SUM(BASIC) DESC;



* **Question 6: Show the employee name, Designation description and basic for all employees.**

**Solution:**

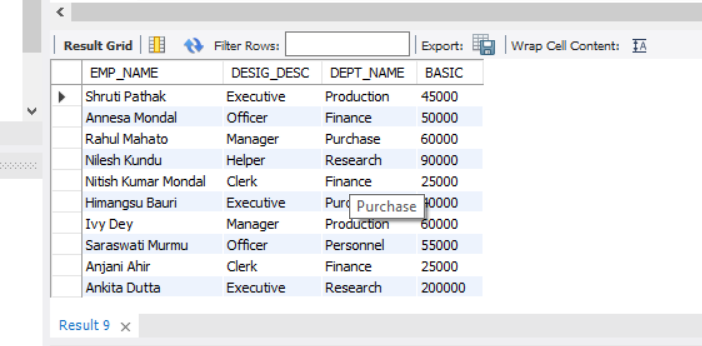
SELECT e.EMP\_NAME, d.DESIG\_DESC , e.BASIC FROM EMPLOYEE e, DESIGNATION d WHERE e.DESIG\_CODE = d.DESIG\_CODE;



# Question 7: Show the employee name, Designation description, Department Name & Basic for all employees.

**Solution:**

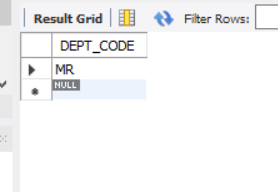
SELECT e.EMP\_NAME, d.DESIG\_DESC, dp.DEPT\_NAME, e.BASIC FROM EMPLOYEE e, DESIGNATION d, DEPARTMENT dp WHERE e.DESIG\_CODE = d.DESIG\_CODE AND e.DEPT\_CODE = dp.DEPT\_CODE;



# Question 8: Find the department Codes in which no employee works.

**Solution:**

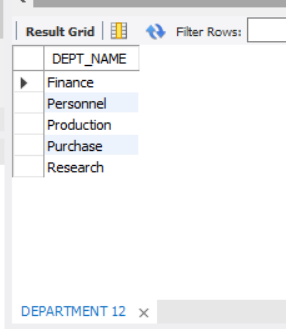
SELECT DEPT\_CODE FROM DEPARTMENT WHERE DEPT\_CODE NOT IN (SELECT DEPT\_CODE FROM EMPLOYEE);



# Question 9: Find the department names where at least one employee works.

**Solution:**

SELECT DEPT\_NAME FROM DEPARTMENT WHERE DEPT\_CODE IN (SELECT DEPT\_CODE FROM EMPLOYEE);



# Question 10: Find the department names where at least 10 employee’s works.

**Solution:**

# [I have not enough entries with respect to one department to run this query. So, I have added few more entries to run this query.]

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E115', 'Meghashri Saha', 'PR', 'OFF', 'F', 'New P.O. Road', 'Pune', 'Maharashtra', '421345',70000,DATE '2020-04-03');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE,DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT)VALUES('E116', 'Priyanka Roy', 'MN','F', 'Pali Hill', 'Mumbai','Maharashtra',

'400025', 65000, DATE '2019-09-02');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E117', 'Sudip Ghosh', 'PR', 'OFF', 'M', 'Ashutosh Mukherjee Road', 'Kolkata', 'West Bengal', '700020',65000, DATE '2019-02-09')

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E118', 'Arijit Maity', 'PR', 'EXE', 'M', '20B Narkeldanga', 'Kolkata','West Bengal', '711097', 45000, DATE '2020-07-03');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E119', 'Sneha Das', 'PR', 'CL', 'F', '5 Canning Street', 'Jalpaiguri', 'West Bengal', '723098', 35000, DATE '2021-08-03');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E120', 'Imanur Rahaman', 'PR', 'OFF', 'M', '128 Danesh Shekh Lane', 'Howrah', 'West Bengal', '711204', 70000, DATE '2019-10-09');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E121', 'Riya Parvin', 'PR', 'MN', 'F', 'Pune Lavasa Campus', 'Pune', 'Maharashtra', '400087', 65000, DATE '2020-09-03');

1 row(s) affected

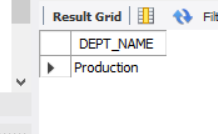
INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E122', 'Reshmi Kundu', 'PR', 'Cl', 'F', 'Lajpat Nagar', 'Delhi', 'Delhi','100087', 25000, DATE '2021-07-03');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES ('E123', 'Manshi Shaw', 'PR', 'EXE', 'F', 'IT Building,Yojana Bhawan', 'Jaipur', 'Rajasthan', '302005',50000, DATE '2019-08-06');

1 row(s) affected

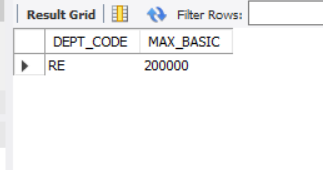
SELECT d.DEPT\_NAME FROM DEPARTMENT d WHERE (SELECT COUNT(\*) FROM EMPLOYEE e WHERE e.DEPT\_CODE = d.DEPT\_CODE) > 9;



# Question 11: Find the department code in which employee with highest Basic works.

**Solution:**

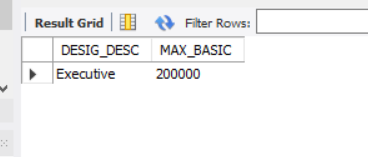
SELECT DEPT\_CODE, BASIC AS MAX\_BASIC FROM EMPLOYEE WHERE BASIC = (SELECT MAX(BASIC) FROM EMPLOYEE);



# Question 12: Find the Designation description of the employee with highest basic.

**Solution:**

SELECT d.DESIG\_DESC , e.BASIC AS MAX\_BASIC FROM EMPLOYEE e, DESIGNATION d WHERE e.DESIG\_CODE = d.DESIG\_CODE AND e.BASIC = (SELECT MAX(BASIC) FROM EMPLOYEE);

