RDBMS

**Practical’s Lists**

1. **Create a table for Customer.**

Column Name Format

cust\_id char(5)

Lname char(15)

Fname char(15)

Area char(2)

phone\_no number(8)

|  |
| --- |
| Create Table Customer(      cust\_id varchar(5),      Lname varchar(15),      Fname varchar(15),      Area varchar(2),      phone\_no bigint(8)      ); |

1. **Create a table for Movie Column Name Format**

mv\_no number (5)

Title char(25)

Type char(10)

Star char(25)

Price number(8,2)

|  |
| --- |
| Create Table Movie (      my\_no integer(5),      Title varchar(25),      Type varchar(10),      Star varchar(25),      Price decimal(8,2)      ); |

1. **Create a table for invoice Column Name Format**

inv\_no char(3)

mv\_no number(5)

cust\_id char(5)

issue\_date date

return\_date date

|  |
| --- |
| Create Table invoice (      inv\_no varchar(3),      my\_no integer(5),      cust\_id varchar(5),      issue\_date date,      return\_date date      ); |

1. **Insert the below Record in the Customer table.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cust\_id** | **lname** | **fname** | **area** | **Phone\_no** |
| a01 | Patel | Vijay | sa | 381334 |
| a02 | Saitwal | Vandana | mu | 556037 |
| a03 | Jaguste | Pramada | da | 372631 |
| a04 | Navindgi | Basu | ba | 666612 |
| a05 | Sreedhran | Ravi | va | - |
| a06 | - | Rukmini | ga | 512527 |

|  |
| --- |
| INSERT INTO `customer`  (`cust\_id`, `Lname`, `Fname`, `Area`, `phone\_no`)  VALUES  ('a01', 'Patel', 'Vijay', 'sa', '381334'),  ('a02', 'Saitwal', 'Vandana', 'mu', '556037'),  ('a03', 'Jaguste', 'Pramada', 'da', '372631'),  ('a04', 'Navindgi', 'Basu', 'ba', '666612'),  ('a05', 'Sreedhran', 'Ravi', 'va', ''),  ('a06', '', 'Rukmini', 'ga', '512527');  SELECT  \* FROM customer; |

1. **Insert the below record in the movie table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **mv\_no** | **title** | **type** | **Star** | **Price** |
| 1 | Bloody Vengeance | action | Jackie Chan | 180.95 |
| 2 | The firm | thriller | Tom Cruise | 200.00 |
| 3 | Pretty woman | romantic | Richarge Gere | 150.00 |
| 4 | Home alone | comedy | Macaulay Culkin | 150.55 |
| 5 | The fugitive | thriller | Harrison Ford | 200.00 |
| 6 | Coma | suspence | Michael Douglas | 100.00 |
| 7 | Dracula | horror | Gray Oldman | 150.00 |
| 8 | Quick change | comedy | Bill Murray | 100.00 |
| 9 | Gone with the wind | drama | Clarke Gable | 200.00 |
| 10 | Carry on doctor | comedy | Leslie Phillips | 100.00 |

|  |
| --- |
| INSERT INTO `movie`  (`my\_no`, `Title`, `Type`, `Star`, `Price`)  VALUES  ('1', 'Bloody Vengeance', 'action', 'Jackie Chan', '180.95'),  ('2', 'The Firm', 'Thriller', 'Tom Cruise', '200'),  ('3', 'Pretty Woman', 'Romantic', 'Richarge Gere', '150'),  ('4', 'Home Alone', 'Comedy', 'Macaulay Culkin', '150.55'),  ('5', 'The Fugitive', 'Thriller', 'Harrison Ford', '200'),  ('6', 'Coma', 'Suspence', 'Michael Douglas', '100'),  ('7', 'Dracula', 'Horror', 'Gray Oldman', '150'),  ('8', 'Quick Charge', 'Comedy', 'Bill Murray', '100'),  ('9', 'Gone with the Wind', 'Drama', 'Clarke Gable', '200'),  ('10', 'Carry on Doctor', 'Comedy', 'Leslie Phillips', '100');  SELECT \* FROM movie; |

