

INDEX

<i>SR.NO.</i>	<i>TITLE</i>	<i>DATE</i>	<i>SIGN</i>
1	To study DDL-create and DML-insert commands.		
2	To Perform various Select operation		
3	To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.		
4	To study Single-row functions.		
5	Displaying data from Multiple Tables (join)		
6	To apply the concept of Aggregating Data using Group functions.		
7	To solve queries using the concept of sub query.		
8	To apply the concept of Cursor		
9	To apply the concept of procedure		
10	To apply the concept of Function		

PRECTICAL 1

////////////////////////////////////*******CREATING TABLES FOR DATABASE*******////////////////////////////////////

SQL> CREATE TABLE DIPOSIT (ACTNO VARCHAR2(5),CNAME VARCHAR2(18),BNAME
VARCHAR2(18),AMOUNT
NUMBER(8,2),ADATE DATE);
Table created.

SQL> CREATE TABLE BRANCH (BNAME VARCHAR2(18),CITY VARCHAR2(18));
Table created.

SQL> CREATE TABLE CUSTOMERS (CNAME VARCHAR2(19),CITY VARCHAR2(18));
Table created.

SQL> CREATE TABLE BORROW(LOANNO VARCHAR2(5),CNAME VARCHAR2(18),BNAME
VARCHAR2(18),AMOUNT NUMBER(8,2));
Table created.

SQL> COMMIT;
Commit complete.

////////////////////////////////////*******INSERTING DATA INTO TABLES*******////////////////////////////////////

SQL> INSERT INTO DEPOSIT
2 VALUES('&ACTNO','&CNAME','&BNAME','&AMOUNT','&ADATE');

SQL> INSERT INTO BRANCH
2 VALUES('&BNAME','&CITY');

SQL> INSERT INTO CUSTOMERS
2 VALUES('&CNAME','&CITY');

SQL> INSERT INTO BORROW
2 VALUES('&LOANNO','&CNAME','&BNAME','&AMOUNT');

SQL> COMMIT;
Commit complete.

/////////////////////////////////*******QUERIES TO PERFORM ON TABLE*******/////////////////////////////////

(1) Describe deposit, branch.

SQL> DESC DEPOSIT;

Name	Null?	Type

ACTNO		VARCHAR2(5)
CNAME		VARCHAR2(18)
BNAME		VARCHAR2(18)
AMOUNT		NUMBER(8,2)
ADATE		DATE

SQL> DESC BRANCH;

Name	Null?	Type

BNAME		VARCHAR2(18)
CITY		VARCHAR2(18)

(2) Describe borrow, customers.

SQL> DESC BORROW;

Name	Null?	Type

LOANNO		VARCHAR2(5)
CNAME		VARCHAR2(18)
BNAME		VARCHAR2(18)
AMOUNT		NUMBER(8,2)

SQL> DESC CUSTOMERS;

Name	Null?	Type

CNAME		VARCHAR2(19)
CITY		VARCHAR2(18)

(3) List all data from table DEPOSIT.

SQL> SELECT * FROM DEPOSIT;

ACTNO	CNAME	BNAME	AMOUNT	A
DATE				
100	ANIL	VRCE	1000	
	01-MAR-95			
101	SUNIL	AJNI	5000	
	04-JAN-96			
102	MEHUL	KAROLBAGH	3500	
	17-NOV-95			
104	MADHURI	CHANDI	1200	
	17-NOV-95			
105	PRMOD	M.G.ROAD	3000	
	27-MAR-96			
106	SANDIP	ANDHERI	2000	
	31-MAR-96			
107	SHIVANI	VIRAR	1000	
	05-SEP-95			
108	KRANTI	NEHRU		
	PLACE	5000	02-JUL-95	
109	MINU	POWAI	7000	
	10-AUG-95			

9 rows selected.

(4) List all data from table BORROW.

SQL> SELECT * FROM BORROW;

LOANN	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000
206	MEHUL	AJNI	5000
311	SUNIL	DHARAMPETH	3000
321	MADHURI	ANDHERI	2000
375	PRMOD	VIRAR	8000
481	KRANTI	NEHRU PLACE	3000

-----6 rows selected.

(5) List all data from table CUSTOMERS.

SQL> SELECT * FROM CUSTOMERS;

CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

10 rows selected

(6) List all data from table BRANCH.