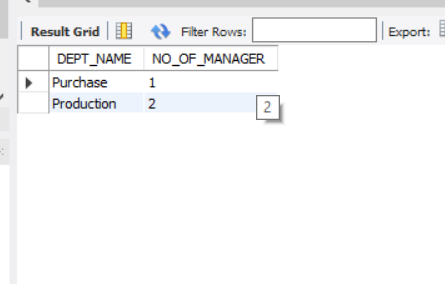


# Question 13: Find the no. of managers in each department.

**Solution:**

SELECT d.DEPT\_NAME, COUNT(\*) AS NO\_OF\_MANAGER FROM EMPLOYEE e, DEPARTMENT d WHERE e.DEPT\_CODE = d.DEPT\_CODE AND e.DESIG\_CODE = (SELECT DESIG\_CODE FROM DESIGNATION WHERE

DESIG\_DESC = 'Manager') GROUP BY d.DEPT\_NAME;

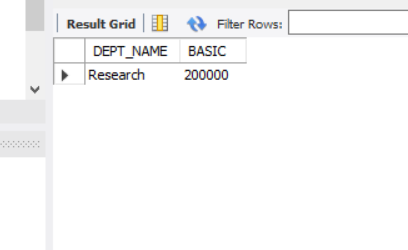


# Question 14: Find the maximum basic from EMP table without using MAX().

**Solution:**

SELECT DISTINCT BASIC AS MAX\_BASIC FROM EMPLOYEE e WHERE

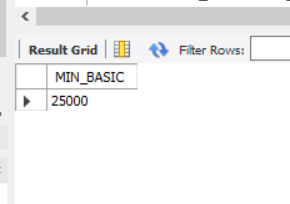
e.BASIC >= ALL(SELECT BASIC FROM EMPLOYEE);



# Question 15: Find the minimum basic from EMP table without using MIN().

**Solution:**

SELECT DISTINCT BASIC AS MIN\_BASIC FROM EMPLOYEE e WHERE e.BASIC <= ALL(SELECT BASIC FROM EMPLOYEE);

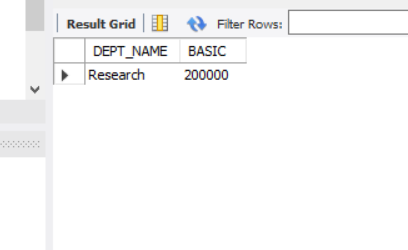


# Question 16: Find the name of the department with highest total basic. Do the same for highest average basic and maximum no. of employee.

**Solution:**

# Highest Total Basic:

SELECT d.DEPT\_NAME, e.BASIC FROM EMPLOYEE e, DEPARTMENT d WHERE e.DEPT\_CODE = d.DEPT\_CODE AND e.BASIC >= ALL(SELECT BASIC FROM EMPLOYEE);



# Highest Average Basic:

SELECT DEPT\_NAME, BASIC AS AVERAGE\_BASIC FROM

(SELECT DEPT\_NAME , AVG(BASIC) AS BASIC FROM EMPLOYEE e, DEPARTMENT d WHERE e.DEPT\_CODE = d.DEPT\_CODE GROUP BY DEPT\_NAME)WHERE BASIC = (SELECT MAX(BASIC) FROM (SELECT DEPT\_NAME, AVG(BASIC) AS BASIC FROM EMPLOYEE e, DEPARTMENT d WHERE e.DEPT\_CODE = d.DEPT\_CODE GROUP BY DEPT\_NAME));

DEPT\_NAME AVERAGE\_BASIC

Personnel

55000

Purchase 55000

# Maximum Number of Employee:

SELECT DEPT\_NAME, EMP\_NO AS MAXIMUM\_NO\_OF\_EMPLOYEE FROM

(SELECT DEPT\_NAME, COUNT(EMP\_CODE) AS EMP\_NO FROM EMPLOYEE e DEPARTMENT d WHERE e.DEPT\_CODE = d.DEPT\_CODE GROUP BY DEPT\_NAME) WHERE EMP\_NO = (SELECT MAX(EMP\_NO) FROM EMPLOYEE e, DEPARTMENT dWHERE e.DEPT\_CODE = d.DEPT\_CODE GROUP BY DEPT\_NAME));

DEPT\_NAME MAXIMUM\_NO\_OF\_EMPLOYEE

Production 10

# Question 17: Insert same rows into EMP table with designation code not existing in DESIGNATION table.

**Solution:**

INSERT INTO EMPLOYEE(EMP\_CODE,EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX,ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT) VALUES('E24','Sarthak dubey', 'PUR', 'SALES', 'M', '45B Gariahat Road', 'Kolkata', 'West Bengal','700034',45000,DATE '2019-02-01');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE,EMP\_NAME, DEPT\_CODE, DESIG\_CODE,SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT)VALUES ('E125','Saheli Das', 'PER', 'SALES', 'F', 'Budhdha Marg', 'Patna', 'Bihar', '600078',30000, DATE '2020-08-13');

1 row(s) affected

INSERT INTO EMPLOYEE(EMP\_CODE,EMP\_NAME, DEPT\_CODE, DESIG\_CODE,SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DT)VALUES ('E126', 'Kastury Ganguli', 'OFF', 'REPRESENTATIVE', 'F', 'Panjiyan Bhawan','Kota', 'Rajasthan', '305001', 45000,DATE '2019-08-10');