1. **Insert the below record in the invoice table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| inv\_no | mv\_no | cust\_id | issue\_date | return\_date |
| i01 | 4 | a01 | 13-jan-96 | 25-jan-96 |
| i02 | 3 | a02 | 12-feb-96 | 15-feb-96 |
| i03 | 1 | a02 | 15-feb-96 | 18-feb-96 |
| i04 | 6 | a03 | 10-mar-96 | 13- mar -96 |
| i05 | 7 | a04 | 05-feb-96 | 08-feb-96 |
| i06 | 2 | a06 | 18-mar-96 | 21-mar-96 |
| i07 | 9 | a05 | 07-jan-96 | 10-jan-96 |
| i08 | 9 | a01 | 11-feb-96 | 14-feb-96 |
| i09 | 1 | a05 | 15-feb-96 | 28-feb-96 |

|  |
| --- |
| INSERT INTO `invoice`  (`inv\_no`, `my\_no`, `cust\_id`, `issue\_date`, `return\_date`)  VALUES  ('i01', '4', 'a01', '1996-01-13', '1996-01-25'),  ('i02', '3', 'a02', '1996-02-12', '1996-02-15'),  ('i03', '1', 'a02', '1996-02-15', '1996-02-18'),  ('i04', '6', 'a03', '1996-03-10', '1996-03-13'),  ('i05', '7', 'a04', '1996-02-05', '1996-02-08'),  ('i06', '2', 'a06', '1996-03-18', '1996-03-21'),  ('i07', '9', 'a05', '1996-01-07', '1996-01-10'),  ('i08', '9', 'ao1', '1996-02-11', '1996-02-14'),  ('i09', '1', 'a05', '1996-02-15', '1996-02-28');  SELECT \* FROM invoice; |

1. **Do the Following:**

## Create the table Client\_Master

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| CLIENTNO | Varchar2 | 6 |
| NAME | Varchar2 | 20 |
| ADDRESS | Varchar2 | 50 |
| CITY | Varchar2 | 20 |
| PINCODE | Int | 8 |
| STATE | Varchar2 | 20 |
| BAL\_DUE | Decimal | 10,2 |
|  |  |  |

|  |
| --- |
| CREATE TABLE Client\_Master (      Client\_No varchar(6),      Name varchar(20),      Address varchar(50),      City varchar(20),      Pincode bigint(8),      State varchar(20),      Bal\_Due Decimal(10,2)      ); |

Insert the following data into table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CLIENTNO** | **NAME** | **ADDRESS** | **CITY** | **PINCODE** | **STATE** | **Bal\_due** |
| C0001 | Rohan Joshi | Khapaitya Chala | Surat | 395003 | Gujarat | 15000 |
| C0002 | Mamta Muzumdar | Salt Lake | Kolkata | 460012 | West Bengal | 5000 |
| C0003 | Chhaya Bankar | Worli | Mumbai | 400054 | Maharashtra | 2000 |
| C0004 | Ashwini Rathod | Ghangaur Ghat | Udaipur | 780011 | Rajasthan | 7000 |
| C0005 | Ivan Bayross | Indiranagar | Bangalore | 560050 | Karnataka | 1500 |
| C0006 | Deepak Sharma | Bandra | Mumbai | 400002 | Maharashtra | 4300 |
| C0007 | Shymali Bhide | Juhu | Mumbai | 470912 | Maharashtra | 2100 |

