Ex. No : 6

CO2:Construct queries using SQL for database creation, interaction, modification, and updation. (Cognitive Knowledge Level: Apply)

Implementation of built-in functions in RDBMS

AIM

RDBMS Built in Functions

There are two types of functions:

1) Single Row Functions: Single row or Scalar functions return a value for every row that is processed in a query.

2) Group Functions: These functions group the rows of data based on the values returned by the query. This is discussed in SQL GROUP Functions. The group functions are used to calculate aggregate values like total or average, which return just one total or one average value after processing a group of rows.

There are four types of single row functions. They are:

1) Numeric Functions: These are functions that accept numeric input and return numeric values.

2) Character or Text Functions: These are functions that accept character input and can return both character and number values.

3) Date Functions: These are functions that take values that are of datatype DATE as input and return values of datatype DATE, except for the MONTHS_BETWEEN function, which returns a number.

4) Conversion Functions: These are functions that help us to convert a value in one form to another form. For Example: a null value into an actual value, or a value from one datatype to another datatype like NVL, TO_CHAR, TO_NUMBER, TO_DATE etc.

Mathematical Functions

```
SQL> select ABS(-100) from dual;
ABS(-100)
----------
100
SQL> select ABS(-6) from dual;
```

```
ABS(-6)
----------
6
SQL> select FLOOR(2345.78) FROM DUAL;
FLOOR(2345.78)
--------------
 2345
SQL> SELECT GREATEST(23,67,90,123,78,50) FROM DUAL;
GREATEST(23,67,90,123,78,50)
----------------------------
123
SQL> SELECT LEAST(34, 21,67,11,89,9) FROM DUAL;
LEAST(34,21,67,11,89,9)
 9
SQL> SELECT LENGTH('RAJESHWARI') FROM DUAL;
LENGTH('RAJESHWARI')
--------------------
10
SQL> SELECT LENGTH(17245637) FROM DUAL;
LENGTH(17245637)
----------------
8
SQL> SELECT SQRT(16) FROM DUAL;
SQRT(16)
----------
4
SQL> SELECT SQRT(99) FROM DUAL;
SQRT(99)
 9.94987437
SQL> SELECT POWER(2,4) FROM DUAL;
POWER(2,4)
----------
16
SQL> SELECT POWER(2,10) FROM DUAL;
POWER(2,10)
-----------
1024
SQL> SELECT power(2,10) FROM DUAL;
POWER(2,10)
-----------
1024
SQL> SELECT ROUND(5.86) FROM DUAL;
ROUND(5.86)
-----------
6
SQL> SELECT ROUND(1001.6) FROM DUAL;
ROUND(1001.6)
```

```
-------------
1002
SQL> SELECT ROUND(1001.3) FROM DUAL;
ROUND(1001.3)
-------------
1001
SQL> SELECT SIN(90) FROM DUAL;
SIN(90)
----------
.893996664
SQL> SELECT COS(45) FROM DUAL;
COS(45)
----------
.525321989
SQL> SELECT TAN(30) FROM DUAL;
TAN(30)
----------
-6.4053312
SQL> SELECT TAN(90) FROM DUAL;
TAN(90)
----------
-1.9952004
SQL> SELECT TAN(180) FROM DUAL;
TAN(180)
----------
1.33869021
SQL> SELECT SIGN(-128) FROM DUAL;
SIGN(-128)
----------
-1
SQL> SELECT SIGN(10) FROM DUAL;
SIGN(10)
----------
1
SQL> SELECT SIGN(0) FROM DUAL;
SIGN(0)
----------
0
SQL> SELECT LN(100) FROM DUAL;
LN(100)
----------
4.60517019
SQL> SELECT LN(10) FROM DUAL;
LN(10)
----------
2.30258509
SQL> SELECT LOG(10,100) FROM DUAL;
```

```
LOG(10,100)
-----------
2
SQL> SELECT LOG(100,10) FROM DUAL;
LOG(100,10)
-----------
.5
SQL> SELECT MOD(4,3) FROM DUAL;
MOD(4,3)
----------
1
SQL> SELECT MOD(4,2) FROM DUAL;
MOD(4,2)
----------
0
SQL> SELECT EXP(2) FROM DUAL;
EXP(2)
----------
7.3890561
SQL> SELECT EXP(-2) FROM DUAL;
EXP(-2)
----------
.135335283
SQL> SELECT EXP(0) FROM DUAL;
EXP(0)
----------
1
```