1 row(s) affected

# Question 18: Delete the rows from EMP table with invalid DESIG\_CODE.

**Solution:**

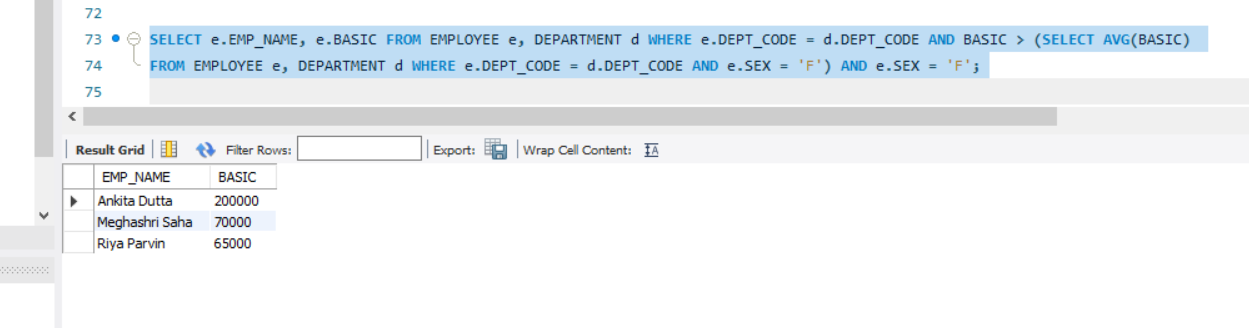
DELETE EMPLOYEE WHERE DESIG\_CODE NOT IN (SELECT DESIG\_CODE FROM DESIGNATION);

3 row(s) affected

# Question 19. Find the name of the female employees with basic greater than the average basic of their respective department

**Solution:**

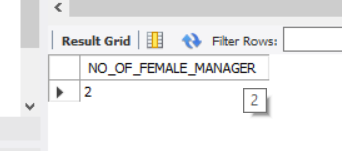
SELECT e.EMP\_NAME, e.BASIC FROM EMPLOYEE e, DEPARTMENT d WHERE e.DEPT\_CODE = d.DEPT\_CODE AND BASIC > (SELECT AVG(BASIC) FROM EMPLOYEE e, DEPARTMENT d WHERE e.DEPT\_CODE = d.DEPT\_CODE AND e.SEX = 'F') AND e.SEX = 'F';



# Question 20: Find the number of female managers

# Solution:

# SELECT COUNT(\*) AS NO\_OF\_FEMALE\_MANAGER FROM EMPLOYEE e, DESIGNATION d WHERE e.DESIG\_CODE = d.DESIG\_CODE AND e.SEX = 'F' AND d.DESIG\_CODE = 'MN';



***ASSIGNMENT 3***

# Problem 1:

**In an organization, number of departments exists. Each department has a name & unique code. Number of employees work in each department. Each employee has unique employee code.**

**Detailed information like name, address, city, basic, date of join are also stored. In a leave register for each employee leave records are kept showing leave type (CL/EL/ML etc.), fromdate and to- date. When an employee retires or resigns then all the leave information pertaining to him are also deleted. Basic salary must be within Rs.5000 to Rs.9000. A department cannot be deleted if any employee record refers to it. Valid grades are A/B/C. Employee name must be in uppercase only.**

**Default value for joining date is system date. Design & implement the tables with necessary constraints to support the scenario depicted above.**

# Solution:

CREATE TABLE DEPARTMENT(

DEPT\_CODE char(10) PRIMARY KEY,

DEPT\_NAME char(30),

NO\_OF\_EMPLOYEE INTEGER);

0 row(s) affected

CREATE TABLE EMPLOYEE(

EMP\_CODE char(10) PRIMARY KEY,

DEPT\_CODE char(10),

EMP\_NAME char(30) NOT NULL CONSTRAINT force\_upcase CHECK(BINARY EMP\_NAME = UPPER(EMP\_NAME)),

ADDRESS char(50),

CITY char(20),

BASIC NUMERIC(5) CONSTRAINT salary\_range CHECK(BASIC BETWEEN 5000 AND 9000),

JOIN\_DATE DATE DEFAULT(CURRENT\_DATE),

GRADE char(1) CHECK(GRADE IN ('A', 'B', 'C')),

CONSTRAINT DEPT\_FOREIGN\_KEY FOREIGN KEY(DEPT\_CODE) REFERENCES DEPARTMENT(DEPT\_CODE));

0 row(s) affected

CREATE TABLE LEAVE\_REG(

RECORD\_ID char(5) PRIMARY KEY,

EMP\_CODE char(10),

TYPE char(2),

FROM\_DATE DATE,

TO\_DATE DATE,

CONSTRAINT VALID\_TYPE CHECK(TYPE IN ('CL', 'EL', 'ML')),

FOREIGN KEY(EMP\_CODE) REFERENCES EMPLOYEE(EMP\_CODE));

0 row(s) affected

* ***Problem 2:***

**Try to violate the constraints that you have implemented in the table & note, what happens. [Try with suitable INSERT/UPDATE/DELETE instruction]**

# Solution:

INSERT INTO DEPARTMENT VALUES('D1', 'ACCOUNTS', 5);

1 row(s) affected

INSERT INTO DEPARTMENT VALUES('D2', 'MARKETING', 12);

1 row(s) affected

INSERT INTO DEPARTMENT VALUES('D3', 'DEVELOPMENT', 10);

1 row(s) affected

INSERT INTO DEPARTMENT VALUES('D4', 'SALES', 20);

1 row(s) affected

INSERT INTO DEPARTMENT VALUES('D5', 'PURCHASE', 18);

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E1', 'D2', 'Dhrub', '20B Old P.O. Road','Kolkata',5500,'2020-01-01', 'B');

Error Code: 3819. Check constraint 'force\_upcase' is violated.

## NOTE: Here I have intentionally violated my mentioned constraint that employee names must be in uppercase letters. I note that I get an error saying “Check constraint 'force\_upcase' is violated.”

INSERT INTO EMPLOYEE VALUES('E1', 'D2', 'DHRUV', '20B Old P.O. Road','Kolkata',5500,'2020-01-01', 'B');

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E2', 'D1', 'KAVYA', '35 Pali Hill','Mumbai',6000,'2020-06-01', 'A');

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E3', 'D2', 'PRIYA', '14A Gopal Banerjee Lane','Kolkata',7000,'2020-03-01', 'C');

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E4', 'D5', 'DIVYA', 'Lajpat Nagar','Delhi',6500,'2019-08-02', 'B');

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E5', 'D3', 'JIGYASA', 'New Palace Lane','Pune',8000,'2019-04-01', 'E');

Error Code: 3819. Check constraint 'employee\_chk\_1' is violated.

