**ASSIGNMENT-1**

**Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) using CREATE, ALTER, DROP, INSERT statements.**

CREATE TABLE EMPLOYEE (

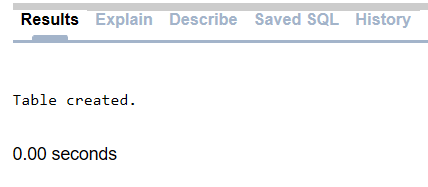
Fname VARCHAR(20), Minit CHAR(1), Lname VARCHAR(20), Ssn CHAR(9) PRIMARY KEY, Bdate DATE,

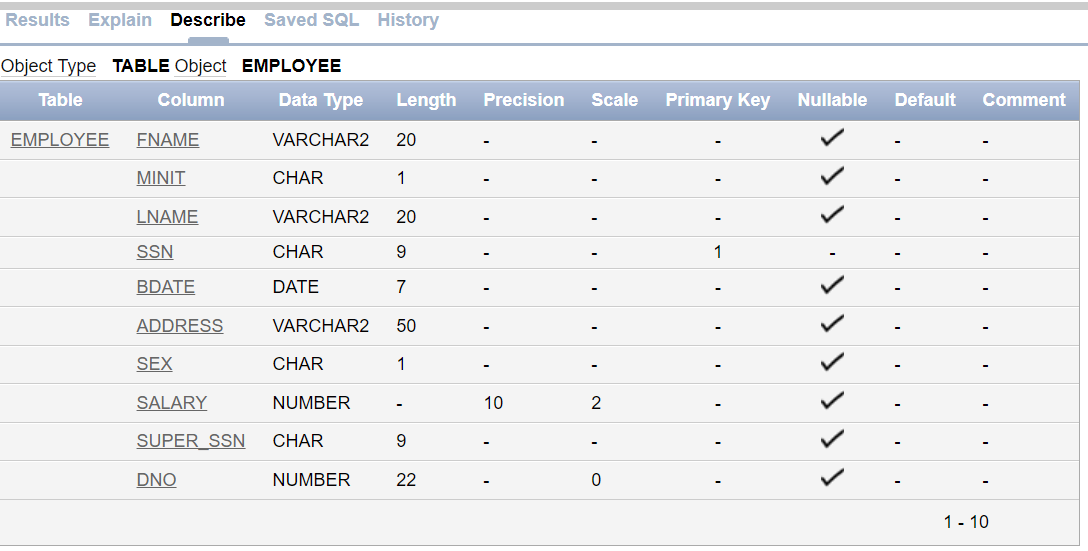
Address VARCHAR(50), Sex CHAR(1), Salary DECIMAL(10, 2), Super\_ssn CHAR(9),

Dno INT CHECK (Dno >= 1 AND Dno <= 10),

CONSTRAINT super\_ssn FOREIGN KEY (Super\_ssn) REFERENCES EMPLOYEE (Ssn) ON DELETE CASCADE

);



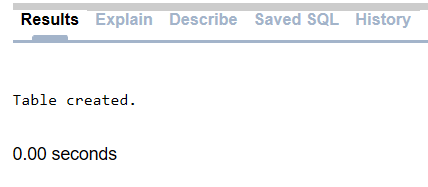
DESC EMPLOYEE;

CREATE TABLE DEPARTMENT (

Dname VARCHAR(20) NOT NULL, Dnumber INT PRIMARY KEY, Mgr\_ssn CHAR(9), Mgr\_start\_date DATE,