SQL> SELECT * FROM BRANCH;

BNAME	CITY
VRCH	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
CHANDI	DELHI
DHARAMPETH	NAGPUR
M.G.ROAD	BANGLORE
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

10 rows selected

(7) Give account no and amount of depositors.

SQL> SELECT ACTNO,AMOUNT
2 FROM DEPOSIT;

ACTNO	AMOUNT
100	1000
101	5000
102	3500
104	1200

105	3000
106	2000
107	1000
108	5000
109	7000

9 rows selected.

(8) Give name of depositors having amount greater than 4000.

```
SQL> SELECT CNAME
      2 FROM DEPOSIT
      3 WHERE AMOUNT>4000;
```

CNAME

SUNIL

KRANTI

MINU

(9) Give name of customers who opened account after date '1-12-96'.

```
SQL> SELECT ADATE
      2 FROM DEPOSIT
      3 WHERE ADATE>'1-DEC-96';
```

no rows selected

PRACTICAL 2

////////////////////////////////////*******CREATING TABLES FOR DATABASE*******////////////////////////////////////

```
SQL> create table job(job_id varchar2(15),job_title varchar2(30),min_sal
number(7,2),max_sal number(7,2));
```

Table created.

```
SQL> create table employee(emp_no number(3),emp_name varchar2(30),emp_sal
number(8,2),emp_comm number(6,1),dept_no number(3));
```

Table created.

```
SQL> create table deposit1(a_no varchar2(5),cname varchar2(15),bname
varchar2(10),amount number(7,2),a_date date);
```

Table created.

```
SQL> create table borrow1(loanno varchar2(5),cname varchar2(15),bname
varchar2(10),amount varchar2(7));
```

Table created.

```
SQL> commit;
```

Commit complete.

////////////////////////////////////*******INSERTING DATA INTO TABLES*******////////////////////////////////////

```
SQL> insert into employee
2 values(&emp_no,&emp_name,&emp_sal,&emp_comm,&dept_no);
```

```
SQL> insert into job
2 values('&job_id','&job_title',&min_sal,&max_sal);
```

```
SQL> insert into deposit1
2 values('&a_no','&cname','&bname',&amount,&date);
```

//////////////////////////////////*******QUERIES TO PERFORM ON TABLE*******\\

(2) Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.

SQL> select A_NO ,AMOUNT
2 FROM DEPOSIT
3 WHERE A_DATE BETWEEN'01-JAN-06'and'25-JUL-06';

A_NO	AMOUNT

101	7000
102	5000
103	6500

(3) Display all jobs with minimum salary is greater than 4000.

SQL> select job_title,min_sal
2 from job
3 where min_sal>4000;

JOB_TITLE	MIN_SAL

Marketing manager	9000
Finance manager	8200
Account	4200
Lecturer	6000

(4) Display name and salary of employee whose department no is 20. Give alias name to name of employee.

SQL> select emp_name"name of employee",emp_sal
2 from employee
3 where dept_no=20;

name of employee	EMP_SAL

Smith	800
Adama	1100

(5) Display employee no,name and department details of those employee whose department lies in(10,20)

SQL> select emp_no,emp_name,dept_no
2 from employee
3 where dept_no between 10 and 20 ;

<i>EMP_NO</i>	<i>EMP_NAME</i>	<i>DEPT_NO</i>
101	Smith	20
103	Adama	20
104	Aman	15
105	Anita	10
106	Sneha	10

To study various options of LIKE predicate

(1) Display all employee whose name start with 'A' and third character is 'a'.

```
SQL> select *
2 from employee
3 where emp_name like'A_a%';
```

<i>EMP_NO</i>	<i>EMP_NAME</i>	<i>EMP_SAL</i>	<i>EMP_COMM</i>	<i>DEPT_NO</i>
103	Adama	1100	0	20
104	Aman	3000	15	
107	Anamika	1975	30	

(2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.

```
SQL> select emp_name,emp_no,emp_sal
2 from employee
3 where emp_name like'Ani__';
```

<i>EMP_NAME</i>	<i>EMP_NO</i>	<i>EMP_SAL</i>
Anita	105	5000

(3) Display the non-null values of employees and also employee name second character should be 'n' and string should be 5 character long.

```
SQL> select *
2 from employee
3 where emp_comm is not null and emp_name like '_n__';
```

<i>EMP_NO</i>	<i>EMP_NAME</i>	<i>EMP_SAL</i>	<i>EMP_COMM</i>	<i>DEPT_NO</i>
105	Anita	5000	50000	10
106	Sneha	2450	24500	10