**NOTE: Here I have intentionally violated my mentioned constraint that Valid grades are A/B/C. I note that I get an error saying “Check constraint 'employee\_chk\_1' is violated.”**

INSERT INTO EMPLOYEE VALUES('E5', 'D3', 'JIGYASA', 'New Palace Lane','Pune',8000,'2019-04-01', 'A');

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E6', 'D3', 'SOURAV', '51A Shyama Prosad Road','Kolkata',5000,'2020-09-15', 'C');

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E7', 'D4', 'SHREYA', 'Mahatma Gandhi Sarani','Gujarat',8000,'2020-10-05', 'A');

1 row(s) affected

INSERT INTO EMPLOYEE VALUES('E7', 'D4', 'ANWESHA', 'Hill Road','Darjeeling',2000,'2019-05-02', 'B');

Error Code: 3819. Check constraint 'salary\_range' is violated.

INSERT INTO EMPLOYEE VALUES('E8', 'D4', 'ANWESHA', 'Hill Road','Darjeeling',5500,'2019-05-02', 'B');

**NOTE: Here I have intentionally violated my mentioned constraint that BASIC must be between 5000-9000. I note that I get an error saying “Check constraint 'salary\_range' is violated”.**

INSERT INTO EMPLOYEE VALUES('E8', 'D4', 'ANWESHA', 'Hill Road','Darjeeling',5500,'2019-05-02', 'B');

1 row(s) affected

INSERT INTO LEAVE\_REG VALUES('L1', 'E4','EL', '2020-03-01','2020-06-01');

1 row(s) affected

INSERT INTO LEAVE\_REG VALUES('L2', 'E2','ML', '2020-07-01','2020-09-01');

1 row(s) affected

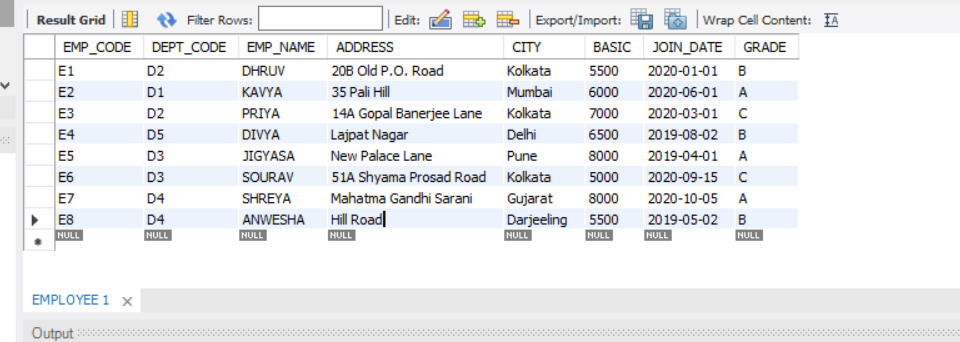
INSERT INTO LEAVE\_REG VALUES('L3', 'E5','ML', '2019-11-15','2019-12-10');

1 row(s) affected

INSERT INTO LEAVE\_REG VALUES('L4', 'E7','CL', '2020-03-01','2020-04-01');

1 row(s) affected

SELECT \* FROM EMPLOYEE;



# Problem 3:

**3. a) Create a view showing employee code, name, dcode & Basic For a particular department.**

1. **Try to ensure a row into the view with valid department & also with invalid ones.**
2. **Find the newly inserted row in the table From which view was created.**
3. **Try to increment basic by Rs.100/-**
4. **Check it in the original table.**
5. **Delete the view.**

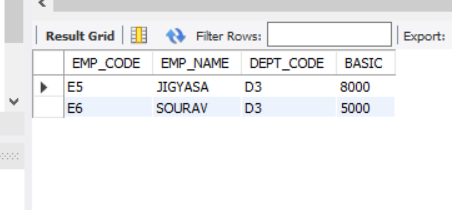
# Solution:

1. CREATE VIEW EMPLOYEE\_DETAILS AS SELECT EMP\_CODE, EMP\_NAME, DEPT\_CODE, BASIC FROM EMPLOYEE WHERE DEPT\_CODE = 'D3';

0 row(s) affected

SELECT \* FROM EMPLOYEE\_DETAILS;

2 row(s) returned



1. INSERT INTO EMPLOYEE\_DETAILS VALUES('E9', 'ANANYA', 'D7', 6700);

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`ass3`.`employee`, CONSTRAINT `DEPT\_FOREIGN\_KEY` FOREIGN KEY (`DEPT\_CODE`) REFERENCES `department` (`DEPT\_CODE`))

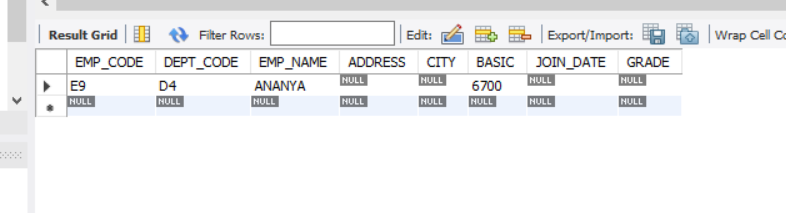
# NOTE: Here I have intentionally tried to insert a row in the view with an invalid DEPT\_CODE. I note that I have an error saying integrity constraint violated. This happens because there is no record in the Department table corresponding to the DEPT\_CODE ‘D7’.