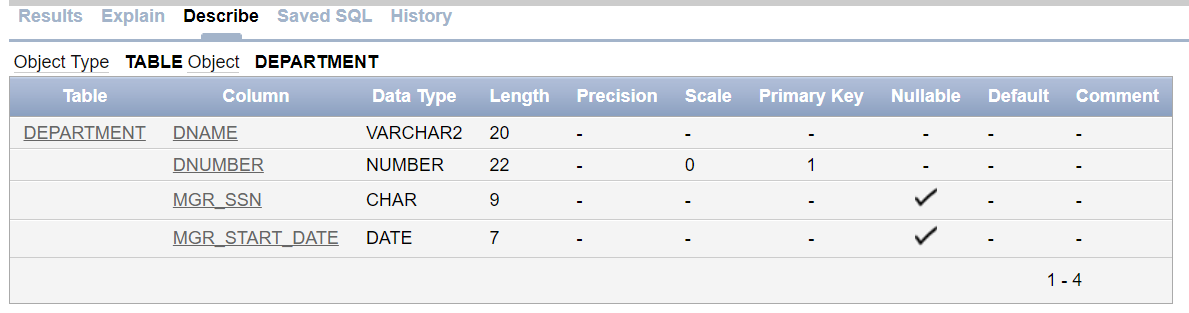
CONSTRAINT fk\_mgr\_ssn FOREIGN KEY (Mgr\_ssn) REFERENCES EMPLOYEE (Ssn) ON DELETE CASCADE

);



DESC DEPARTMENT;

ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber) ON DELETE CASCADE;



INSERT INTO EMPLOYEE VALUES ('John', 'B', 'Smith', '123456789', '01-09-1985', '731 Fondren, Houston, TX', 'M', 30000, NULL, 5);

INSERT INTO EMPLOYEE VALUES ('Franklin', 'T', 'Wong', '333445555', '12-08-1955', '638 Voss, Houston, TX', 'M', 40000, NULL, 5);

INSERT INTO EMPLOYEE VALUES ('Alicia', 'J', 'Zelaya', '999887777', '1-19-1968', '3321 Castle, Spring, TX', 'F', 25000, NULL, 4);

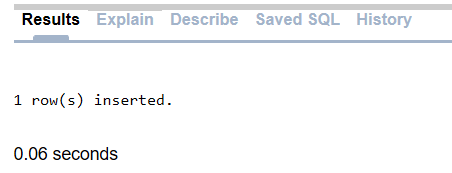
INSERT INTO EMPLOYEE VALUES ('Jennifer', 'S', 'Wallace', '987654321', '6-20-1941', '291 Berry, Bellaire, TX', 'F', 43000, NULL, 4);

INSERT INTO EMPLOYEE VALUES ('Ramesh', 'K', 'Narayan', '666884444', '9-15-1962', '5631 Rice, Houston, TX', 'M', 38000, NULL, 5);

INSERT INTO EMPLOYEE VALUES ('Joyce', 'A', 'English', '453453453', '7-31-1972', '5631 Rice, Houston, TX', 'F', 25000, NULL, 5);

INSERT INTO EMPLOYEE VALUES ('Ahmad', 'V', 'Jabbar', '987987987', '3-29-1969', '980 Dallas, Houston, TX', 'M', 25000, NULL, 4);

INSERT INTO EMPLOYEE VALUES ('James', 'E', 'Borg', '888665555', '11-10-1937', '450 Stone, Houston, TX', 'M', 55000, NULL, 1);



UPDATE EMPLOYEE SET super\_ssn = '888665555' WHERE ssn = '333445555';

UPDATE EMPLOYEE SET super\_ssn = '333445555' WHERE ssn = '123456789';

UPDATE EMPLOYEE SET super\_ssn = '987654321' WHERE ssn = '999887777';

UPDATE EMPLOYEE SET super\_ssn = '888665555' WHERE ssn = '987654321';

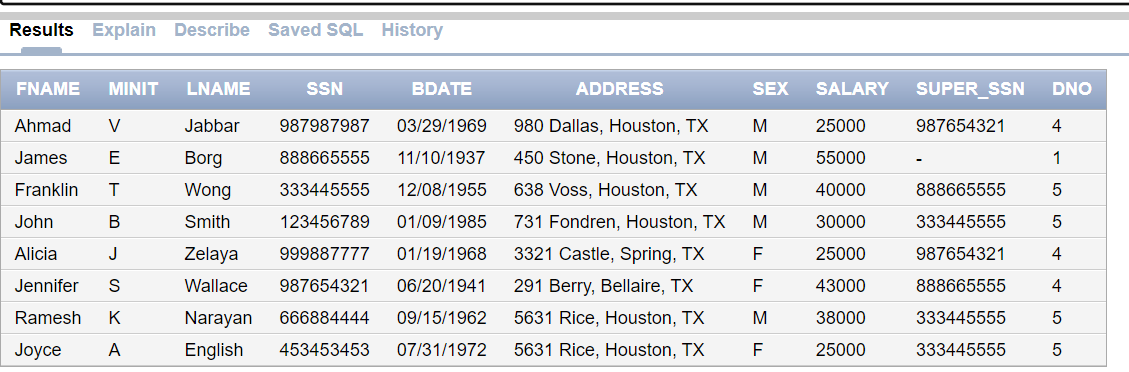
UPDATE EMPLOYEE SET super\_ssn = '333445555' WHERE ssn = '666884444';