|  |
| --- |
| INSERT INTO Client\_Master  (Client\_No, Name, Address, City, Pincode, State, Bal\_Due)  VALUES  ('C0001', 'Roshan Joshi', 'Khapaitya Chakla', 'Surat', 395003, 'Gujarat', '15000'),  ('C0002', 'Mamta Mazumdar', 'Salt Lake', 'Kolkata', 460012, 'West Bengal', '5000'),  ('C0003', 'Chhaya Bankar', 'Worli', 'Mumbai', 400054, 'Maharashtra', '2000'),  ('C0004', 'Ashwini Rathod', 'Ghangaur Ghat', 'Udaipur', 780011, 'Rajasthan', '7000'),  ('C0005', 'Ivan Bayross', 'Indiranagar', 'Bangalore', 560050, 'karnataka', '1500'),  ('C0006', 'Deepak Sharma', 'Bandra', 'Mumbai', 400002, 'Maharashtra', '4300'),  ('C0007', 'Shymali Bhide', 'Juhu', 'Mumbai', 470912, 'Maharashtra', '2100');  SELECT \* FROM client\_master; |

**Queries :-**

* 1. List the details of the client according to the bal\_due
  2. List all clients who are located in Mumbai
  3. Show different types of state in “Client\_Master” table by eliminating the repeated states.
  4. Change the city of client no “C0005” to Mangalore.
  5. Change the bal\_due of client no “C0001” to Rs. 1000
  6. Delete from Client\_master where the state holds the value ”Rajasthan”
  7. Add a column name “Mobile” of data type “Number” & size=”10”.
  8. Create a table “Balance\_Details” having three 3 fields (ClientNo, Name, Bal\_Due) from the source table name “Client\_master” and rename the field Bal\_Due to Balance.

|  |
| --- |
| CREATE TABLE Balance\_Details (      Client\_No Varchar(6),      Name varchar(20),      Bal\_Due decimal(10,2)      );  INSERT INTO `balance\_details`  (`Client\_No`, `Name`, `Bal\_Due`)  VALUES  ('C0001', 'Roshan Joshi', '15000'),  ('C0002', 'Mamta Muzumdar', '500'),  ('C0003', 'Chhaya Bankar', '2000'),  ('C0004', 'Ashwini Rathod', '7000'),  ('C0005', 'Ivan Bayross', '1500'),  ('C0006', 'Deepak Sharma', '4300'),  ('C0007', 'Shymali Bhide', '2100'); |

* 1. Change the name of “Client\_Master” table to “Customer”

|  |
| --- |
| * + 1. SELECT \* FROM Client\_Master ORDER BY Bal\_Due; |
| * + 1. SELECT \* from client\_master where City = 'Mumbai'; |
| * + 1. SELECT State FROM client\_master GROUP BY State; |
| * + 1. UPDATE client\_master SET City = 'Mangalore' where Client\_No = 'C0005'; |
| * + 1. UPDATE client\_master SET Bal\_Due = 1000 where Client\_No = 'C0001'; |
| * + 1. DELETE from client\_master where State = 'Rajasthan'; |
| 7. ALTER TABLE client\_master Add (Phone\_no bigint(10)); |
| 8. CREATE TABLE Balance\_Details (Client\_No Varchar(6), Name varchar(20), Bal\_Due decimal(10,2));  ALTER TABLE `balance\_details` CHANGE `Bal\_Due` `Balance` DECIMAL(10,2); |
| 9. ALTER TABLE client\_master RENAME TO Customers; |

1. **DO the Following:**

Table Name : Employee

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Employee\_no** | **First\_name** | **Last\_name** | **Salary** | **Joining date** | **Department** |
| 1 | John | Abraham | 100000 | 01-JAN-13 | Banking |
| 2 | Michael | Clarke | 80000 | 01-APR-13 | Insurance |
| 3 | Roy | Thomas | 70000 | 21-May-13 | Banking |
| 4 | Tom | Jose | 60000 | 08-Dec-13 | Insurance |
| 5 | Jerry | Pinto | 65000 | 11-Feb-14 | Marketing |
| 6 | Philip | Mathew | 45000 | 01-Jul-14 | Services |
| 7 | John | Henry | 55000 | 01-Jan-15 | Technical |
| 8 | Ivan | Bayross | 60000 | 01-Aug-15 | Sales |
| |  | | --- | | CREATE TABLE Employee (      Employee\_no integer(3),      First\_name VARCHAR(10),      Last\_name varchar(10),      Salary decimal(8,2),      Joining\_date date,      Department VARCHAR(10)      );  INSERT INTO Employee  (Employee\_no, First\_name, Last\_name, Salary, Joining\_date, Department)  VALUES  (1,'John','Abraham','100000','2013-01-01','Banking'),  (2,'Michael','Clarke','80000','2013-04-01', 'Insurance'),  (3,'Roy','Thomas','70000','2013-05-21','Banking'),  (4,'Tom','Jose','60000','2013-12-08','Insurance'),  (5,'Jerry','Pinto','65000','2014-02-11','Marketing'),  (6,'Philip','Mathew','45000',2014-01-07,'Services'),  (7,'John','Henry','55000','2015-01-01','Technical'),  (8,'Ivan','Bayross','60000','2015-08-01','Sales'); |   **Table Name : Incentives** | | | | | | | |