INSERT INTO EMPLOYEE\_DETAILS VALUES('E9', 'ANANYA', 'D4', 6700);

1 row(s) affected

1. SELECT \* FROM EMPLOYEE WHERE EMP\_CODE = 'E9';

1 row(s) returned

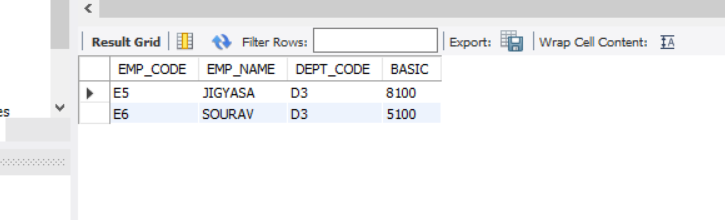


1. UPDATE EMPLOYEE\_DETAILS SET BASIC = BASIC + 100;

2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0

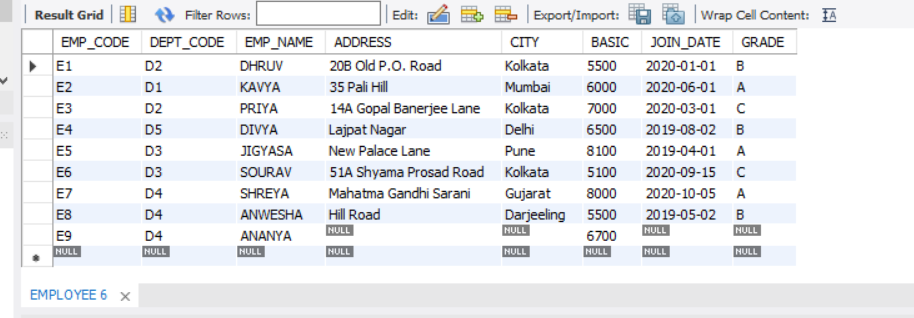
1. SELECT \* FROM EMPLOYEE\_DETAILS;

2 row(s) returned



SELECT \* FROM EMPLOYEE;

9 row(s) returned



1. DROP VIEW EMPLOYEE\_DETAILS;
2. row(s) affected

* ***Problem 4:***

1. **Create a view showing Emp\_Code, name, Dept\_Name, basic, leave type, From date & to date.**
2. **Try to insert a row in the view. Check what happens?**
3. **Try to increment basic by Rs.100.**
4. **Delete the view.**

***Solution:***

1. CREATE VIEW EMPLOYEE\_LEAVE\_DETAILS AS SELECT e.EMP\_CODE, e.EMP\_NAME, e. BASIC, e.DEPT\_CODE, d.DEPT\_NAME, l.TYPE, l.FROM\_DATE, l.TO\_DATE FROM EMPLOYEE e, DEPARTMENT d, LEAVE\_REG l WHERE e.DEPT\_CODE = d.DEPT\_CODE AND e.EMP\_CODE = l.EMP\_CODE;
2. row(s) affected

1. INSERT INTO EMPLOYEE\_LEAVE\_DETAILS VALUES ('E10', 'PRAKASH', 5600,'D3', 'DEVELOPMENT', 'CL','02-JAN-2022', '20JAN-2022');

Error Code: 1394. Can not insert into join view 'ass3.employee\_leave\_details' without fields list

***NOTE: Here I have tried to insert a row in the view EMPLOYEE\_LEAVE\_DETAILS and I have faced an error. This action is not possible as I am trying to insert some values in three tables simultaneously of which two table’s Primary Key is not mentioned.***

1. UPDATE EMPLOYEE\_LEAVE\_DETAILS SET BASIC = BASIC + 100;

4 row(s) affected Rows matched: 4 Changed: 4 Warnings: 0

1. DROP VIEW EMPLOYEE\_LEAVE\_DETAILS;

0 row(s) affected

# Problem 5:

* + 1. **Create a table having Emp\_code , Name, Dept\_name, & basic From the existing tables along with the records of the employee who are in a particular department (say, d1) and with a basic Rs. 7000/-**
    2. **From the existing table, add the employees with the basic salary greater than or equal to 7000/-**
    3. **Alter the table to add a net pay column.**
    4. **Replace net pay with 1.5\* Basic.**
    5. **Try to remove the net net pay column. [It may require no. of steps]**

**Solution:**

1. CREATE TABLE PERSONNEL(

EMP\_CODE char(10) PRIMARY KEY, EMP\_NAME char(30) , DEPT\_NAME char(30), BASIC NUMERIC(5))

AS SELECT EMP\_CODE, EMP\_NAME, DEPT\_NAME, BASIC FROM EMPLOYEE, DEPARTMENT

WHERE EMPLOYEE.DEPT\_CODE = DEPARTMENT.DEPT\_CODE AND BASIC = 7000 AND DEPARTMENT.DEPT\_CODE = 'D3';

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

1. INSERT INTO PERSONNEL SELECT EMP\_CODE, EMP\_NAME, DEPT\_NAME, BASIC FROM EMPLOYEE, DEPARTMENT WHERE EMPLOYEE.DEPT\_CODE = DEPARTMENT.DEPT\_CODE AND BASIC >= 7000 AND EMP\_CODE NOT IN (SELECT DISTINCT EMP\_CODE FROM PERSONNEL);

3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0

1. ALTER TABLE PERSONNEL ADD (NET\_PAY NUMERIC(9,2));

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

1. UPDATE PERSONNEL SET NET\_PAY = 1.5 \* BASIC;

3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0

1. ALTER TABLE PERSONNEL DROP COLUMN NET\_PAY;

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

# Problem 6:

# Drop all the tables that you have created.

# Solution:

# DROP TABLE PERSONNEL;

# 0 row(s) affected

# 

# DROP TABLE LEAVE\_REG;

# 0 row(s) affected

# DROP TABLE EMPLOYEE;

# 0 row(s) affected

# 

# DROP TABLE DEPARTMENT;

# 0 row(s) affected

***ASSIGNMENT 4***

# Problem 1:

1. Create EMP table with ECODE (Primary key), ENAME, DCODE, GRADE, BASIC & JN-DT as the columns. [Except BASIC & JN-DT, all columns are of char type and site of Grade is 1.]
2. Insert number of rows.