UPDATE EMPLOYEE SET super\_ssn = '333445555' WHERE ssn = '453453453';

UPDATE EMPLOYEE SET super\_ssn = '987654321' WHERE ssn = '987987987';

UPDATE EMPLOYEE SET super\_ssn = NULL WHERE ssn = '888665555';

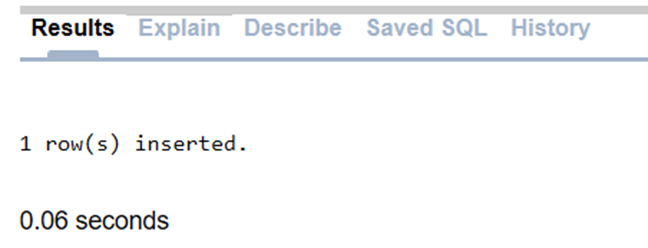
SELECT \* from EMPLOYEE;



INSERT INTO DEPARTMENT VALUES ('Research', 5, NULL, '05-22-1988');

INSERT INTO DEPARTMENT VALUES ('Administration', 4, NULL, '01-01-1995');

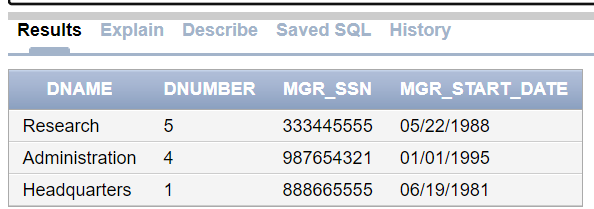
INSERT INTO DEPARTMENT VALUES ('Headquarters', 1, NULL, '06-19-1981');



UPDATE DEPARTMENT SET mgr\_ssn = '333445555' WHERE Dnumber = 5;

UPDATE DEPARTMENT SET mgr\_ssn = '987654321' WHERE Dnumber = 4;

UPDATE DEPARTMENT SET mgr\_ssn = '888665555' WHERE Dnumber = 1;

SELECT \* from DEPARTMENT

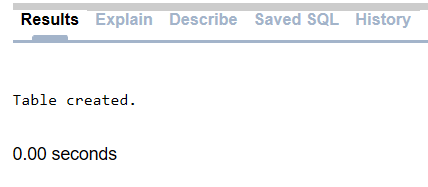
-------------------------------------------------------DEPT\_LOCATION TABLE----------------------------------------------------------

CREATE TABLE DEPT\_LOCATIONS (

Dnumber INT, Dlocation VARCHAR(50), PRIMARY KEY (Dnumber, Dlocation),

FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)

);



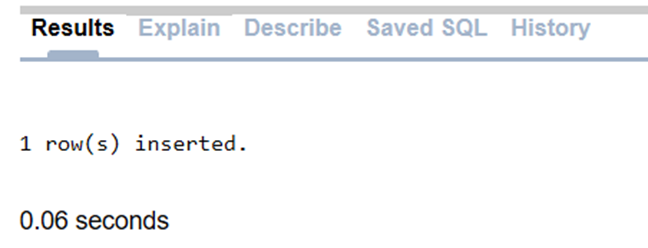
INSERT INTO DEPT\_LOCATIONS VALUES (1, 'Houston');

INSERT INTO DEPT\_LOCATIONS VALUES (4, 'Stafford');

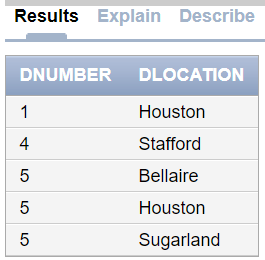
INSERT INTO DEPT\_LOCATIONS VALUES (5, 'Bellaire');