Date Functions

```
SQL> SELECT CURRENT_DATE FROM DUAL;
CURRENT_D
---------
14-AUG-19
SQL> SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;
EXTRACT(YEARFROMSYSDATE)
------------------------
2019
SQL> SELECT EXTRACT(DAY FROM SYSDATE) FROM DUAL;
EXTRACT(DAYFROMSYSDATE)
-----------------------
14
SQL> SELECT EXTRACT(MONTH FROM SYSDATE) FROM DUAL;
EXTRACT(MONTHFROMSYSDATE)
-------------------------
8
```

```
SQL> SELECT SYSDATE FROM DUAL;
SYSDATE
---------
AUG-19
```

String Functions

```
SQL> select ascii('t') from dual;
ASCII('T')
----------
116
SQL> select ascii('a') from dual;
ASCII('A')
----------
97
SQL> select ascii('A') from dual;
ASCII('A')
----------
65
SQL> select ascii('Z') from dual;
ASCII('Z')
----------
90
SQL> select ascii('z') from dual;
ASCII('Z')
----------
122
SQL> SELECT UPPER('bldea sb arts and kcp science college') from dual;
UPPER('BLDEASBARTSANDKCPSCIENCECOLLEG’)
-------------------------------------
BLDEA SB ARTS AND KCP SCIENCE COLLEGE
SQL> select LOWER('welcome to dbms lab') from dual;
LOWER('WELCOMETODBM
-------------------
welcome to dbms lab
SQL> select LOWER('WELCOME TO DBMSLAB') from dual;
LOWER('WELCOMETODB
------------------
welcome to dbmslab
SQL> SELECT REPLACE('HELLO','H','K') FROM DUAL;
REPLA
-----
KELLO
SQL> SELECT REPLACE('COMPUTER','C','K') FROM DUAL;
REPLACE( --------
KOMPUTER
SQL> SELECT REPLACE('HELLO','L','A') FROM DUAL;
```

```
REPLA
-----
HEAAO
SQL> SELECT TRIM('A' FROM 'ANACONDA') FROM DUAL;
TRIM('
--
NACOND
SQL> SELECT LTRIM('ANACONDA','A') FROM DUAL;
LTRIM('
-------
NACONDA
SQL> SELECT LTRIM('ANIL','A') FROM DUAL;
LTR
---
NIL
SQL> SELECT RTRIM('ANITA','A') FROM DUAL;
RTRI
---
ANIT
SQL> SELECT RTRIM('ANACONDA','A') FROM DUAL;
RTRIM('
-------
ANACOND
SQL> SELECT RTRIM('ANACONDA ','A') FROM DUAL;
RTRIM('ANAC
-----------
ANACONDA
```

Ex. No : 7

Implementation of various aggregate functions in SQL

CO2: Construct queries using SQL for database creation, interaction, modification, and updation. (Cognitive Knowledge Level: Apply)

AIM

Create the tables with the following fields

Faculty (FacultyCode, FacultyName)
Subject (SubjectCode,SubjectName,MaxMark,FacultyCode)
Student(StudentCode,StudentName,DOB,StudentsBranch(CS/EC/EE/ME), AdmissionDate)
M_Mark (StudentCode, SubjectCode, Mark)

Do the following queries

a) Display the number of faculties.

b) Display the total mark for each student.

c) Display the subject,average mark for each subject.

d) Display the name of subjects for which atleast one student got below 40%.

e) Display the name,subject and percentage of mark who got below 40 %.

f) Display the faculties and alloted subjects for each faculty

g) Display the name of faculties who take more than one subject.

