

# 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));**

### 2. 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 (mv\_no integer(5), Title varchar(25), Type varchar(10), Star varchar(25), Price decimal(8,2));**

### 3. 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);**

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

Cust_id	Iname	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');**

**Insert the below record in the Movie table**

**5.**

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');**

6. 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`, `mv_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', 'a01',
'1996-02-11', '1996-02-14'), ('i09', '1', 'a05', '1996-02-15', '1996-02-28');
```

7. 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)
```

### 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.

```
INSERT INTO `balance_details` (`Client_No`, `Name`, `Balance`) 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);
```

9. Change the name of "Client\_Master" table to "Customer"

1. SELECT * FROM Client_Master ORDER BY Bal_Due;
2. SELECT * from client_master where City = 'Mumbai';
3. SELECT State FROM client_master GROUP BY State;
4. UPDATE client_master SET City = 'Mangalore' where Client_No = 'C0005';
5. UPDATE client_master SET Bal_Due = 1000 where Client_No = 'C0001';
6. 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;

**8. 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) PRIMARY KEY, 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', '01/07/2014', '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

```

1. ALTER TABLE employee ADD PRIMARY KEY (Employee_no);
2. Alter table Incentives ADD Foreign Key (Employee_Ref_Id) References
Employee (Employee_no);
3. SELECT * FROM employee;
4. SELECT First_name, Last_name FROM employee;
5. SELECT First_name "Employee Name" from Employee;
6. SELECT First_name, Last_name FROM employee;
7. SELECT First_name "Employee Name" from Employee;
8. SELECT DISTINCT Department from Employee;

```

#### **9. 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

```
1. SELECT * from Employee order by First_name ASC;
2. SELECT * from Employee order by First_name DESC;
3. SELECT * from Employee order by First_name ASC, Salary DESC;
4. SELECT * from Employee where First_name = 'John';
5. SELECT * from Employee where First_name = 'John';
6. SELECT * from Employee where First_name in ('John','Roy');
7. SELECT * from Employee where First_name like 'J%';
8. SELECT * from Employee where First_name like '%o%';
9. SELECT * from Employee where First_name like '%n';
10. SELECT * from Employee where First_name like '___n';
11. SELECT * from Employee where First_name like 'J___';
12. SELECT * from Employee where Salary > 60000;
13. SELECT * from Employee where Salary < 80000;
14. SELECT * from Employee where Salary BETWEEN 50000 and 80000;
15. SELECT * from Employee where First_name in ('John','Michael');
16. SELECT POSITION("o" IN "John") AS "Position of o in John";
17. SELECT MIN(Salary) as Minimum_salary from Employee;
18. SELECT MAX(Salary) as Maximum_salary from Employee;
19. SELECT Department, COUNT(*) FROM Employee GROUP BY Department;
20. SELECT AVG(Salary) as Average_salary from Employee;
```

#### 10. 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.

```
1.  SELECT UPPER(First_name) from employee;
2.  SELECT LOWER(First_name) from employee;
3.  SELECT POSITION("o" IN "John") AS "Position of o in John";
4.  SELECT SUBSTRING( First_name, 1, 3 ) FROM employee;
5.  SELECT (RTRIM(First_name) as "First Name" from employee;
6.  SELECT LTRIM(First_name) FROM employee;
7.  SELECT LENGTH(First_name) as "Length of First Name" from employee;
8.  SELECT REPLACE(First_name, 'o','$') from employee;
9.  SELECT CONCAT(First_name, "_", Last_name) AS Employee FROM
employee;
10. SELECT CONCAT(First_name, "_", YEAR(Joining_date), "_",
MONTH(Joining_date), "_", Day(Joining_date)) AS Employee FROM employee;
11. SELECT * FROM Employee WHERE YEAR(Joining_date) = 2013;
12. SELECT * FROM Employee WHERE MONTH(Joining_date) = 01;
13. SELECT * FROM Employee WHERE Joining_date < 2013-01-01;
14. SELECT * FROM employee where Joining_date > '2013-01-31';
15. SELECT DATE(Joining_date) as Date, TIME(Joining_date) as Time from
employee;
16. 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;
```

#### 11. Queries

1. Find out how many employees are there in each department
2. Find out total salary per department.
3. Find out the average salary per department.
4. Show list of departments who has more than 1 employee
5. Show list of department whose total salary is greater than 50000
6. Show list of department whose average salary is less than 50000
7. Show list of department whose average salary is between 50000 and 80000
8. Show the total no of employees whose joining month is same.
9. Show the total no of employees whose joining year is same.