INSERT INTO DEPT\_LOCATIONS VALUES (5, 'Sugarland');

INSERT INTO DEPT\_LOCATIONS VALUES (5, 'Houston');



select \* from DEPT\_LOCATIONS;

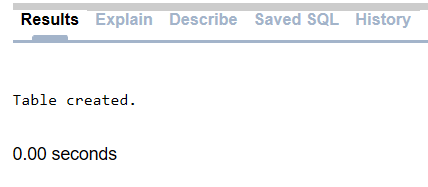


-------------------------------------------------------PROJECT TABLE----------------------------------------------------------

CREATE TABLE PROJECT (

Pname VARCHAR(50), Pnumber INT PRIMARY KEY, Plocation VARCHAR(50), Dnum INT,

FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );



INSERT INTO PROJECT VALUES ('ProductX', 1, 'Bellaire', 5);

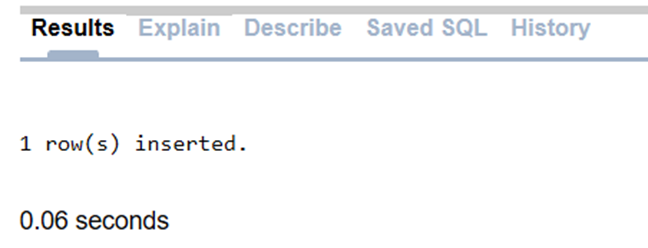
INSERT INTO PROJECT VALUES ('ProductY', 2, 'Sugarland', 5);

INSERT INTO PROJECT VALUES ('ProductZ', 3, 'Houston', 5);

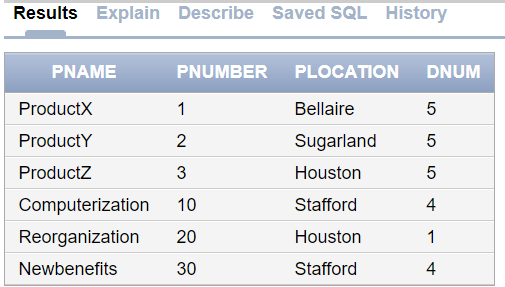
INSERT INTO PROJECT VALUES ('Computerization', 10, 'Stafford', 4);

INSERT INTO PROJECT VALUES ('Reorganization', 20, 'Houston', 1);

INSERT INTO PROJECT VALUES ('Newbenefits', 30, 'Stafford', 4);



select \* from PROJECT;



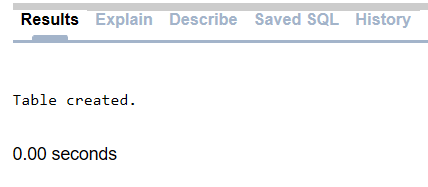
-------------------------------------------------------WORKS\_ON TABLE----------------------------------------------------------

CREATE TABLE WORKS\_ON (

Essn CHAR(9), Pno INT, Hours DECIMAL(5, 2), PRIMARY KEY (Essn, Pno),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn), FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber)

);



INSERT INTO WORKS\_ON VALUES ('123456789', 1, 32.5);

INSERT INTO WORKS\_ON VALUES ('123456789', 2, 7.5);

INSERT INTO WORKS\_ON VALUES ('666884444', 3, 40.0);

INSERT INTO WORKS\_ON VALUES ('453453453', 1, 20.0);

INSERT INTO WORKS\_ON VALUES ('333445555', 2, 10.0);

INSERT INTO WORKS\_ON VALUES ('333445555', 3, 10.0);

INSERT INTO WORKS\_ON VALUES ('333445555', 10, 10.0);

INSERT INTO WORKS\_ON VALUES ('987654321', 30, 30.0);

INSERT INTO WORKS\_ON VALUES ('123456789', 30, 10.0);

INSERT INTO WORKS\_ON VALUES ('333445555', 30, 30.0);