h) Display name,subject,mark, % of mark in ascending order of mark

Commands

```
 Create Table Faculty (F_Code Number Primary Key, F_Name Varchar(15));
          insert into Faculty values(&facultycode,'&facultyname');
```

```
          SELECT * FROM Faculty;

                    F_CODE              F_NAME
                    -----------         ---------------------
                          105           Jayakumar
                          104           Sangeetha
```

```
            102                 Bindu
            101                 Silgy
             103                Vidhya
```

```
create table Subject (subjectcode varchar(5) primary key not null,subjectname
char(15),maxmark number(5,2),faculty_code int,foreign key(faculty_code) references
Faculty(f_code));
```

```
insert into Subject values('&subjectcode','&subjectname',&maxmark,&facultycode);
SUBJECTCODE     SUBJECTNAME     MAXMARK FACULTYCODE
-------------------    ------------------------    -------------------   -----------------------

503                           DBMS              100             105
501                           Maths            150               101
502                           FSA              100             102
504                           OS                75               103
505                            DC              200             104
508                            DBMS lab        1001              103
```

```
create table Student(studentcode varchar(5) primary key not null,studentname
char(15),dob date,studentbranch char(3),adate date,check(studentbranch
in('cs','ec','ee','me')));
```

```
insert into Student values('&studentcode','&studentname','&dob','&studentbranch','&adate');

Enter value for studentcode: 1
Enter value for studentname: Amitha
Enter value for dob: 12-jan-1987
Enter value for studentbranch: cs
Enter value for adate: 1-jun-2000
old   1: insert into Student
values('&studentcode','&studentname','&dob','&studentbranch','&adate')
new   1: insert into Student values('1','Amitha','12-jan-1987','cs','1-jun-2000')
insert into student values(2,'vaidehi','25-dec-88','me','1-jun-2000');
insert into student values(3,'varun','2-oct-88','me','2-jun-2000');
insert into student values(4,'turner','5-sep-88','ec','1-jun-2000');
insert into student values(5,'vani','20-jul-88','ee','5-jun-2000');
insert into student values(6,'binu','13-aug-88','me','10-jun-2000');
insert into student values(7,'chitra','14-nov-86','me','9-jun-1999');
insert into student values(8,'dona','2-dec-91','cs','2-jun-2000');
insert into student values(9,'elana','5-feb-90','cs','1-jun-2000');
insert into student values(10,'fahan','20-mar-88','ec','5-jun-2000');
insert into student values(11,'ginu','13-apr-88','ec','10-jun-2000');
insert into student values(12,'hamna','14-may-85','ee','9-jun-1999');
```

```
create table M_mark(studentcode varchar(5) references
Student(studentcode),subjectcode varchar(5) references Subject(subjectcode),mark
number(5,2),primary key(studentcode,subjectcode));
```

```
insert into M_mark values('&studentcode','&subjectcode',&mark);
```

```
insert into M_mark values(1,501,40);
insert into M_mark values(1,502,70);
insert into M_mark values(1,503,50);
insert into M_mark values(1,504,80);
insert into M_mark values(1,505,40);
insert into M_mark values(1,508,70);
insert into M_mark values(2,501,90);
insert into M_mark values(2,502,89);
insert into M_mark values(2,503,77);
insert into M_mark values(2,504,95);
insert into M_mark values(2,505,74);
insert into M_mark values(2,508,98);
insert into M_mark values(3,501,40);
insert into M_mark values(3,502,43);
insert into M_mark values(3,503,40);
insert into M_mark values(3,504,40);
insert into M_mark values(3,505,40);
insert into M_mark values(3,508,35);
insert into M_mark values(4,501,50);
insert into M_mark values(5,501,60);
insert into M_mark values(6,501,67);
insert into M_mark values(7,501,23);
insert into M_mark values(8,501,43);
insert into M_mark values(9,501,42);
insert into M_mark values(10,505,74);
insert into M_mark values(11,508,98);
insert into M_mark values(12,501,40);
insert into M_mark values(5,502,43);
insert into M_mark values(6,503,40);
insert into M_mark values(7,504,40);
insert into M_mark values(8,505,40);
insert into M_mark values(9,508,35);
insert into M_mark values(10,501,50);
insert into M_mark values(11,501,60);
insert into M_mark values(12,503,67);
insert into M_mark values(5,504,23);
insert into M_mark values(6,504,23);
insert into M_mark values(9,504,1);
insert into M_mark values(10,504,1);
insert into M_mark values(6,502,43);
insert into M_mark values(7,505,42);
```

a) Display the number of faculties.

```
select count(*) “No: of Faculties” from faculty;
```

```
No: of Faculties
----------------
       5
```

b) Display the total mark for each student.

```
select studentname,sum(mark) “Total Mark” from M_mark,Student where
Student.studentcode= M_mark.studentcode group by studentname;
```

```
STUDENTNAME      SUM(MARK)
---------------  ----------
binu             150
hamna            107
turner           50
fahan            124
vaidehi          523
chitra           105
Amitha           350
ginu             158
varun            238
vani             126
dona              83
elana            77
```