|  |  |  |
| --- | --- | --- |
| **Employee\_Ref\_Id** | **Incentive\_date** | **Incentive\_amount** |
| 1 | 01-Feb-13 | 5000 |
| 2 | 01-Dec-13 | 3000 |
| 3 | 01-Mar-13 | 4000 |
| 4 | 21-Mar-15 | 4500 |
| 5 | 01-Sep-15 | 3500 |

|  |
| --- |
| CREATE TABLE Incentives (      Employee\_Ref\_Id integer(3),      Incentive\_date date,      Incentive\_amount decimal(8,2)      );  INSERT INTO Incentives  (Employee\_Ref\_Id,Incentive\_date,Incentive\_amount)  VALUES  (1,'2013-02-01','5000'),  (2,'2013-12-01','3000'),  (3,'2013-03-01','4000'),  (4,'2015-04-21','4500'),  (5,'2015-09-01','3500'); |

* 1. Create primary key Employee\_id in Employee Table
  2. Create EMPLOYEE\_REF\_ID in INCENTIVES table as foreign key with respect to EMPLOYEE\_ID in employee table
  3. Get all employee details from the employee table
  4. Get First\_Name,Last\_Name from employee table.
  5. Get First\_Name from employee table using alias name “Employee Name”
  6. Get First\_Name from employee table in upper case
  7. Get First\_Name from employee table in lower case.
  8. Get unique DEPARTMENT from employee table

|  |
| --- |
| ALTER TABLE employee ADD PRIMARY KEY (Employee\_no);  Alter table Incentives ADD Foreign Key (Employee\_Ref\_Id) References Employee (Employee\_no);  SELECT \* FROM employee;  SELECT First\_name, Last\_name FROM employee;  SELECT First\_name "Employee Name" from Employee;  SELECT upper(First\_name) FROM employee;  SELECT lower(First\_name) FROM employee;  SELECT DISTINCT Department from Employee; |

* 1. **Queries of Employee table.**
     1. Get all employee details from the employee table order by First\_Name Ascending
     2. Get all employee details from the employee table order by First\_Name descending
     3. Get all employee details from the employee table order by First\_Name Ascending and Salary descending
     4. Get employee details from employee table whose employee name is “John” (like)
     5. SELECT \* from EMPLOYEE where FIRST\_NAME='John'
     6. Get employee details from employee table whose employee name are “John” and “Roy”
     7. Get employee details from employee table whose first name starts with 'J'
     8. Get employee details from employee table whose first name contains 'o'
     9. Get employee details from employee table whose first name ends with 'n'
     10. Get employee details from employee table whose first name ends with 'n' and name contains 4 letters
     11. Get employee details from employee table whose first name starts with 'J' and name contains 4 letters
     12. Get employee details from employee table whose Salary greater than 60000
     13. Get employee details from employee table whose Salary less than 80000
     14. Get employee details from employee table whose Salary between 50000 and 80000
     15. Get employee details from employee table whose name is 'John' and 'Michael'.
     16. Get position of 'o' in name 'John' from employee table (skip)
     17. Get employee details from employee table whose salary is minimum
     18. Get employee details from employee table whose salary is maximum
     19. Count the total number of department from employee table
     20. Calculate the average salary of employee from employee