(4) Display the null values of employee and also employee name's third character should be 'a'.

```
SQL> select *  
2 from employee  
3 where emp_comm is null and emp_name like '__a%';
```

<i>EMP_NO</i>	<i>EMP_NAME</i>	<i>EMP_SAL</i>	<i>EMP_COMM</i>	<i>DEPT_NO</i>
104	Aman	3000	15	
107	Anamika	1975	30	

(5) What will be output if you are giving LIKE predicate as '%_%' ESCAPE '\'

```
SQL> select *  
2 from employee  
3 where emp_name like '%\_%'ESCAPE'\';
```

no rows selected

PRACTICAL 3

To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

(1) List total deposit from deposit.

SQL> select sum(amount) from deposit;

SUM(AMOUNT)

28700

(2) List total loan from karolbagh branch

SQL> select sum(amount) from deposit where BNAME='KAROLBAGH';

SUM(AMOUNT)

3500

(3) Give maximum loan from branch vrce.

SQL> select max(amount) from deposit where BNAME='VRCE';

MAX(AMOUNT)

1000

(4) Count total number of customers

SQL> select count(CNAME) from customers;

COUNT(CNAME)

10

(5) Count total number of customer's cities.

SQL> select count(distinct CITY) from customers;

COUNT(DISTINCTCITY)

7

(6) Create table supplier from employee with all the columns.

SQL> create table supplier as select * from EMPLOYEE;

Table created

SQL> desc supplier

Name	Null?	Type
EMP_NO		NUMBER(3)
EMP_NAME		VARCHAR2(30)
EMP_SAL		NUMBER(8,2)
EMP_COMM		NUMBER(6,1)
DEPT_NO		NUMBER(3)

(7) Create table sup1 from employee with first two columns.

SQL> create table sup1 as select EMP_NO,EMP_NAME from EMPLOYEE;

Table created.

SQL> desc sup1

Name	Null?	Type
EMP_NO		NUMBER(3)
EMP_NAME		VARCHAR2(30)

(8) Create table sup2 from employee with no data.

SQL> create table sup2 as select * from EMPLOYEE where 1=2;

Table created.

SQL> desc sup2

Name	Null?	Type
EMP_NO		NUMBER(3)
EMP_NAME		VARCHAR2(30)
EMP_SAL		NUMBER(8,2)
EMP_COMM		NUMBER(6,1)
DEPT_NO		NUMBER(3)

(9) Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field.

SQL> insert into sup2 (select * from employee where emp_name like '_n_____');

2 rows created.

(10) Delete all the rows from sup1.

SQL> delete from sup1;

0 rows deleted.

(11) Delete the detail of supplier whose sup_no is 103.

SQL> delete from supplier where EMP_NO=103;

1 row deleted.

(12) Rename the table sup2.

SQL> rename sup2 to sup3;

Table renamed.

SQL> select * from sup3;

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO
105	Anita	5000	50000	10
106	Sneha	2450	24500	10

(13) Destroy table sup1 with all the data.

SQL> truncate table sup1;

Table truncated.

(14) Update the value dept_no to 10 where second character of emp. name is 'm'.

SQL> update employee set dept_no=10 where emp_name like '_m%';

2 rows updated.

(15) Update the value of employee name whose employee number is 103.

SQL> update employee set emp_name='Lakshman' where emp_no=103;

1 row updated.

SQL> select * from employee;

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO
--------	----------	---------	----------	---------

101 Smith	800		10
102 Snehal	1600	300	25
103 Lakshman	1100	0	20
104 Aman	3000		10
105 Anita	5000	50000	10
106 Sneha	2450	24500	10
107 Anamika	1975		30

7 rows selected.

PRECTICAL 4

4. To study Single-row functions.

(1) Write a query to display the current date. Label the column Date

```
SQL> SELECT sysdate "Date"
       2 FROM dual;
```

Date

24-NOV-16

(2) For each employee, display the employee number, job, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary

```
SQL> SELECT EMP_NO, EMP_NAME, EMP_SAL,
       2 ROUND(EMP_SAL * 1.15, 0) "New_Salary"
       3 FROM employee;
```

EMP_NO	EMP_NAME	EMP_SAL	New_Salary
101	SMITH	800	920
102	SNEHAL	1600	1840
103	ADAMA	1100	1265
104	AMAN	3000	3450
105	ANITA	5000	5750
106	SNEHA	2450	2818
107	ANAMIKA	2975	3421

