

Experiment No: 2

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

Creation, modification, configuration, and deletion of databases Commands

AIM:

Creation of a database and tables using DDL commands

COMMANDS

Create Database

```
mysql> create database testdb;
```

Query OK, 1 row affected (0.01 sec)

Use Database created

```
mysql> use testdb;
```

Database changed

Create Table

```
create table student (sname varchar(30), stid varchar(10), stage int(2), starea varchar(20));
```

Query OK, 0 rows affected (0.34 sec)

Description of student

```
desc student;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| sname | varchar(30) | YES  |     | NULL    |       |
| stid  | varchar(10) | YES  |     | NULL    |       |
| stage | int(2)      | YES  |     | NULL    |       |
| starea | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

MODIFY TABLE DESCRIPTION

alter table student modify stage int(5);

Query OK, 0 rows affected (0.05 sec)

Records: 0 Duplicates: 0 Warnings: 0

desc student;

```
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| sname | varchar(30) | YES  |     | NULL    |       |
| stid  | varchar(10) | YES  |     | NULL    |       |
| stage | int(5)      | YES  |     | NULL    |       |
| starea | varchar(20) | YES  |     | NULL    |       |
| stdept | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

alter table student drop stdept;

Query OK, 0 rows affected (0.55 sec)

Records: 0 Duplicates: 0 Warnings: 0

desc student;

mysql> desc student;

```
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| sname | varchar(30) | YES  |     | NULL    |       |
| stid  | varchar(10) | YES  |     | NULL    |       |
| stage | int(5)      | YES  |     | NULL    |       |
| starea | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

CLEAR ALL VALUES IN TABLE

truncate table student;

Query OK, 0 rows affected (0.25 sec)

CSL333 Database Management Systems Lab

Dept of CSE, SNGCE

```
mysql> desc student;
```

```
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| sname | varchar(30) | YES  |     | NULL    |       |
| stid  | varchar(10) | YES  |     | NULL    |       |
| stage | int(5)      | YES  |     | NULL    |       |
| starea | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

DELETE TABLE BOTH SCHEMA AND DATA

```
drop table student;
```

Query OK, 0 rows affected (0.18 sec)

```
mysql> desc student;
```

ERROR 1146 (42S02): Table 'testdb.student' doesn't exist

DELETE DATABASE

```
mysql> DROP DATABASE databasename;
```

Database dropped

RESULT

Query has run successfully and result is obtained.

By constructing queries using SQL I was able to identify the queries for dealing with database activities.

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

Export ER diagram from the database and verify relationships

AIM

Creation of database schema - DDL (create tables, set constraints, enforce relationships, create indices, delete and modify tables). Export ER diagram from the database and verify relationships**

Consider the employee database given below

emp (emp_id, emp_name, Street_No, city)

works (emp_id, company name, salary)

company (company name, city)

manages (emp_id, manager_id)

Note: Emp_id should start with 'E' in Emp table and emp_id in works table must be the emp_id from emp table . emp_id and manager_id in manages table must be the emp_id from emp table

- I. Add these four tables with sufficient constraints.
- II. Alter table emp add a constraint that emp_name cannot be null
- III. Export ER diagram from database and verify relationships.

COMMANDS

- I. A) Create table emp

Create table emp(emp_id char(8) check(emp_id like 'E%') primary key, emp_name varchar(18), street_no int, city varchar(18));

- B) Create table company

Create table company(company_name varchar(18) primary key, city varchar(18));

- C) Create table works

Create table works(emp_id char(8) references emp(emp_id), company_name varchar(18) references company(company_name), salary float, primary key(emp_id, company_name));

- D) Create table manages

Create table manages(emp_id char(8) references emp2(emp_id),manager_id char(8) references emp2(manager_id),unique(emp_id,manager_id));

II. Alter table emp

alter table emp MODIFY emp_name varchar(18) NOT NULL;

III. Export ER Diagram

RESULT

Query has run successfully and result is obtained.

By constructing queries using SQL I was able to identify the queries for dealing with database activities.

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

Database initialization - Data insert, Data import to a database (bulk import using UI and SQL Commands)**.

AIM

To insert data to tables used in experiment no 3 using insert commands and bulk import using UI and sql commands.