|  |
| --- |
| SELECT \* from Employee order by First\_name ASC;  SELECT \* from Employee order by First\_name DESC;  SELECT \* from Employee order by First\_name ASC, Salary DESC;  SELECT \* from Employee where First\_name = 'John';  SELECT \* from Employee where First\_name = 'John';  SELECT \* from Employee where First\_name in ('John','Roy');  SELECT \* from Employee where First\_name like 'J%';  SELECT \* from Employee where First\_name like '%o%';  SELECT \* from Employee where First\_name like '%n';  SELECT \* from Employee where First\_name like '\_\_\_n';  SELECT \* from Employee where First\_name like 'J\_\_\_';  SELECT \* from Employee where Salary > 60000;  SELECT \* from Employee where Salary < 80000;  SELECT \* from Employee where Salary BETWEEN 50000 and 80000;  SELECT \* from Employee where First\_name in ('John','Michael');  SELECT POSITION("o" IN "John") AS "Position of o in John";  SELECT MIN(Salary) as Minimum\_salary from Employee;  SELECT MAX(Salary) as Maximimum\_salary from Employee;  SELECT Department, COUNT(\*) FROM Employee GROUP BY Department;  SELECT AVG(Salary) as "Average Salary" from Employee; |

* 1. **Queries**
     1. Get First\_Name from employee table in upper case
     2. Get First\_Name from employee table in lower case.
     3. Get position of 'o' in name 'John' from employee table
     4. SELECT first 3 characters of FIRST\_NAME from EMPLOYEE
     5. Get FIRST\_NAME from employee table after removing white spaces from right side
     6. Get FIRST\_NAME from employee table after removing white spaces from left side.
     7. Get length of FIRST\_NAME of all employees from employee table
     8. Get First\_Name from employee table after replacing 'o' with '$'
     9. Get First\_Name and Last\_Name as single column from employee table separated by a '\_'
     10. Get FIRST\_NAME ,Joining year, Joining Month and Joining Date from employee table separated by '\_'
     11. Get employee details from employee table whose joining year is “2013”.
     12. Get employee details from employee table whose joining month is “January”
     13. Get employee details from employee table who joined before January 1st 2013
     14. Get employee details from employee table who joined after January 31st
     15. Get Joining Date and Time from employee table
     16. Get difference between JOINING\_DATE and INCENTIVE\_DATE from employee and incentives table.

|  |
| --- |
| SELECT UPPER(First\_name) from employee;  SELECT LOWER(First\_name) from employee;  SELECT POSITION("o" IN "John") AS "Position of o in John";  SELECT SUBSTRING( First\_name, 1, 3 ) FROM employee;  SELECT RTRIM(First\_name) as "First Name" from employee;  SELECT LTRIM(First\_name) FROM employee;  SELECT LENGTH(First\_name) as "Length of First Name" from employee;  SELECT REPLACE(First\_name, 'o','$') from employee;  SELECT CONCAT(First\_name, "\_", Last\_name) AS Employee FROM employee;  SELECT CONCAT(First\_name, "\_", YEAR(Joining\_date), "\_", MONTH(Joining\_date), "\_", Day(Joining\_date)) AS Employee FROM employee;  SELECT \* FROM Employee WHERE YEAR(Joining\_date) = 2013;  SELECT \* FROM Employee WHERE MONTH(Joining\_date) = 01;  SELECT \* FROM Employee WHERE Joining\_date < 2013-01-01;  SELECT \* FROM  employee where Joining\_date > '2013-01-31';  SELECT DATE(Joining\_date) as Date, TIME(Joining\_date) as Time from employee;  SELECT First\_name, Incentive\_date - Joining\_date as "Date Difference" from employee A inner join incentives B on A.Employee\_no=B.Employee\_Ref\_Id; |