(3) Modify your query no 4.(2) to add a column that subtracts the old salary from the new salary. Label the column Increase

```
SQL> SELECT EMP_NO, EMP_NAME, EMP_SAL,
       2 ROUND(EMP_SAL * 1.15, 0) "New_Salary"
       3 ROUND(EMP_SAL * 1.15, 0) - NEW_SALARY "INCREASE"
       4 FROM EMPLOYEE;
```

EMP_NO	EMP_NAME	EMP_SAL	New_Salary
101	SMITH	800	920
102	SNEHAL	1600	1840
103	ADAMA	1100	1265
104	AMAN	3000	3450

105 ANITA	5000	5750
106 SNEHA	2450	2818
107 ANAMIKA	2975	3421

(4) Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

```
SQL> SELECT INITCAP(EMP_NAME) "Name",
2 LENGTH(EMP_NAME) "Length"
3 FROM employee
4 WHERE EMP_NAME LIKE 'J%'
5 OR EMP_NAME LIKE 'M%'
6 OR EMP_NAME LIKE 'A%'
7 ORDER BY EMP_NAME;
```

Name	Length
Adama	5
Aman	4
Anamika	7
Anita	5

(5) Write a query that produces the following for each employee:
<employee last name> earns <salary> monthly

```
SQL> SELECT last_name || ' earns '
|| TO_CHAR(salary, 'fm$99,999.00')
|| ' monthly but wants '
|| TO_CHAR(salary * 3, 'fm$99,999.00')
|| '. ' "Dream Salaries"
FROM employees;
```

PRACTICAL 5

5. Displaying data from Multiple Tables (join)

(1) Give details of customers ANIL.

```
SQL> select d.actno,d.cname,d.amount,d.adate,c.city,b.city,d.bname
2 from deposit d,customers c,branch b
3 where d.cname=c.cname and d.bname=b.bname and d.cname= 'anil';
```

```
actno bname cname amount adate      city    city
100   vice   anil   1000   01-mar-95 calcutta nagpur
*****
```

(2) Give name of customer who are borrowers and depositors and having living city nagpur

```
SQL> select c.cname
2 from customers.c,deposit.d,borrow.b
3 where c.city = 'nagpur' and c.cname = d.cname and c.cname = b.cname;
```

```
cname
Madhuri
Pramod
*****
```

(3) Give city as their city name of customers having same living branch.

```
SQL> select c.city
2 from customers.c,branch.b,
3 where c.city= b.city;
```

```
Output
City
Nagpur
Delhi
Bombay
*****
```

(4) Write a query to display the last name, department number, and department name for all employees.

```
SQL> select e.emp_name,e.dept_no,d.dept_name
2 from employee.e,dept.d
3 where e.dept_no = d.dept_no;
```

```
emp_name dept_no dept_name
smith    20      ce
snehal   25      me
adama    20      ce
```

<i>anita</i>	<i>10</i>	<i>it</i>
<i>sneha</i>	<i>10</i>	<i>it</i>
<i>anamika</i>	<i>30</i>	<i>civil</i>

->Department Table

<i>dept_no</i>	<i>dept_name</i>	<i>dept_loc</i>
<i>20</i>	<i>ce</i>	<i>newyork</i>
<i>10</i>	<i>it</i>	<i>baroda</i>
<i>30</i>	<i>civil</i>	<i>newyork</i>
<i>35</i>	<i>ec</i>	<i>ahmedabad</i>
<i>25</i>	<i>me</i>	<i>surat</i>
<i>45</i>	<i>me</i>	<i>baroda</i>

(5) Create a unique listing of all jobs that are in department 30. Include the location of the department in the output

```
SQL> select j.job_id,j.job_name,e.depy_no,d.dept_no
2 from jobj,employee.e,dept.d
3 wheremj.job_id = e.job_id and e.dept_no = d.dept_no and e.dept_no = 30;
```

<i>job_id</i>	<i>job_name</i>	<i>dept_no</i>	<i>dept_loc</i>
<i>lec</i>	<i>lecturer</i>	<i>30</i>	<i>newyork</i>

(6) Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK.

```
SQL> select e.emp_name,d.dept_name,e.dept_no
2 from employee.e,dept.d
3 where e.dept_no=d.dept_no and d.dept_loc='newyork';
```