c) Display the subject,average mark for each subject.

```
select subjectname,round(avg(mark),2) “Average mark” from Subject,M_mark where
Subject.subjectcode= M_mark.subjectcode group by subjectname;
```

```
SUBJECTNAME      Average mark
---------------  ------------
DBMS lab         67.2
DC                51.67
FSA              57.6
DBMS             54.8
Maths            50.42
OS               55.6
```

d) Display the name of subjects for which atleast one student got below 40%.

```
select subject.subjectname,count(student1.studentname)”NO: OF STUDENTS” from
subject,m_mark,student1 where student1.studentcode= m_mark.studentcode and
m_mark.mark<(40* maxmark)/100 and subject.SubjectCode=m_mark.Subjectcode
group by subject. Subjectname having count(distinct(m_mark.subjectcode))>=1;
```

```
SUBJECTNAME      NO: OF STUDENTS
---------------  -------------------------
DBMS lab            2
```

```
Maths                    1
OS                       4
```

e) Display the name,subject and percentage of mark who got below 40 %.

select **studentname**,
subjectname,mark,maxmark,round((m_mark.mark/maxmark)*100,2)"Percentage"
from subject, student1, m_mark where mark<(40*maxmark/100) and subject.
SubjectCode = m_mark. subjectcode and student1.**studentcode**
=m_mark.studentcode;

f) Display the faculties and alloted subjects for each faculty.

select Faculty.f_name,Subject.subjectname from Faculty,Subject where
Faculty.F_code=Subject.FACULTYCODE;

```
F_NAME              SUBJECTNAME
---------------     ---------------
Vidhya              DBMS lab
Jayakumar           DBMS
Silgy               Maths
Bindu               FSA
Vidhya              OS
Sangeetha           DC
```

g) Display the name of   faculties who take more than one subject.

Select f_name name from Faculty where (select count(subjectcode) from Subject
where Subject.facultycode=Faculty.f_code)>1 group by Faculty.f_name;

or

select Faculty.f_name,count(subject.SubjectCode) “NO OF SUBJECTS” from
Faculty,subject where (select count(*) from Subject where
Subject.facultycode=Faculty.f_code)>1 and Subject.facultycode=Faculty.f_code
group by Faculty.f_name;

```
F_NAME         NO OF SUBJECTS

--------------    ----------------------

Vidhya                  2
```

h) Display name,subject,mark, % of mark in ascending order of mark

select studentname,subjectname,mark from Student1,Subject,M_mark where
Student1.studentcode=M_mark.studentcode and Subject.subjectcode=
M_mark.subjectcode order by mark;

Ex. No : 8

Implementation of Order By, Group By & Having clause

CO2: Construct queries using SQL for database creation, interaction, modification, and updation. (Cognitive Knowledge Level: Apply)

AIM

Create two tables

Dept(Department_Id, Department_Name , Manager_id, Loc)

Emp(Emp_no , Emp_name,Job , Salary , Hiredate,Comm , Depno)

MANAGER_ID is the empno of the employee whom the employee reports to. DEPTNO is a foreign key.Insert these values into department table

1) Display the name and salary for all employees whose salary is not in the range of 5000 and 35000

2) Display the employee name, job ID, and start date of employees hired between February 20, 1990, and May 1, 1998. Order the query in ascending order by start date.

3) list the name and salary of employees who earn between 5,000 and12,000, and are in department 2 or 4. Label the columns Employee and Monthly Salary,respectively.

4)Display the name and hire date of every employee who was hired in 1994.

5). Display the name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

6) Display the name and job title of all employees who do not have a manager.

7). Display the names of all employees where the third letter of the name is an *a*.

8). Display the name of all employees who have an *a* and an *e* in their name.
9). Display the name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to 2,0000, 4000, or 7,000.

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

11)For each employee, display the employee's name, and calculate the number of months between today and the date the employee was hired and years worked. Label the column

MONTHS_WORKED. Order your results by the number of months employed. Round the number of months and year up to the closest whole number.

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

13) Create a query to display the name and hire date of any employee hired after employee Mathew

14) Display the names and hire dates for all employees who were hired before their managers, along with their manager’s names and hire dates. Label the columns Employee, EmpHired, Manager, and Mgr Hired, respectively.

15) Write a query to display the number of people with the same job.
16). Display the manager number and the salary of the lowest paid employee for that manager.Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is less than 6,000. Sort the output in descending order of salary.

17. Write a query to display each department’s name, location, number of employees, and the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.