* 1. **Queries**

## Find out how many employees are there in each department

* + 1. Find out total salary per department.
    2. Find out the average salary per department.
    3. Show list of departments who has more than 1 employee
    4. Show list of department whose total salary is greater than 50000
    5. Show list of department whose average salary is less than 50000
    6. Show list of department whose average salary is between 50000 and 80000
    7. Show the total no of employees whose joining month is same.
    8. Show the total no of employees whose joining year is same.
    9. Find total salary who have joined in same month
    10. Find total salary who have joined in same month and total salary is greater than 50000
    11. SELECT employee details from employee table if data exists in incentive table
    12. Display the employee name of all those who received their intencives
    13. Find out the employees who have their incentives less than 5000
    14. Update incentive table where employee name is 'John'
    15. SELECT first\_name, incentive amount from employee and incentives table for those employees who have incentives
    16. SELECT first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000
    17. SELECT first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount less than 3000
    18. SELECT first\_name, incentive amount from employee and incentives table for all employes even if they didn't get incentives

|  |
| --- |
| SELECT Department, count(\*) FROM employee group by Department;  SELECT Department, SUM(salary)FROM employee group by Department;  SELECT Department,AVG(salary) FROM employee group by Department;  SELECT Department FROM employee group by Department having count(employee\_no) >1;  SELECT Department FROM employee group by Department having SUM(salary) > 50000;  SELECT Department FROM employee group by Department having AVG(salary) < 50000;  SELECT Department FROM employee group by Department having AVG(salary) between 50000 and 80000;  SELECT \* FROM employee e WHERE month(joining\_date) IN (SELECT month(joining\_date) FROM employee WHERE e.employee\_no <> employee\_no);  SELECT \* FROM employee e WHERE year(joining\_date) IN (SELECT year(joining\_date) FROM employee WHERE e.employee\_no <> employee\_no);  SELECT SUM(salary) FROM employee e WHERE month(joining\_date) IN (SELECT month(joining\_date) FROM employee WHERE e.employee\_no <> employee\_no);  SELECT first\_name,incentive\_amount FROM employee inner join incentives on employee.employee\_no=incentives.employee\_ref\_id;  SELECT first\_name,incentive\_amount FROM employee inner join incentives on employee.employee\_no=incentives.employee\_ref\_id WHERE Incentive\_amount<5000;  SELECT first\_name,incentive\_amount FROM employee inner join incentives on employee.employee\_no=incentives.employee\_ref\_id WHERE Incentive\_amount > 3000;  SELECT first\_name,Incentive\_amount FROM employee inner join incentives on employee.employee\_no=incentives.employee\_ref\_id WHERE Incentive\_amount < 3000;  SELECT first\_name,Incentive\_amount FROM employee left join incentives on employee.employee\_no=incentives.employee\_ref\_id; |

* 1. **Do the Following:**

Create a table as following:

Table Name 🡪 Dept

deptno Dname Loc

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

|  |
| --- |
| CREATE TABLE dept (      deptno INT(3),      dname VARCHAR(10),      location VARCHAR(10)      );  INSERT INTO dept  (deptno, dname, location)  VALUES  (10,'Accounting','New York'),  (20,'Research','Dallas'),  (30,'Sales','Chicago'),  (40,'Operations','Boston'); |

Table Name 🡪 **emp**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Emp\_no** | **Ename** | **Job** | **Mgr** | **hiredate** | **Sal** | **Comm** | **deptno** |
| 7639 | King | President |  | 17-11-1981 | 5000 |  | 10 |
| 7698 | Blake | Manager | 7839 | 01-05-1981 | 2850 |  | 30 |
| 7782 | Clerk | Manager | 7839 | 09-06-1981 | 2450 |  | 10 |
| 7566 | Jones | Manager | 7839 | 02-04-1981 | 2975 |  | 20 |
| 7788 | Scott | Analyst | 7566 | 13-07-1987 | 3000 |  | 20 |
| 7902 | Ford | Analyst | 7566 | 03-12-1981 | 3000 |  | 20 |
| 7369 | Smith | Clerk | 7902 | 17-12-1980 | 800 |  | 20 |
| 7499 | Allen | Salesman | 7698 | 20-02-1981 | 1600 | 300 | 30 |
| 7521 | Ward | Salesman | 7698 | 22-02-1981 | 1250 | 500 | 30 |
| 7654 | Martin | Salesman | 7698 | 28-09-1981 | 1250 | 1400 | 30 |
| 7844 | Turnor | Salesman | 7698 | 08-09-1981 | 1500 |  | 30 |
| 7876 | Adams | Clerk | 7788 | 13-07-1987 | 1100 |  | 20 |
| 7900 | James | Clerk | 7698 | 03-12-1981 | 950 |  | 30 |
| 7934 | Miller | Clerk | 7782 | 23-01-1982 | 1300 |  | 10 |