COMMANDS

1. INSERT COMMANDS

```
insert into emp values('E-101','Adarsh',101,'MG Road');
insert into emp values('E-102','Bonny',101,'MG Road');
insert into emp values('E-103','Catherine', 102, 'Cochin');
insert into emp values('E-104','Glenn', 104, 'Ernakulam');
insert into emp values('E-105','George', 201,'MG Road');
insert into emp values('E-106','Hayes', 101, 'MG Road');
insert into emp values('E-107','Johnson',102,'Cochin');
insert into emp values('E-108','Jones', 101, 'Cochin');
insert into emp values('E-109','Karthik', 101, 'Ernakulam');
insert into emp values('E-110','Lavanya', 101, 'Palace Road');
insert into emp values('E-111','Niharika', 102, 'Ernakulam');
```

```
insert into company values('SBI', 'MG Road');
insert into company values('SBT', 'MG Road' );
insert into company values('Federal','Broadway');
insert into company values('Indian Bank', 'Cochin');
insert into company values('SIB', 'Ernakulam');
insert into company values('HDFC', 'Palace Road');
insert into company values('Axis','Cochin');
insert into company values('City bank', 'Ernakulam');
```

```
insert into works values('E-101', 'SBI', 71000);
insert into works values('E-102', 'SBI', 90000);
insert into works values('E-103', 'SBT', 40000);
insert into works values('E-104', 'Federal', 37000);
insert into works values('E-105', 'SBT', 17000);
```

```

insert into works values('E-106', 'Indian Bank', 30000);
insert into works values('E-107', 'SIB', 21000);
insert into works values('E-108', 'SIB', 18000);
insert into works values('E-109', 'Indian Bank', 28000);
insert into works values('E-110', 'SBT', 250000);
insert into works values('E-111', 'Federal', 40000);

```

```

insert into manages values('E-101', 'E-102');
insert into manages values('E-102', Null);
insert into manages values('E-103', 'E-110');
insert into manages values('E-104', 'E-111');
insert into manages values('E-105', 'E-110');
insert into manages values('E-106', 'E-109');
insert into manages values('E-107', Null);
insert into manages values('E-108', Null);
insert into manages values('E-109', Null);
insert into manages values('E-110', Null);
insert into manages values('E-111', null);

```

Export table values to a text file

First see where is the path set for `secure_file_priv`, we can do export and import in this location only (else need to configure it) so use following command:

```
mysql> SHOW VARIABLES LIKE 'secure_file_priv';
```

Variable_name	Value
secure_file_priv	/var/lib/mysql-files/

1 row in set (0.00 sec)

```
mysql> SELECT * FROM WORKS INTO OUTFILE "/var/lib/mysql-files/out2.txt";
```

Query OK, 0 rows affected (0.06 sec)

To show the contents of the file to which data is exported use cat

```
virgo@virgo-Vostro-230:~$ sudo cat "/var/lib/mysql-files/out2.txt"
```

```
e101 sbt 1000
```

Load values from a text file to SQL Table

```
mysql> LOAD DATA INFILE "/var/lib/mysql-files/out2.txt" INTO TABLE EMP4.WORKS;
```

Query OK, 1 row affected (0.05 sec)

Records: 1 Deleted: 0 Skipped: 0 Warnings: 0

RESULT

Query has run successfully and result is obtained.

By constructing queries using SQL I was able to identify the queries for dealing with database activities.

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

Practice SQL commands for DML (insertion, updating, altering, deletion of data, and viewing/querying records based on condition in databases)

AIM

Consider the employee database created in Find results for the following questions

- a. Find the names of all employees who work for SBI.
- b. Find all employees in the database who live in the same cities as the companies for which they work.
- c. Find all employees and their managers in the database who live in the same cities and on the same street number as do their managers.
- d. Find all employees who earn more than the average salary of all employees of their company.
- e. Find the company that pay least total salary along with the salary paid.
- f. Give all managers of SBI a 10 percent raise.
- g. Find the company that has the most employees
- h. Find those companies whose employees earn a higher salary, on average than the average salary at Indian Bank.
- i. Query to find name and salary of all employees who earn more than each employee of 'Indian Bank'

