LAB-SHEET 3
DBMS

Name : Anuvind M P
Roll no : AM.EN.U4AIE22010

1.

Question 1

Consider the relations Employee and Department given in Lab 2 and answer the following queries in SQL

DEPARTMENT

```
--create table department(dept_no int primary key,dname varchar(14) NOT NULL,loc varchar(20));

--insert into department values(10,'accounting','new
york'),(20,'research','dallas'),(30,'sales','chicago'),(40,'operations','boston');
```

EMPLOYEE

```
 --create table employee(empno int primary key,
                                        --    ename varchar(20) not null,
                                              -- job varchar(10),
                                              -- mgr_id int,
                                              -- hired_date date,
                                              -- basic_sal numeric(6,2) default(1000),
                                              -- incentive numeric(6,2) check(incentive < basic_sal),
                                              -- deptno int references department(dept_no));

-- INSERT INTO employee VALUES
--  (7369, 'smith', 'clerk', 7902, '1980-12-17', 6800, NULL, 20),
--  (7499, 'allen', 'salesman', 7698, '1981-02-20', 1160, 300, 30),
--  (7521, 'ward', 'salesman', 7698, '1981-02-22', 1125, 500, 30),
--  (7566, 'jones', 'manager', 7839, '1981-04-02', 2297, NULL, 20),
--  (7654, 'Martin', 'salesman', 7698, '1981-09-28', 1125, 1124, 30),
--  (7698, 'blake', 'manager', 7839, '1981-05-01', 2285, NULL, 30),
--  (7782, 'clark', 'manager', 7839, '1981-06-09', 2245, NULL, 10),
--  (7788, 'scott', 'analyst', 7566, '1982-12-09', 1300, NULL, 20);
```

1.Create a table Department_locn with fields deptno and location where both are its primary keys and deptno references Department table.

```
create table Department_loc(dept_no int references department(dept_no),
                        location varchar,
```

primary key(dept_no,location));

2.Change the empno of the employee whose name is ‘Meena’.

INSERT INTO employee VALUES(7333, 'meenu', 'clerk', 7999, '1980-12-16', 6810, NULL, 30);

```
update employee
 set empno=7334
 where ename='Meena';
```

3.Drop the column DOB from Employee.

```
ALTER TABLE Employee
DROP COLUMN hired_date;
```

4.Retrieve information of departments with deptno 1, 4, 8

select * from department where dept_no in(10,20,30);

Data Output | Messages | Notifications

	dept_no [PK] integer	dname character varying (14)	loc character varying (20)
1	10	accounting	new york
2	20	research	dallas
3	30	sales	chicago

5.Show the different salaries of employees eliminating duplicate values.

select distinct basic_sal from employee;

Data Output | Messages | Notifications

	basic_sal numeric (6,2)
1	2245.00
2	2285.00
3	1300.00
4	6810.00
5	1125.00
6	2297.00
7	1160.00
8	6800.00

6.Show details of the department sorted by department number.

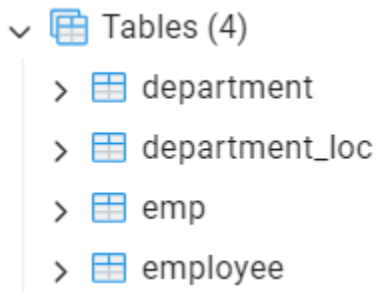
select * from department order by dept_no;

Data Output | Messages | Notifications

	dept_no [PK] integer	dname character varying (14)	loc character varying (20)
1	10	accounting	new york
2	20	research	dallas
3	30	sales	chicago
4	40	operations	boston

7.Create a table emp having three fields empno, empname and salary from the source table employee . The table emp should not have any records.

```
CREATE TABLE emp AS
SELECT empno, ename, basic_sal
FROM employee
WHERE 1 = 0;
```

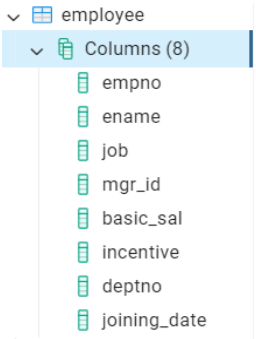


8.Insert data into emp using employee as the source of data.

```
INSERT INTO emp (empno, ename, basic_sal)
SELECT empno, ename, basic_sal
FROM employee;
```

9.Add a column joindate to employee table.

```
alter table employee
add joining_date date;
```

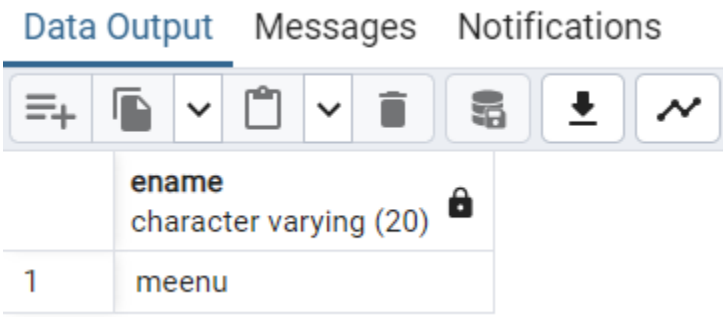


10. Drop primary key from department.

```
alter table department
drop constraint department_pkey;
```

11.Display names of employees whose name begins with ‘m’ and has 2 ‘e’s

```
select ename from employee where ename like 'm%e%e%';
```