# Solution:

SQL> CREATE TABLE EMP(

* 1. ECODE char(10) PRIMARY KEY,
  2. ENAME char(30),
  3. DCODE char(10),
  4. GRADE char(1),
  5. BASIC number,
  6. JN\_DT DATE);

Table created.

SQL> INSERT INTO EMP VALUES('E1', 'Priya', 'D2', 'B', 5000, '02-JAN2020');

1 row created.

SQL> INSERT INTO EMP VALUES('E2', 'Manish', 'D1', 'C', 5500, '02-JAN2020');

1 row created.

SQL> INSERT INTO EMP VALUES('E3', 'Akash', 'D2', 'B', 5500, '06-FEB2020');

1 row created.

SQL> INSERT INTO EMP VALUES('E4', 'Rachna', 'D4', 'A', 7000, '06-APR2019');

1 row created.

SQL> INSERT INTO EMP VALUES('E5', 'Rachna', 'D3', 'B', 7500, '10AUG-2019');

1 row created.

SQL> INSERT INTO EMP VALUES('E6', 'Ritu', 'D4', 'C', 6000, '12-SEP2019');

1 row created.

SQL> INSERT INTO EMP VALUES('E7', 'Aritra', 'D3', 'A',8000, '06-JAN2021');

1 row created.

SQL> INSERT INTO EMP VALUES('E8', 'Piyush', 'D2', 'C', 7500, '17-OCT2021');

1 row created.

SQL> INSERT INTO EMP VALUES('E9', 'Navin', 'D1', 'B', 6800, '13-MAR2020');

1 row created.

SQL> INSERT INTO EMP VALUES('E10', 'Pritha', 'D4', 'A', 8500, '12-JUN2019');

1 row created.

# Problem 2:

Change the column heading as shown below, So that in subsequent SELECT statement newly set heading will be shown:

ECODE EMPLOYEE CODE

ENAME NAME

DCODE DEPT.CODE JN-DT JONING DATE

# Solution:

SQL> COLUMN ECODE HEADING 'EMPLOYEE\_CODE'; SQL> COLUMN ENAME HEADING 'NAME';

SQL> COLUMN DCODE HEADING 'DEPT\_CODE'; SQL> COLUMN JN\_DT HEADING 'JOINING DATE'; SQL> SELECT \* FROM EMP;

EMPLOYEE\_C NAME DEPT\_CODE G BASIC JOINING DATE

-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E1 | Priya | D2 | B | 5000 | 02-JAN-20 |
| E2 | Manish | D1 | C | 5500 | 02-JAN-20 |
| E3 | Akash | D2 | B | 5500 | 06-FEB-20 |
| E4 | Rachna | D4 | A | 7000 | 06-APR-19 |
| E5 | Rachna | D3 | B | 7500 | 10-AUG-19 |
| E6 | Ritu | D4 | C | 6000 | 12-SEP-19 |
| E7 | Aritra | D3 | A | 8000 | 06-JAN-21 |
| E8 | Piyush | D2 | C | 7500 | 17-OCT-21 |
| E9 | Navin | D1 | B | 6800 | 13-MAR-20 |
| E10 | Pritha | D4 | A | 8500 | 12-JUN-19 |

10 rows selected.

# Problem 3:

Set the format of columns as mentioned below, So that in subsequent SELECT statement, values appear in the specified format:

\*format of BASIC is such that a value of 7000 will be shown as 7,000

\*Format of GRADE will be such that full column name appears in the display.

\*For JN-DT format is such that 01-JAN-00 will be shown as JANURY 01, 2000.

# Solution:

SQL> COLUMN BASIC FORMAT '99,999'; SQL> COLUMN GRADE FORMAT A5;

SQL> ALTER SESSION SET NLS\_DATE\_FORMAT = 'MONTH DD,YYYY';

Session altered.

SQL> SELECT \* FROM EMP;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_C NAME | DEPT\_CODE | GRADE | BASIC JOINING DATE |  |
| E1 Priya | D2 B | 5,000 |  | JANUARY 02, 2020 |
| E2 Manish | D1 C | 5,500 |  | JANUARY 02, 2020 |
| E3 Akash | D2 B | 5,500 |  | FEBRUARY 06, 2020 |
| E4 Rachna | D4 A | 7,000 |  | APRIL 06, 2019 |
| E5 Rachna | D3 B | 7,500 |  | AUGUST 10, 2019 |
| E6 Ritu | D4 C | 6,000 |  | SEPTEMBER 12, 2019 |

|  |  |
| --- | --- |
| E7 Aritra D3 A 8,000 | JANUARY 06, 2021 |
| E8 Piyush D2 C 7,500 | OCTOBER 17, 2021 |
| E9 Navin D1 B 6,800 | MARCH 13, 2020 |
| E10 Pritha D4 A 8,500 | JUNE 12, 2019 |

10 rows selected.

# Problem 4:

1. Show the display attributes of all the columns.
2. Show the display attributes of particular column.
3. Suppress the newly set attributes of JN-DT .Try a select statement. (d) Reset the newly set attributes of JN-DT (e) Reset the newly set attributes of all columns.

(f) Shown the display attributes of all columns.