<i>emp_name</i>	<i>dept_no</i>	<i>dept_name</i>
<i>smith</i>	<i>20</i>	<i>ce</i>
<i>adama</i>	<i>20</i>	<i>ce</i>
<i>anamika</i>	<i>20</i>	<i>civil</i>

PRACTICAL 6

6. To apply the concept of Aggregating Data using Group functions.

(1) List total deposit of customer having account date after 1-jan-96.

```
SQL> select sum(amount)
2  from deposit
3  where adate > '1-jan-96';
```

SUM(AMOUNT)

```
-----
      10000
*****
```

(2) List total deposit of customers living in city Nagpur.

```
SQL> select sum(d.amount)
2  from deposit.d,customers.c
3  where d.cname = c.cname and c.city = 'nagpur';
```

SUM(AMOUNT)

```
-----
      4200
*****
```

(3) List maximum deposit of customers living in bombay.

```
SQL> select max(d.amount)
2  from deposit d,customer c
3  where d.cname = c.cname and city = 'bombay'
```

MAX(D.AMOUNT)

```
-----
      5000
*****
```

(4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

```
SQL> select max(emp_sal)"maximun",sum(emp_sal)"sum",min(emp_sal)"minimun"
2  round(avg(emp_sal)"average")
3  from employee;
```

<i>maximum</i>	<i>minimum</i>	<i>sum</i>	<i>average</i>
5000	800	16925	2418

(5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

```
SQL> select max(emp_sal)-min(emp_sal)"difference"
2 from employee;
```

Difference

```
-----
4200
```

(6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998

```
SQL> select count(emp_name)
2 from employee
3 where hire_date like '%95' or hire_date like '%96' or hire_date like '%97' or hire_date
like '%98';
```

Count(emp_name)

```
-----
3
```

(7) Find the average salaries for each department without displaying the respective department numbers.

```
SQL> select avg(emp_sal)
2 from employee
3 group by dept_no;
```

AVG(EMP_SAL)

```
-----
1600
2975
800
2050
5000
2450
```

(8) Write a query to display the total salary being paid to each job title, within each department.

```
SQL> select dept_no,sum(emp_sal)
2 from employee
3 group by dept_no;
```

```
DEPT_NO  SUM(EMP_SAL)
-----  -
```

25	1600
30	2975
20	800
15	4100
10	5000
12	2450

(9) Find the average salaries > 2000 for each department without displaying the respective department numbers.

```
SQL> select dept_no,avg(emp_sal)
2 from employee
3 group by dept_no having avg(emp_sal)>2000;
```

DEPT_NO AVG(EMP_SAL)

```
-----
30      2975
15      2050
10      5000
12      2450
```

(10) Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary.

```
SQL> select dept_no,sum(emp_sal)
2 from employee
3 group by having sum(emp_sal)>3000
4 order by (emp-sal);
```

DEPT_NO SUM(EMP_SAL)

```
-----
30      3300
10      7450
```

(11) List the branches having sum of deposit more than 5000 and located in city bombay.

```
SQL> select d.bname,sum(d.amount)
2 from deposit.d branch.b
3 where b.bname=d.bname and b.city='bombay'
4 group by d.bname having sum (d.amount)>5000;
```

BNAME SUM(D.AMOUNT)
POWALI 7000

PRACTICAL 7

7. To solve queries using the concept of sub query.

(1) Write a query to display the last name and hire date of any employee in the same department as SCOTT. Exclude SCOTT

```
SQL>select emp_name, hire_date
      From employee
      Where dept_no in (select dept_no from employee where emp_name like
'SCOTT') and
      emp_name < > 'SCOTT' ;
```

Emp_name	hire_date
Adama	1-JAN-96

(2) Give name of customers who are depositors having same branch city of mr. sunil.

```
SQL> select d1.cname , d1.bname
      From deposit d1 , branch b1
      Where b1.city in (select b2.city from deposit d2 , branch b2 where d2.cname='sunil') and
      d1.bname=b1.bname
```

cname	bname
Sunil	Ajni
Anil	Vrce

(3) Give deposit details and loan details of customer in same city where pramod is living.

```
SQL>select d1.actno, d1.bname , d1.amount,d1.adate, b1.loan no, b1.bname, b1.amount
      From deposit d1 , borrow b1 , customers c1
      Where c1.cname=d1.cname and d1.cname=b1.cname and c1.city in (select c2.city from
customer .c2 where c2.cname='prmod');
```

bname	actno	amount	adate	loanno	bname	amount
Chandi		104	1200	17-DEC-05	321	Andheri
M.G.Road	105	3000	27-MAR-96	375	virar	8000