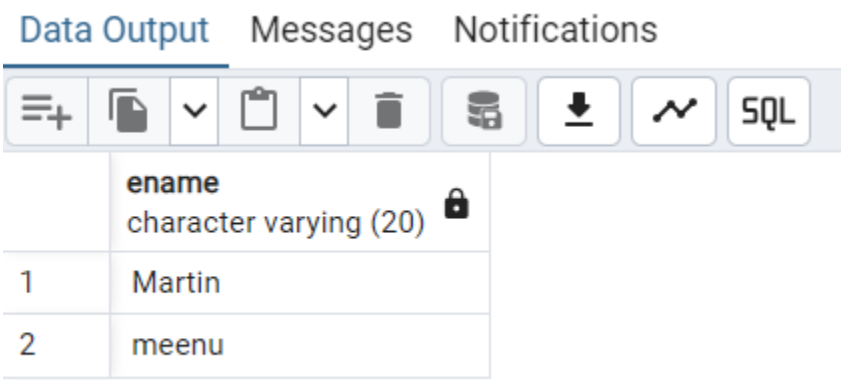


Data Output | Messages | Notifications

	ename character varying (20)
1	meenu

12. Display names of employees whose name begins with ‘m’ and salary > 10000.

```
select ename from employee where ename like 'M%' or ename like 'm%' and basic_sal>1000;
```

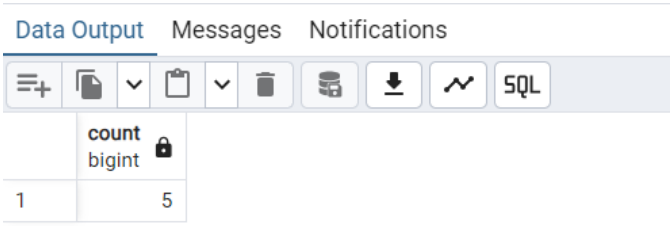


Data Output | Messages | Notifications | SQL

	ename character varying (20)
1	Martin
2	meenu

13. Give the number of employees whose salary is greater than 2000.

```
select count(empno) from employee where basic_sal>2000;
```



Data Output | Messages | Notifications | SQL

	count bigint
1	5

14. Arrange the employees by their department numbers.

select * from employee order by deptno;

Data Output | Messages | Notifications

	empno [PK] integer	ename character varying (20)	job character varying (10)	mgr_id integer	basic_sal numeric (6,2)	incentive numeric (6,2)	deptno integer	joining_date date
1	7782	clark	manager	7839	2245.00	[null]	10	[null]
2	7566	jones	manager	7839	2297.00	[null]	20	[null]
3	7788	scott	analyst	7566	1300.00	[null]	20	[null]
4	7369	smith	clerk	7902	6800.00	[null]	20	[null]
5	7698	blake	manager	7839	2285.00	[null]	30	[null]
6	7333	meenu	clerk	7999	6810.00	[null]	30	[null]
7	7499	allen	salesman	7698	1160.00	300.00	30	[null]
8	7521	ward	salesman	7698	1125.00	500.00	30	[null]
9	7654	Martin	salesman	7698	1125.00	1124.00	30	[null]

15. Display the department number along with number of employees in each department.

select deptno,count(empno) from employee group by deptno;

	deptno integer	count bigint
1	30	5
2	10	1
3	20	3

16. Find the departments having more than 10 employees.

select deptno,count(empno) from employee group by deptno having count(empno)>3;