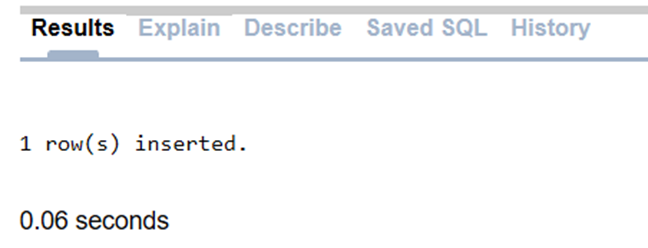
INSERT INTO WORKS\_ON VALUES ('999887777', 10, 10.0);

INSERT INTO WORKS\_ON VALUES ('999887777', 30, 10.0);

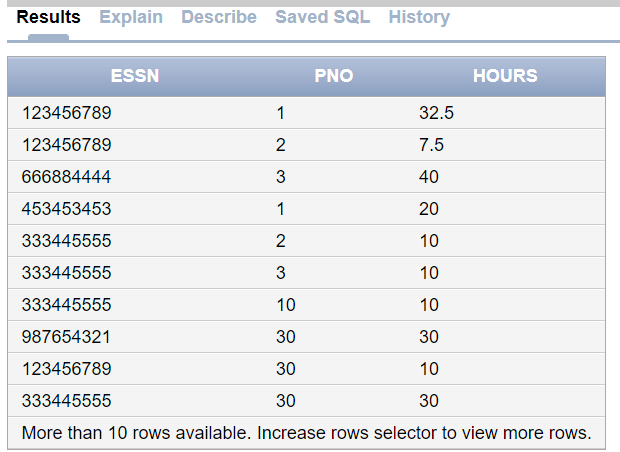
INSERT INTO WORKS\_ON VALUES ('987987987', 20, 20.0);

INSERT INTO WORKS\_ON VALUES ('987654321', 20, 15.0);

INSERT INTO WORKS\_ON VALUES ('888665555', 20, NULL);



select \* from WORKS\_ON;



-------------------------------------------------------DEPENDENT TABLE----------------------------------------------------------

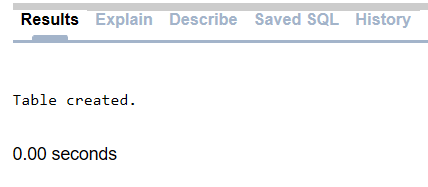
CREATE TABLE DEPENDENT (

Essn CHAR(9), Dependent\_name VARCHAR(50), Sex CHAR(1), Bdate DATE, Relationship VARCHAR(50),

PRIMARY KEY (Essn, Dependent\_name),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn)

);



INSERT INTO DEPENDENT VALUES ('333445555', 'Alice', 'F', '04-05-1986', 'Daughter');

INSERT INTO DEPENDENT VALUES ('333445555', 'Theodore', 'M', '10-25-1983', 'Son');

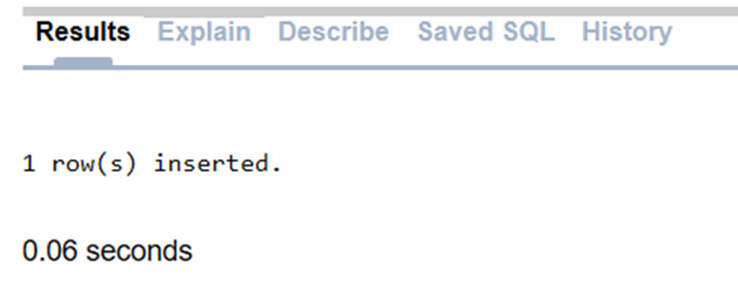
INSERT INTO DEPENDENT VALUES ('333445555', 'Joy', 'F', '05-03-1958', 'Spouse');

INSERT INTO DEPENDENT VALUES ('987654321', 'Abner', 'M', '02-28-1942', 'Spouse');

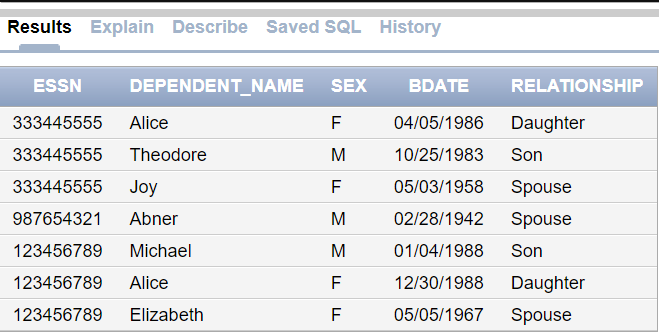
INSERT INTO DEPENDENT VALUES ('123456789', 'Michael', 'M', '01-04-1988', 'Son');