COMMANDS

- a) Find the names of all employees who work for SBI.

```
SELECT emp_name FROM works,emp WHERE company_name='SBI'
and emp.emp_id=works.emp_id;
```

```
EMP_NAME
-----
Adarsh
```

- b) Find all employees in the database who live in the same cities as the companies for which they work.

```
SELECT emp.emp_name FROM emp, works,company WHERE
```

emp.emp_id = works.emp_id AND works.company_name=
company.company_name AND emp.city = company.city

EMP_NAME

Adarsh

George

- c) Find all employees and their managers in the database who live in the same cities and on the same street number as do their managers.

SELECT emp.emp_name,e2.emp_name “manager name” FROM emp,emp e2,
manages WHERE emp.emp_id = manages.emp_id AND e2.Emp_id=
manages.manager_id AND emp.street_no = e2.street_no AND emp.city = e2.city

EMP_NAME

Adarsh

manager name

Bonny

- d) Find all employees who earn more than the average salary of all employees of their company.

SELECT emp_name,emp.emp_id,salary FROM works ,emp WHERE salary >
(SELECT AVG (salary) FROM works S WHERE works.company_name
=S.company_name) and emp.emp_id=works.emp_id

EMP_NAME

Bonny

Hayes

Johnson

Lavanya

Niharika

EMP_ID

E-102

E-106

E-107

E-110

E-111

SALARY

90000

30000

21000

250000

40000

- e). Find the company that pay least total salary along with the salary paid.
SELECT company_name,sum(salary) “SALARY PAID” from Works GROUP
BY company_name HAVING sum(salary) <= all (SELECT sum(salary) FROM
Works GROUP BY company_name)

COMPANY_NAME

SIB

SALARY PAID

39000

- f.) Give all managers of SBI a 10 percent raise.

UPDATE works SET salary = salary * 1.1 WHERE emp_id in (select manager_id
from manages) and company_name ='SBT';

g). Find the company that has the most employees

```
SELECT company_name FROM works GROUP BY company_name
HAVING COUNT (DISTINCT emp_id) >= ALL (SELECT COUNT (DISTINCT
emp_id) FROM works GROUP BY company_name)
```

COMPANY_NAME

SBT

h) Find those companies whose employees earn a higher salary, on average than the average salary at Indian Bank.

```
SELECT company_name FROM works GROUP BY company_name HAVING
AVG(salary)> (SELECT AVG(salary) FROM works WHERE company_name =
'Indian Bank' GROUP BY
company_name)
```

COMPANY_NAME

SBI

Federal

SBT

i).Query to find name and salary of all employees who earn more than each employee of 'Indian Bank'

```
SELECT emp_name,salary FROM works,emp
WHERE salary > (SELECT MAX(salary) FROM works WHERE company_name =
'Indian Bank' GROUP BY company_name) and emp.emp_id=works.emp_id;
```

EMP_NAME	SALARY
-----	-----
Adarsh	71000
Bonny	99000
Catherine	40000
Glenn	37000
Lavanya	250000
Niharika	40000

```

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

- Display the number of faculties.
- Display the total mark for each student.
- Display the subject, average mark for each subject.
- Display the name of subjects for which atleast one student got below 40%.
- Display the name, subject and percentage of mark who got below 40 %.
- Display the faculties and allotted subjects for each faculty
- Display the name of faculties who take more than one subject.
- 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 allotted 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 and 12,000, 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 2,000, 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
- CSL333 Database Management Systems Lab Dept of CSE, SNGCE

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 and 12,000, and are in department 2 or 4. Label the columns Employee and Monthly Salary, respectively.

SELECT emp_name "Employee", salary "Monthly Salary", deptno FROM emp WHERE salary BETWEEN 5000 AND 30000 AND deptno 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.deptno=dept.department_id;

<u>EMP_NAME</u>	<u>JOB</u>
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%';

<u>EMP_NAME</u>
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%';

<u>EMP_NAME</u>
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);

<u>EMP_NAME</u>	<u>JOB</u>	<u>SALARY</u>
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.deptno, dept.department_name FROM emp , dept WHERE emp.deptno = 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.deptno=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;
```