Data Output | Messages | Notifications

	deptno integer	count bigint
1	30	5

Question 2

Do all the questions we discussed in the class based on set operations.

```
--create table customer(cno int primary key,cname varchar(30),ctype varchar(10) );

--create table cust_fd(cno int references customer(cno),fd_no int primary key,fd_amt numeric(50),int_rate numeric(50));

--create table cust_loan(cno int references customer(cno),ln_no int primary key,ln_type varchar,ln_amt numeric);

--create table emp_details(cno int references customer(cno),ename varchar,sal numeric,br_no int);

--create table account(acc_no int primary key,cno int references customer(cno),veri_emp_no int,acc_type varchar );

-- INSERT INTO customer (cno, cname, ctype) VALUES
-- (1, 'John Doe', 'Premium'),
-- (2, 'Jane Smith', 'Basic'),
-- (3, 'Alice Johnson', 'Premium'),
-- (4, 'Bob Brown', 'Basic'),
-- (5, 'Charlie Adams', 'Premium'),
-- (6, 'Diana Clark', 'Basic'),
-- (7, 'Evan Taylor', 'Premium'),
-- (8, 'Fiona Lewis', 'Basic'),
-- (9, 'George Martin', 'Premium'),
-- (10, 'Hannah Scott', 'Basic');

-- INSERT INTO cust_fd (cno, fd_no, fd_amt, int_rate) VALUES
-- (1, 1001, 50000, 4.5),
-- (2, 1002, 75000, 5.0),
-- (3, 1003, 60000, 4.8),
-- (4, 1004, 80000, 5.2),
-- (5, 1005, 55000, 4.7),
-- (6, 1006, 45000, 5.1),
-- (7, 1007, 95000, 4.9),
-- (8, 1008, 105000, 5.3),
-- (9, 1009, 62000, 4.6),
-- (10, 1010, 78000, 5.4);

-- INSERT INTO cust_loan (cno, ln_no, ln_type, ln_amt) VALUES
-- (1, 2001, 'Home Loan', 250000),
-- (2, 2002, 'Personal Loan', 50000),
-- (3, 2003, 'Car Loan', 120000),
-- (4, 2004, 'Education Loan', 30000),
-- (5, 2005, 'Home Loan', 270000),
-- (6, 2006, 'Car Loan', 150000),
-- (7, 2007, 'Personal Loan', 80000),
-- (8, 2008, 'Business Loan', 500000),
-- (9, 2009, 'Education Loan', 45000),
```

-- (10, 2010, 'Personal Loan', 100000);

-- INSERT INTO emp_details (cno, ename, sal, br_no) VALUES
-- (1, 'Mark Spencer', 55000, 101),
-- (2, 'Lucy Martin', 60000, 102),
-- (3, 'John David', 52000, 103),
-- (4, 'Emily White', 58000, 104),
-- (5, 'Michael James', 61000, 105),
-- (6, 'Sophia King', 54000, 106),
-- (7, 'Oliver Green', 57000, 107),
-- (8, 'Amelia Turner', 62000, 108),
-- (9, 'Mason Lee', 51000, 109),
-- (10, 'Isabella Wright', 53000, 110);

-- INSERT INTO account (acc_no, cno, veri_emp_no, acc_type) VALUES
-- (3001, 1, 1, 'Savings'),
-- (3002, 2, 2, 'Current'),
-- (3003, 3, 3, 'Savings'),
-- (3004, 4, 4, 'Current'),
-- (3005, 5, 5, 'Savings'),
-- (3006, 6, 6, 'Current'),
-- (3007, 7, 7, 'Savings'),
-- (3008, 8, 8, 'Current'),
-- (3009, 9, 9, 'Savings'),
-- (3010, 10, 10, 'Current');

--a)List the customer number of those customers who have got both
--loan and fd at the bank

--select cno from cust_fd intersect select cno from cust_loan;

--b)List the customer number of those customers who have got either
--loan or fd at the bank

--select cno from cust_fd union select cno from cust_loan;

--c)List the customer number of those customers who have got FD
--but not loan at the bank

-- Insert customer details for those who have FD but no loan

--INSERT INTO customer (cno, cname, ctype) VALUES
-- (11, 'Ivy Cole', 'Premium'),
-- (12, 'Jackie Ford', 'Basic'),
-- (13, 'Kyle Evans', 'Premium');

-- Insert customers with FD but no loan
-- Assume customer numbers 11, 12, and 13 have FD but no loan

```
--INSERT INTO cust_fd (cno, fd_no, fd_amt, int_rate) VALUES
-- (11, 1011, 85000, 4.6),
-- (12, 1012, 70000, 4.8),
-- (13, 1013, 92000, 5.1);

--select cno from cust_fd except select cno from cust_loan;
```

OUTPUTS

INTERSECT

Data Output | Messages | Notifications

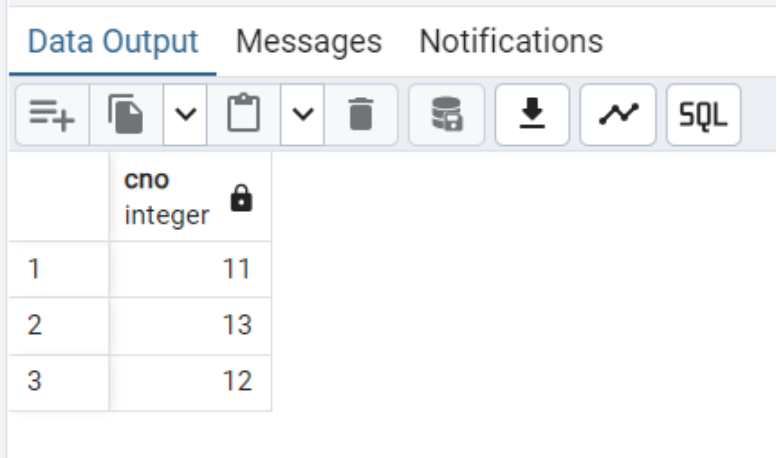
	cno integer
2	3
3	5
4	4
5	10
6	6
7	2
8	7
9	1
10	8

UNION

	cno integer
1	10
2	2
3	3
4	5
5	8
6	4
7	6
8	9
9	7

Total rows: 10 of 10

EXCEPT



	cno integer
1	11
2	13
3	12