10. Find total salary who have joined in same month

11. Find total salary who have joined in same month and total salary is greater than 50000

12. SELECT employee details from employee table if data exists in incentive table

13. Display the employee name of all those who received their incentives

14. Find out the employees who have their incentives less than 5000

15. Update incentive table where employee name is 'John'

16. SELECT first\_name, incentive amount from employee and incentives table for those employees who have incentives

17. SELECT first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000

18. SELECT first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount less than 3000

19. SELECT first\_name, incentive amount from employee and incentives table for all employees even if they didn't get incentives

```
1. SELECT Department, count(*) FROM employee group by Department;
2. SELECT Department, SUM(salary)FROM employee group by Department;
3. SELECT Department,AVG(salary) FROM employee group by Department;
4. SELECT Department FROM employee group by Department having count(employee_no)
>1;
5. SELECT Department FROM employee group by Department having SUM(salary) > 50000;
6. SELECT Department FROM employee group by Department having AVG(salary) < 50000;
7. SELECT Department FROM employee group by Department having AVG(salary) between
50000 and 80000;
8. SELECT * FROM employee e WHERE month(joining_date) IN (SELECT
month(joining_date) FROM employee WHERE e.employee_no <> employee_no);
9. SELECT * FROM employee e WHERE year(joining_date) IN (SELECT
year(joining_date) FROM employee WHERE e.employee_no <> employee_no);
10. SELECT SUM(salary) FROM employee e WHERE month(joining_date) IN (SELECT
month(joining_date) FROM employee WHERE e.employee_no <> employee_no);
11. SELECT first_name,incentive_amount FROM employee inner join incentives on
employee.employee_no=incentives.employee_ref_id;
12. SELECT first_name,incentive_amount FROM employee inner join incentives on
employee.employee_no=incentives.employee_ref_id WHERE Incentive_amount<5000;
13. SELECT first_name,incentive_amount FROM employee inner join incentives on
employee.employee_no=incentives.employee_ref_id WHERE Incentive_amount > 3000;
14. SELECT first_name,Incentive_amount FROM employee inner join incentives on
employee.employee_no=incentives.employee_ref_id WHERE Incentive_amount < 3000;
15. SELECT first_name,Incentive_amount FROM employee left join incentives on
employee.employee_no=incentives.employee_ref_id;
```

12. 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, 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.

1. **SELECT \* FROM DEPT WHERE Deptno in(10,40);**
2. **SELECT \* FROM emp WHERE Deptno = 30 and Sal > 1500;**
3. **SELECT \* FROM emp WHERE Job NOT IN('Salesman','Clerk');**
4. **SELECT \* FROM emp WHERE Ename in('Blake', 'Scott', 'King', 'Ford');**
5. **SELECT \* FROM emp WHERE Ename Like 'S\_\_\_\_\_';**
6. **SELECT \* FROM emp WHERE Ename like '%r';**
7. **SELECT \* FROM emp WHERE YEAR(hiredate)=1981;**
8. **SELECT \* FROM emp WHERE hiredate BETWEEN '1980-08-01' and '1980-08-31';**
9. **SELECT AVG(Sal) AS 'Average Salary' FROM emp WHERE Job = 'Clerk';**
10. **SELECT \* FROM emp WHERE Job not in('Manager','President') ORDER BY Sal ASC;**
11. **SELECT COUNT(Mgr) AS 'Total MGR',COUNT(Sal) AS 'Total Salary' FROM emp;**
12. **SELECT Ename,Sal+Comm AS 'Salary with Commision' FROM emp;**
13.
14. **SELECT \* FROM emp WHERE Sal<3000;**
15. **SELECT \* FROM emp GROUP BY Deptno ORDER BY Sal DESC;**
16. **SELECT \* FROM emp WHERE Mgr not in ('');**
17. **SELECT \* FROM emp WHERE Deptno=20 and Job='Clerk';**
18. **SELECT \* FROM emp ORDER BY Sal DESC LIMIT 3;**
19. **SELECT \* FROM emp ORDER BY Sal ASC LIMIT 3;**

### **13. PL/SQL Programs**

1. Hello World Program in PL/SQL.
2. PL/SQL Program To add Two Numbers.
3. PL/SQL Program For Prime Number.
4. PL/SQL Program To Find Factorial of a Number.
5. PL/SQL Program to Print Table of a Number.
6. PL/SQL Program for Reverse of a Number
7. PL/SQL Program for Fibonacci Series
8. PL/SQL Program to Check Number is Odd or Even
9. PL/SQL Program to Reverse a String
10. PL/SQL Program for Palindrome Number
11. PL/SQL Program to Swap two Numbers
12. PL/SQL Program for Armstrong Number
13. PL/SQL Program to Find Greatest of Three Numbers
14. PL/SQL Program to Print Patterns

### **14. PL/SQL Cursor Programs**

1. Write a Program for Implicit Cursor
2. Write a Program For Explicit Cursor

### **15. 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)**

A. Create a trigger to insert Roll no and Contact number of student on insertion of any record in Table Student.

B. Create a trigger to insert Roll no New Contact number and old Contact number of student on updation of contact number in Table Student.

### **16. 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