# Solution:

1. SQL> COLUMN; COLUMN GRADE ON FORMAT A5

COLUMN BASIC ON FORMAT 99,999

COLUMN JN\_DT ON HEADING 'JOINING DATE'

COLUMN DCODE ON HEADING 'DEPT\_CODE'

COLUMN ENAME ON HEADING 'NAME'

COLUMN ECODE ON HEADING 'EMPLOYEE\_CODE'

COLUMN result\_plus\_xquery ON HEADING 'Result Sequence'

COLUMN other\_plus\_exp ON FORMAT a44

COLUMN other\_tag\_plus\_exp ON FORMAT a29

COLUMN object\_node\_plus\_exp ON FORMAT a8

COLUMN plan\_plus\_exp ON FORMAT a60

COLUMN parent\_id\_plus\_exp ON HEADING 'p'

FORMAT 990

COLUMN id\_plus\_exp ON HEADING 'i'

FORMAT 990

COLUMN droptime\_plus\_show\_recyc ON HEADING 'DROP TIME'

FORMAT a19

COLUMN objtype\_plus\_show\_recyc ON HEADING 'OBJECT TYPE'

FORMAT a12

COLUMN objectname\_plus\_show\_recyc ON HEADING 'RECYCLEBIN NAME

FORMAT a30

COLUMN origname\_plus\_show\_recyc ON HEADING 'ORIGINAL NAME'

FORMAT a16

COLUMN value\_col\_plus\_show\_param ON HEADING 'VALUE'

FORMAT a30

COLUMN name\_col\_plus\_show\_param ON

HEADING 'NAME'

FORMAT a36

COLUMN units\_col\_plus\_show\_sga ON FORMAT a15

COLUMN name\_col\_plus\_show\_sga ON FORMAT a24

COLUMN ERROR ON FORMAT

A65 word\_wrap

COLUMN LINE/COL ON FORMAT A8

COLUMN ROWLABEL ON FORMAT A15

***(b)***

SQL> COLUMN JN\_DT;

COLUMN JN\_DT ON

HEADING 'JOINING DATE'

1. SQL> COLUMN JN\_DT OFF; SQL> SELECT \* FROM EMP;

EMPLOYEE\_C NAME DEPT\_CODE GRADE BASIC JN\_DT

|  |  |
| --- | --- |
| E1 Priya D2 B 5,000 | JANUARY 02, 2020 |
| E2 Manish D1 C 5,500 | JANUARY 02, 2020 |
| E3 Akash D2 B 5,500 | FEBRUARY 06, 2020 |
| E4 Rachna D4 A 7,000 | APRIL 06, 2019 |
| E5 Rachna D3 B 7,500 | AUGUST 10, 2019 |
| E6 Ritu D4 C 6,000 | SEPTEMBER 12, 2019 |
| E7 Aritra D3 A 8,000 | JANUARY 06, 2021 |
| E8 Piyush D2 C 7,500 | OCTOBER 17, 2021 |
| E9 Navin D1 B 6,800 | MARCH 13, 2020 |
| E10 Pritha D4 A 8,500 | JUNE 12, 2019 |

10 rows selected.

1. SQL> COLUMN JN\_DT CLEAR; SQL> SELECT \* FROM EMP;

EMPLOYEE\_C NAME DEPT\_CODE GRADE BASIC JN\_DT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E1 | Priya | D2 | B | 5,000 | JANUARY 02, 2020 |
| E2 | Manish | D1 | C | 5,500 | JANUARY 02, 2020 |
| E3 | Akash | D2 | B | 5,500 | FEBRUARY 06,2020 |

|  |  |  |
| --- | --- | --- |
| E4 Rachna D4 | A 7,000 APRIL 06, 2019 | |
| E5 Rachna D3 | B 7,500 AUGUST 10,2019 | |
| E6 Ritu D4 | C 6,000 SEPTEMBER 12,2019 | |
| E7 Aritra D3 | A 8,000 | JANUARY 06,2021 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E8 | Piyush D2 |  |  | C | 7,500 | OCTOBER 17, 2021 |
| E9 | Navin D1 |  |  | B | 6,800 | MARCH 13, 2020 |
|  | | | | A | 8,500 | JUNE 12,2019 |
| E10 Pritha D4 10 rows selected.  ***(e)*** SQL> CLEAR COLUMN; columns cleared SQL> SELECT \* FROM EMP; | | | | | | |
| ECODE | ENAME DCODE | G | BASIC |  | | JN\_DT |

|  |  |
| --- | --- |
| E1 Priya D2 B 5000 | JANUARY 02, 2020 |
| E2 Manish D1 C 5500 | JANUARY 02, 2020 |
| E3 Akash D2 B 5500 | FEBRUARY 06,2020 |
| E4 Rachna D4 A 7000 | APRIL 06,2019 |
| E5 Rachna D3 B 7500 | AUGUST 10,2019 |
| E6 Ritu D4 C 6000 | SEPTEMBER 12,2019 |
| E7 Aritra D3 A 8000 | JANUARY 06 ,2021 |
| E8 Piyush D2 C 7500 | OCTOBER 17,2021 |
| E9 Navin D1 B 6800 | MARCH 13,2020 |
| E10 Pritha D4 A 8500 | JUNE 12,2019 |

10 rows selected.

***(f)*** SQL> COLUMN;

SP2-0045: \* no COLUMN defined

# Problem 5:

. (a) Show the records from EMP table in the ascending order of DCODE.

DCODE value will be shown only for the first record of that department (same of DCODE is not repeated)

1. Further take measures so that, after displaying the records of a department it skips one line.
2. Further take measures so that records one also ordered on the basis of GRADE with in a department & same GRADE value is not repeated.
3. Take measure so that at the end of each GRADE in a department it will show average Basic for that grade in that department. At the end of each department, it will show the average & total Basic for the department. At the end of all departments it will show the overall total basic & average basic.