18). Write a query to display the name and hire date of any employee in the same department as amit. Exclude JOHN.

19. Write a query that displays the employee numbers names of all employees who work in a department with any employee whose name contains a *u*.

20)display employee name and department name of all employees that work in a department that has at least 3 employees. Order the list in alphabetical order first by department name, then by employee name.

21. Write a query to list the length of service of the employees (of the form n years and m months).

COMMANDS

```
CREATE TABLE dept(department_id    int primary key , department_name VARCHAR(20) NOT NULL , manager_id int, loc varchar(10));
```

```
create table emp(EMP_no int Primary Key,Emp_Name Varchar(10),Job Varchar(10),Hiredate Date,Salary Float,Comm Float,Depno Int References Dept(Department_Id));
```

```
INSERT INTO emp VALUES(1,'Steven', ‘Marketing’,’06-jan-1995’,24000, NULL,2);
INSERT INTO emp VALUES(2,'Neena', 'FI_ACCOUNT', '06-feb-1987’,34000, NULL,1);
INSERT INTO emp VALUES(3,'Lex', ‘FI_MGR', '06-jan-1980’,240000, NULL,1);
INSERT INTO emp VALUES(4,'Alexander', 'Sa_Rep', '06-jun-1987’,20000, NULL,4);
INSERT INTO emp VALUES(5,'Bruce', 'IT_PROG', '06-jul-1990’,24000, NULL,4);
INSERT INTO emp VALUES(6,'David', 'IT_PROG', '06-sep-1991’,22000, NULL,4);
INSERT INTO emp VALUES(7,'vipin', 'IT_PROG', '16-nov-1987’,28000, NULL,4);
INSERT INTO emp VALUES(8,'Diana', 'Pur_Man', '26-jan-1987’,24000, NULL,3);
INSERT INTO emp VALUES(9,'John', 'FI_ACCOUNT', '1-dec-1992’, 24000, NULL,1);
INSERT INTO emp VALUES(10,'Ismael', ‘CLERK’, '29-mar-1994’, 4000, NULL,3);
INSERT INTO emp VALUES(11,'Mathew', ‘CLERK’, '12-oct-1992’, 46000, 200,3);
INSERT INTO emp VALUES(12,'Hayes', ‘Marketing’, '21-apr-1998’,14000, 1000,2);
INSERT INTO emp VALUES(13,'sarun', ‘Marketing’, '18-may-1993’,18000, NULL,2);
INSERT INTO emp VALUES(14,'Henin',‘FI_MGR', '06-aug-1980’,240000, NULL,1);
INSERT INTO emp VALUES(15,'Greesh',‘Clerk', '06-aug-1980’,240000, NULL,5);

INSERT INTO dept values(1, 'Administration', null, ‘Boston’);
INSERT INTO dept values(2, 'Marketing', null, ‘Boston’);
INSERT INTO dept values(3, 'Purchase', null, ‘perryridge’);
INSERT INTO dept values(4, 'Programming',null, ‘Hudson’);
INSERT INTO dept values(5, 'HR', null, ‘Hudson’);

Alter table dept add foreign key(manager_id references emp(emp_id));

Update dept set manager_id=2 where department_id=1;
Update dept set manager_id=1 where department_id=2;
Update dept set manager_id=8 where department_id=3;
Update dept set manager_id=7 where department_id=4;
```

1) Display the name and salary for all employees whose salary is not in the range of 5000 and 35000

```
SELECT emp_name, salary FROM emp WHERE salary NOT BETWEEN 5000 AND 35000;
```

```
EMP_NAME         SALARY
----------       ----------
Lex              240000
Ismael             4000
Mathew            46000
Henin            240000
```

2) Display the employee name, job ID, and start date of employees hired between February 20, 1990, and May 1, 1998. Order the query in ascending order by start date.

```
SELECT emp_name, job, hiredate FROM emp WHERE hiredate BETWEEN '20-Feb-1990' AND '01-May-1998' ORDER BY hiredate
```

```
EMP_NAME            JOB              HIREDATE
==========       ==========          =========
Bruce             IT_PROG             06-JUL-90
```

```
David          IT_PROG         06-SEP-91
Mathew         CLERK           12-OCT-92
John           FI_ACCOUNT      01-DEC-92
Steven         Marketing       18-MAY-93
Ismael         CLERK           29-MAR-94
Hayes          Marketing       21-APR-98
```