|  |
| --- |
| CREATE TABLE EMP (      Emp\_no int(5),      Ename varchar(15),      Job varchar(15),      Mgr int(5),      hiredate date,      Sal decimal(8,2),      Comm integer(4),      deptno integer(3)      );  INSERT INTO emp  (Emp\_no,Ename,Job,Mgr,hiredate,Sal,Comm,deptno)  VALUES  (7839, 'King', 'President', '', '1981-11-17', 5000, '', 10),  (7698,'Blake','Manager',7839,'1981-05-01',2850,'',30),  (7782,'Clerk','Manager',7839,'1981-06-09',2450,'',10),  (7566,'Jones','Manager',7839,'1981-04-02',2975,'', 20),  (7788,'Scott','Analyst',7566,'1987-07-13',3000,'',20),  (7902,'Ford','Analyst',7566,'1981-12-03',3000,'',20),  (7369,'Smith','Clerk',7902,'1980-12-17',800,'',20),  (7499, 'Allen', 'Salesman', 7698, '1981-02-20', 1600, 300, 30),  (7521, 'Ward', 'Salesman', 7698, '1981-02-22', 1250, 500, 30),  (7654, 'Martin', 'Salesman', 7698, '1981-09-28', 1250, 1400, 30),  (7844, 'Turnor', 'Salesman', 7698, '1981-09-08', 1500, '', 30),  (7876, 'Adams', 'Clerk', 7788, '1987-07-13', 1100, '', 20),  (7900, 'James', 'Clerk', 7698, '1981-07-13', 950, '', 20),  (7934, 'Miler', 'Clerk', 7782, '1982-01-23', 1300, '', 10); |

12. Queries

1. SELECT all record from emp table where deptno =10 or 40.
2. SELECT all record from emp table where deptno=30 and sal>1500.
3. SELECT all record from emp where job not in SALESMAN or CLERK.
4. SELECT all record from emp where ename in 'BLAKE','SCOTT','KING'and'FORD'
5. SELECT all records where ename starts with ‘S’ and its lenth is 6 char.
6. SELECT all records where ename may be any no of character but it should end with ‘R’.
7. List the emps who are joined in the year 1981
8. List the emps who are joined in the month of Aug 1980
9. Display the avg salaries of all CLERKS
10. List all the emps except ‘president’ & ‘Manager’ in asc order of salaries
11. Count MGR and their salary in emp table.
12. In emp table add comm+sal as total sal.
13. SELECT any salary <3000 from emp table.
14. SELECT all salary <3000 from emp table.
15. SELECT all the employee group by deptno and sal in descending order.
16. List the emps who are working under Manager
17. List all the clerks of deptno 20
18. Find the 3rd MAX salary in the emp table.
19. Find the 3rd MIN salary in the emp table.