# Solution:

* 1. SQL> BREAK ON DCODE;

SQL> SELECT \* FROM EMP ORDER BY DCODE ASC;

ECODE ENAME DCODE G BASIC JN\_DT

-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [E2](#_bookmark0) | [Manish](#_bookmark0) | [D1](#_bookmark0) | [C](#_bookmark0) | [5500 02-JAN 4](#_bookmark0) |
| [E9](#_bookmark1) | [Navin](#_bookmark1) |  | [B](#_bookmark1) | [6800 13-MAR 11](#_bookmark1) |
| [E1](#_bookmark2) | [Priya](#_bookmark2) | [D2](#_bookmark2) | [B](#_bookmark2) | [5000 02-JAN 12](#_bookmark2) |
| [E3](#_bookmark3) | [Akash](#_bookmark3) |  | [B](#_bookmark3) | [5500 06-FEB 12](#_bookmark3) |
| [E8](#_bookmark4) | [Piyush](#_bookmark4) |  | [C](#_bookmark4) | [7500 17-OCT 12](#_bookmark4) |
| [E5](#_bookmark5)  [Aritra](#_bookmark5) | [Rachna](#_bookmark5) | [D3](#_bookmark5) [A](#_bookmark5) | [B](#_bookmark5) [8000](#_bookmark5) | [7500 10-AUG-19 E7](#_bookmark5)  [06-JAN 12](#_bookmark5) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E10 | Pritha | D4 | A | 8500 12-JUN-19 |
| E4 | Rachna |  | A | 7000 06-APR-19 |
| E6 | Ritu |  | C | 6000 12-SEP-19 |

10 rows selected.

* 1. SQL> BREAK ON DCODE SKIP 1;

SQL> SELECT \* FROM EMP ORDER BY DCODE ASC;

ECODE ENAME DCODE G BASIC JN\_DT

-

E2 Manish D1 C 5500 02-JAN-20

E9 Navin B 6800 13-MAR-20

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E1 | Priya | D2 | B | 5000 | 02-JAN-20 |
| E3 | Akash |  | B | 5500 | 06-FEB-20 |
| E8 | Piyush |  | C | 7500 | 17-OCT-21 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E5 | Rachna | D3 | B | 7500 | 10-AUG-19 |
|  | E7 Aritra | A | 8000 |  | 06-JAN-21 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E10 | Pritha | D4 | A | 8500 | 12-JUN-19 |
| E4 | Rachna |  | A | 7000 | 06-APR-19 |
| E6 | Ritu |  | C | 6000 | 12-SEP-19 |

10 rows selected.

SQL> BREAK ON DCODE SKIP 1 ON GRADE;

SQL> SELECT \* FROM EMP ORDER BY DCODE ASC, GRADE ASC;

ECODE ENAME DCODE G BASIC JN\_DT

- E9 Navin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D1 | B 6800 | 13-MAR-20 |  | |
| E2 | Manish |  | C | 5500 02-JAN-20 |
| E1 E3  E8 | Priya Akash  Piyush | D2 | B  C | 5000 02-JAN-20  5500 06-FEB-20  7500 17-OCT-21 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E7 | Aritra | D3 | A | 8000 06-JAN-21 |
| E5 | Rachna |  | B | 7500 10-AUG-19 |

E10 Pritha D4 A 8500 12-JUN-19

E4 Rachna 7000 06-APR-19

E6 Ritu C 6000 12-SEP-19

10 rows selected.

* 1. SQL> COLUMN GRADE FORMAT A20;

SQL> BREAK ON DCODE SKIP 1 ON GRADE SKIP 1;

SQL> COMPUTE AVG LABEL 'AVG BASIC ON GRADE' OF BASIC ON GRADE;

SQL> SELECT \* FROM EMP ORDER BY DCODE ASC, GRADE ASC;

ECODE ENAME DCODE GRADE BASIC JN\_DT

E9 Navin D1 B 6800 13-MAR-20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 6800

E2 Manish D1 C 5500 02-JAN-20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 5500

E1 Priya D2 B 5000 02-JAN-20 E3 Akash 5500 06-FEB-20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 5250

E8 Piyush D2 C 7500 17-OCT-21

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 7500

E7 Aritra D3 A 8000 06-JAN-21

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 8000

E5 Rachna D3 B 7500 10-AUG-19

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 7500

E10 Pritha D4 A 8500 12-JUN-19 E4 Rachna 7000 06-APR-19

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 7750

E6 Ritu D4 C 6000 12-SEP-19

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 6000

10 rows selected.

## Overall Average and Overall total BASIC at the end of the table EMP:

SQL> BREAK ON REPORT SKIP 3 ON DCODE SKIP 3 ON GRADE SKIP 1; SQL> COMPUTE AVG LABEL 'AVG BASIC ON GRADE' OF BASIC ON GRADE;

SQL> COMPUTE AVG LABEL "AVG BASIC ON DEPT" SUM LABEL "TOTAL BASIC ON DEPT" OF BASIC ON DCODE;

SQL> COMPUTE AVG LABEL "AVG BASIC" SUM LABEL "TOTAL BASIC" OF BASIC ON REPORT; SQL> SELECT \* FROM EMP ORDER BY DCODE ASC, GRADE ASC;

ECODE ENAME DCODE GRADE BASIC JN\_DT

E9 Navin D1 B 6800 13-MAR-20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ----------

AVG BASIC ON GRADE 6800

E2 Manish D1 C 5500 02-JAN-20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 5500

\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC 6150

TOTAL BASIC 12300

E1 Priya D2 B 5000 02-JAN-20 E3 Akash 5500 06-FEB-20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 5250

E8 Piyush D2 C 7500 17-OCT-

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 7500

\*\*\*\*\*\*\*\*\*\* ----------

AVG BASIC 6000

TOTAL BASIC 18000

E7 Aritra D3 A 8000 06-JAN-21

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ----------

AVG BASIC ON GRADE 8000

E5 Rachna B 7500 10-AUG-19

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* AVG BASIC ON GRADE 7500

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | \*\*\*\*\*\*\*\*\* |  | ---------- |  |
| AVG BASIC |  | 7750 |  |
| TOTAL BASI |  | 15500 |  |
| E10 | Pritha | D4 | A | 8500 | 12-JUN-19 |

E4 Rachna 7000 06-APR-19

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------- AVG BASIC ON GRADE 7750

|  |  |
| --- | --- |
| E6 Ritu D4 C | 6000 12-SEP-19 |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | ---------- |
| AVG BASIC ON GRADE | 6000 |
| \*\*\*\*\*\*\*\*\* | ---------- |
| AVG BASIC | 7166.66667 |
| TOTAL BASIC | 21500 |

10 rows selected.