3) list the name and salary of employees who earn between 5,000 and12,000, and are in department 2 or 4. Label the columns Employee and Monthly Salary,respectively.

```
SELECT emp_name "Employee", salary "Monthly Salary" ,depno FROM emp
WHERE salary BETWEEN 5000 AND 30000 AND depno IN (2, 4);
```

```
Employee       Monthly Salary
==========     ==============
Alexander      20000
Bruce          24000
vipin           28000
Hayes          14000
Steven         18000
David          22000
```

4)Display the name and hire date of every employee who was hired in 1994.

```
SELECT emp_name, hiredate FROM emp WHERE hiredate LIKE '%94';
```

```
EMP_NAME       HIREDATE
==========     =========
Ismael         29-MAR-94
```

5). Display the name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

```
SELECT emp_name, salary, comm FROM emp WHERE comm >0
ORDER BY salary DESC, comm DESC;
```

Or

```
SELECT emp_name, salary, comm FROM emp WHERE comm IS NOT NULL
ORDER BY salary DESC, comm DESC;
```

```
EMP_NAME        SALARY          COMM

==========      ==========      ==========

Mathew         46000           200

Hayes           14000          1000
```

6) Display the name and job title of all employees who do not have a manager.

SELECT emp_name, job FROM emp,dept WHERE manager_id IS NULL and emp.depno=dept.department_id;

```
EMP_NAME                JOB
==========          ==========
Greesh                  Clerk
```

7). Display the names of all employees where the third letter of the name is an *a*.

SELECT emp_name FROM emp WHERE emp_name LIKE '__a%';

```
EMP_NAME
==========
Diana
```

8). Display the name of all employees who have an *a* and an *e* in their name.

SELECT emp_name FROM emp WHERE emp_name LIKE '%a%' AND emp_name LIKE '%e%';

```
EMP_NAME
==========
Neena
Alexander
Ismael
Mathew
Hayes
```

9). Display the name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to 2,0000, 4000, or 7,000.

SELECT emp_name, job, salary FROM emp WHERE job IN ('Sa_rep', 'CLERK') AND salary NOT IN (2000, 4000, 7000);

```
EMP_NAME            JOB             SALARY
==========       ==========       ==========
Alexander        Sa_rep           20000
Mathew           CLERK            46000
```

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

SELECT INITCAP(emp_name) "Name", LENGTH(emp_name) "Length" FROM emp

```
WHERE emp_name LIKE 'J%' OR emp_name LIKE 'M%' OR emp_name LIKE
'A%'ORDER BY emp_name;
```

```
Name                          Length
-----------------------------  ----------
Alexander                        9
John                             4
Mathew                          6
```

11)For each employee, display the employee's name, and calculate the number of months between today and the date the employee was hired and years worked. Label the column MONTHS_WORKED. Order your results by the number of months employed. Round the number of months and year up to the closest whole number.
.

```
SELECT emp_name, ROUND(MONTHS_BETWEEN(SYSDATE, hiredate))
MONTHS_WORKED, round(MONTHS_BETWEEN(SYSDATE, hiredate)/12,2)  "NO:
Of YEARS" FROM emp ORDER BY MONTHS_BETWEEN(SYSDATE, hiredate);
```

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

```
SELECT emp.emp_name, emp.depno, dept.department_name FROM emp , dept
WHERE emp.depno = dept.department_id order  by dept.department_name;
```

13) Create a query to display the name and hire date of any employee hired after employee Mathew

```
SELECT emp_Name, HireDate FROM Emp WHERE ((HireDate)>any(SELECT
HireDate FROM Emp WHERE emp_Name='Mathew'));
```

```
EMP_NAME                 HIREDATE
------------------------    ----- ---------
Hayes                       21-APR-98
Ismael                       29-MAR-94
Steven                      18-MAY-93
John                        01-DEC-92
```

14) Display the names and hire dates for all employees who were hired before their managers, along with their manager's names and hire dates. Label the columns Employee, EmpHired, Manager, and Mgr Hired, respectively.

```
SELECT emp.emp_name employee , emp.hiredate "EMP HIRE DATE", emp.salary,
manager.emp_name manager, manager.hiredate "MANAGER HIRE DATE"  FROM emp ,
dept, emp manager WHERE dept.manager_id = manager.emp_no and
emp.depno=dept.department_id and
emp.hiredate < manager.hiredate;
```

```
EMPLOYEE          EMP HIRE DATE    MANAGER      MANAGER HIRE DATE
-------------------   ------------------------ -----------------   ----------------------------------
Lex                06-JAN-80          Neena               06-FEB-87
Alexander         06-JUN-87          vipin               16-NOV-87
Steven           18-MAY-93          Steven              06-JAN-95
Henin             06-AUG-80          Neena               06-FEB-87
```