(4) Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.

```
SQL>select emp_no, emp_name, emp_sal,
      From employee
      Where emp_sal >(select avg(emp_sal) from employee ) order by emp_sal
```

emp_name	emp_no	emp_sal
Sneha	106	2450
Anamika	107	2975
Aman	104	3000
Anita	105	5000

(5) Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000

SQL>select c.dcname , c.amount , c.city , c.bname
From customer c.deposit
Where c.amount > 2000 and c.dcname = c.name and c.city in (select city from customer
Where c.dcname = 'shivani')0;

Cname	amount	city	bname
Kranti	5000	Bombay	Nehrurplace

(6) Display the last name and salary of every employee who reports to ford.

SQL>select em.manager_name , e.emp_sal
From employee e , emp_manager em
Where e.emp_no = emp.emp_no and manager_name = 'Rakesh';

Emp_name	emp_sal
Anamika	2975

(7) Display the department number, name, and job for every employee in the Accounting department.

SQL>select e.job_id , e.dept_no , d.dept_name
From department d , employee e
Where d.dept_no = e.dept_no and d.dept_name='CE';

Dept_no	dept_name	job_id
20	CE	
20	CE	

(8) List the name of branch having highest number of depositors.

SQL>select bname

From deposit d
Group by bname having count (bname) >= all (select count) (cname from deposit d group by
bname);

bname

Andheri

(9) Give the name of cities where in which the maximum numbers of branches are located.

SQL>select city
from branch
group by city having count (banme) >= all(select count (bname) from branch group by city);

city

Nagpur

Delhi

Bombay

PRACTICAL 8

8:- To apply the concept of Cursor

1.

SQL> Declare

v_eno employees.employee_id%type;

v_sal employees.salary%type;

Cursor emp_cur is

Select employee_id,salary

From employees

Where department_id=90;

Begin

Open emp_cur;

loop

Fetch emp_cur into v_eno,v_sal;

DBMS_OUTPUT.PUT_LINE(to_char(v_eno)|| ' '||to_char(v_sal));

Exit when emp_cur%NOTFOUND;

end loop;

Close emp_cur;

End;

2.

SQL> Declare

2 v_eno employees.employee_id%type;

3 v_sal employees.salary%type;

4 Cursor emp_cur is

5 Select employee_id,salary

6 From employees

7 ;

8 Begin

9 Open emp_cur;

10 loop

```
11 Fetch emp_cur into v_eno,v_sal;
12 dbms_output.put_line(to_char(v_eno)||' '||to_char(v_sal));
13 exit when emp_cur%NOTFOUND;
14 end loop;
15 end;
16
```

Declare

```
v_eno employees.employee_id%type;
v_sal employees.salary%type;
Cursor emp_cur is
Select employee_id,salary
From employees
;
Begin
Open emp_cur;
loop
Fetch emp_cur into v_eno,v_sal;
dbms_output.put_line(to_char(v_eno)||' '||to_char(v_sal));
exit when emp_cur%ROWCOUNT >10;
end loop;
end;
```

OUTPUT

```
100 24000
101 17000
102 17000
103 9000
104 6000
105 4800
106 4800
107 4200
```

108 12000
109 9000
110 8200

PL/SQL procedure successfully completed.

PRACTICAL 9

9:- To apply the concept of procedure

```
SQL> Create or replace procedure  
show_sal(v_id employees.employee_id%type)  
is  
    v_sal employees.salary%type;  
begin  
    select salary  
    into v_sal  
    from employees  
    where employee_id=v_id;  
    dbms_output.put_line(v_sal);  
end show_sal;  
/
```

Procedure created.

```
SQL> call show_sal(104);  
6000  
Call completed.
```

```
SQL> call show_sal(108);  
12000  
Call completed.
```

PRACTICAL 10

11 :- To apply the concept of Function

```
SQL> Create or replace function
get_sal(v_id employees.employee_id%type)
Return number
Is
v_sal employees.salary%type;
begin
select salary
into v_sal
from employees
where employee_id=v_id;
return v_sal;
end get_sal;
/
Function created.
SQL> variable sal number;
SQL> execute:sal:= get_sal(101);
PL/SQL procedure successfully completed.
```

```
SQL> print sal;
```

SAL

17000