|  |
| --- |
| SELECT \* FROM DEPT WHERE deptno in (10, 40);  SELECT \* FROM emp WHERE Deptno = 30 and Sal > 1500;  SELECT \* FROM emp WHERE Job NOT IN('Salesman','Clerk');  SELECT \* FROM emp WHERE Ename in('Blake', 'Scott', 'King', 'Ford');  SELECT \* FROM emp WHERE Ename Like 'S\_\_\_\_\_';  SELECT \* FROM emp WHERE Ename like '%r';  SELECT \* FROM emp WHERE YEAR(hiredate)=1981;  SELECT \* FROM emp WHERE hiredate BETWEEN '1980-08-01' and '1980-08-31';  SELECT AVG(Sal) AS 'Average Salary' FROM emp WHERE Job = 'Clerk';  SELECT \* FROM emp WHERE Job not in('Manager','President') ORDER BY Sal ASC;  SELECT COUNT(Mgr) AS 'Total MGR',COUNT(Sal) AS 'Total Salary' FROM emp;  SELECT Ename,Sal+Comm AS 'Salary with Commision' FROM emp;  -- 13 number is wrong question  SELECT \* FROM emp WHERE Sal<3000;  SELECT \* FROM emp GROUP BY Deptno ORDER BY Sal DESC;  SELECT \* FROM emp WHERE Mgr not in ('');  SELECT \* FROM emp WHERE Deptno=20 and Job='Clerk';  SELECT \* FROM emp ORDER BY Sal DESC LIMIT 3;  SELECT \* FROM emp ORDER BY Sal ASC LIMIT 3; |

1. **PL/SQL Programs**
   1. Hello World Program in PL/SQL.

|  |
| --- |
| begin  dbms\_output.put\_line('Hello World');  end;  / |

* 1. PL/SQL Program To add Two Numbers.

|  |
| --- |
| DECLARE  num1 integer;  num2 integer;  add integer;  BEGIN  num1 := 21;  num2 := 4;  add := num1 + num2;  DBMS\_OUTPUT.PUT\_LINE(' Addition of '||num1||' and '||num2||'  is -> '||sum);  END;  / |

* 1. PL/SQL Program For Prime Number.

|  |
| --- |
| declare  num1 number;  num2 number;  temp number;    begin  num1:=:Enter\_Number;  num2:= 2;  temp:= 1;  for num2 in 2..num1/2      loop          if mod(num1, num2) = 0          then              temp:= 0;              exit;          end if;      end loop;        if temp = 1      then          dbms\_output.put\_line(num1||' is a Prime Number');      else          dbms\_output.put\_line(num1||' is NOT a Prime Number');      end if;  end; |

* 1. PL/SQL Program To Find Factorial of a Number.

|  |
| --- |
|  |

* 1. PL/SQL Program to Print Table of a Number.

|  |
| --- |
|  |

* 1. PL/SQL Program for Reverse of a Number

|  |
| --- |
|  |

* 1. PL/SQL Program for Fibonacci Series

|  |
| --- |
|  |

* 1. PL/SQL Program to Check Number is Odd or Even

|  |
| --- |
|  |

* 1. PL/SQL Program to Reverse a String

|  |
| --- |
|  |

* 1. Pl/SQL Program for Palindrome Number

|  |
| --- |
|  |

* 1. PL/SQL Program to Swap two Numbers

|  |
| --- |
|  |

* 1. PL/SQL Program for Armstrong Number

|  |
| --- |
|  |

* 1. PL/SQL Program to Find Greatest of Three Numbers

|  |
| --- |
|  |

* 1. PL/SQL Program to Print Patterns

|  |
| --- |
|  |

1. **PL/SQL Cursor Programs**
   1. Write a Program for Implicit Cursor
   2. Write a Program For Explicit Cursor
2. **Trigger Programs Create three tables**

Student (Roll\_no, Name, Contact, Marks) Student\_copy (Roll\_no, Contact)

Student\_update\_copy (Roll\_no, New\_Contact, Old\_contact)

1. Create a trigger to insert Roll no and Contact number of student on insertion of any record in Table Student.
2. Create a trigger to insert Roll no New Contact number and old Contact number of student on updation of contact number in Table Student.
3. **Procedure Programs.**
   1. Write a procedure to insert data in employee table.
   2. Write a procedure to update contact number of employee in employee table.
   3. Write a procedure to find name of manager for given employee id.
   4. Write a procedure to get all the details (emp\_id, name, city of residence,
   5. company name, city of work, manager name, salary) of given