15) Write a query to display the number of people with the same job.

```
SELECT job, COUNT(*) "No: of Jobs" FROM emp GROUP BY job;
```

```
JOB              NO: OF JOBS
----------       -------------------
IT_PROG                4
Pur_Man                1
CLERK                  2
FI_ACCOUNT             2
FI_MGR                 2
Marketing              3
```

16). Display the manager number and the salary of the lowest paid employee for that manager.Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is less than 6,000. Sort the output in descending order of salary.

```
SELECT min(salary) "MINIMUM SALARY",manager_id, department_name FROM
emp,dept where emp.depno=dept.department_id AND manager_id IS NOT NULL
GROUP BY manager_id, department_name HAVING MIN(salary) > 6000
ORDER BY "MINIMUM SALARY" DESC
```

```
MINIMUM SALARY        MANAGER_ID       DEPARTMENT_NAME
----------------------------   ----------------------   -------------------------------
      24000                 2          Administration
      20000                 7          Programming
      14000                 1          Marketing
```

```
select  emp_name "manager",emp.depno,emp.emp_no, (select min(salary)  from emp e
where (emp.depno=e.depno) group by e.depno having min(salary)>15000) "salary" from
emp,dept where emp.emp_no=dept. MANAGER_ID          and emp.depno=dept.
DEPARTMENT_ID
```

```
select  emp_name "manager", (select min(salary)  from emp e where (emp.depno=e.depno) group by
e.depno having min(salary)>13000) "salary" from emp,dept where emp.emp_no=dept.
MANAGER_ID      and emp.depno=dept. DEPARTMENT_ID
```

```
select min(emp.salary)  from emp,emp e where (emp.depno=e.depno) group by e.depno having
min(emp.salary)>15000
```

17. Write a query to display each department’s name, location, number of employees, and the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.

```
SELECT d.department_name "Name", d.loc "Location ",
COUNT(*) "Number of People", ROUND(AVG(salary),2) "Salary"
FROM emp e, dept d
WHERE e.depno = d.department_id GROUP BY d.department_name, d.loc;
```

```
Name              Location   Number of People    Salary
-------------------- ----------      -------------------     -------------
Administration     Boston          4                  134500
Marketing          Boston          3                  18666.67
Programming        Hudson         4                  23500
Purchase           perryridge     3                  24666.67
```

18). Write a query to display the name and hire date of any employee in the same department as amit. Exclude JOHN.

```
SELECT emp_name, hiredate FROM emp WHERE depno = (SELECT depno
FROM emp WHERE emp_name = 'John') and emp_name<>’John’;
```

```
EMP_NAME                    HIREDATE
------------------------------        ---------------
Neena                        06-FEB-87
Lex                          06-JAN-80
Henin                        06-AUG-80
```

19. Write a query that displays the employee numbers names of all employees who work in a department with any employee whose name contains a *u*.

```
SELECT emp_no, emp_name,department_name FROM emp,dept
WHERE depno IN (SELECT depno FROM emp WHERE emp_name like '%u%') and
emp.depno=dept.department_id;
```

```
 EMP_NO     EMP_NAME               DEPARTMENT_NAME

----------        ----------------------------- -----------------------------

      6           David                       Programming
      7           vipin                       Programming
      5           Bruce                       Programming
      4           Alexander                   Programming
```

20)display employee name and department name of all employees that work in a department that has at least 3 employees. Order the list in alphabetical order first by department name, then by employee name.

SELECT Emp_name, department_name **FROM** emp, dept **WHERE** emp.depno = dept.department_id **AND** emp.depno in (**SELECT** depno **FROM** emp **GROUP BY** depno **HAVING** count(*) >4) **ORDER BY** department_name, emp_name;

21. Write a query to list the length of service of the employees (of the form n years and m months).

SELECT emp_name "employee",to_char(trunc(months_between(sysdate,hiredate)/12))||' years '|| to_char(trunc(mod(months_between (sysdate, hiredate),12)))||' months ' "length of service" **FROM** emp;