INSERT INTO DEPENDENT VALUES ('123456789', 'Alice', 'F', '12-30-1988', 'Daughter');

INSERT INTO DEPENDENT VALUES ('123456789', 'Elizabeth', 'F', '05-05-1967', 'Spouse');



select \* from DEPENDENT;



**ASSIGNEMENT-2**

**Implementing the queries for Insertion, Updation, Deletion operations. Use ROLL BACK, COMMIT & SAVE POINTS Concepts with INSERT, UPDATE, DELETE statements.**

1. **Write a query to insert new department Prodcuction to existing DEPARTMENT table and commit and rollback**

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

SQL> insert into DEPARTMENT values('Production',3,123456789,'16-DEC-1992');

1 row created.

SQL> commit;

Commit complete.

SQL> rollback;

Rollback complete.

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Production 3 123456789 16-DEC-92

1. **Delete department where dnum=5 and undo the changes.**

SQL> delete from DEPARTMENT where dno=5;

1 row deleted.

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Production 3 123456789 16-DEC-92

SQL> select \* from EMPLOYEE;

FNAME M LNAME SSN BDATE

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

ADDRESS S SALARY SUPER\_SSN

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

DNO

----------

Ahmad V Jabbar 987987987 29-MAR-69

980 Dallas, Houston, TX M 25000 987654321

4

James E Borg 888665555 10-NOV-37

450 Stone, Houston, TX M 55000

1

Alicia J Zelaya 999887777 19-JAN-68

3321 Castle, Spring, TX F 25000 987654321

4

Jennifer S Wallace 987654321 20-JUN-41

291 Berry, Bellaire, TX F 43000 888665555

4

SQL> rollback;

Rollback complete.

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Production 3 123456789 16-DEC-92

SQL> select \* from EMPLOYEE;

FNAME M LNAME SSN BDATE

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

ADDRESS S SALARY SUPER\_SSN

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

DNO

----------

Ahmad V Jabbar 987987987 29-MAR-69

980 Dallas, Houston, TX M 25000 987654321

4

James E Borg 888665555 10-NOV-37

450 Stone, Houston, TX M 55000

1

Franklin T Wong 333445555 08-DEC-55

638 Voss, Houston, TX M 40000 888665555

5

John B Smith 123456789 09-JAN-85

731 Fondren, Houston, TX M 30000 333445555

5

Alicia J Zelaya 999887777 19-JAN-68

3321 Castle, Spring, TX F 25000 987654321

4

Jennifer S Wallace 987654321 20-JUN-41

291 Berry, Bellaire, TX F 43000 888665555

4

Ramesh K Narayan 666884444 15-SEP-62

5631 Rice, Houston, TX M 38000 333445555

5

Joyce A English 453453453 31-JUL-72

5631 Rice, Houston, TX F 25000 333445555

5

8 rows selected.

1. **Create two savepoints for deleting department 5 and 3 and also undo the changes from ssavepoint which is first created**

SQL> savepoint s1;

Savepoint created.

SQL> delete from DEPARTMENT where dno=5;

1 row deleted.

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Production 3 123456789 16-DEC-92

SQL> savepoint sp2;

Savepoint created

SQL> delete from DEPARTMENT where dno=3;

1 row deleted.

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

SQL> rollback to s1;

Rollback complete.

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Production 3 123456789 16-DEC-92

1. **Create three save points and undo after second savepoint**
   * 1. **Update the Dname as Finance for Dnum=3**
     2. **Update the Dname as Marketing for Dnum=1**
     3. **Update the Dname as HR for Dnum=5**

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Production 3 123456789 16-DEC-92

SQL> savepoint a1;

Savepoint created.

SQL> update DEPARTMENT set Dname='Finance' where Dnumber=3;

1 row updated.

SQL> select\* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Finance 3 123456789 16-DEC-92

SQL> savepoint a2;

Savepoint created.

SQL> update DEPARTMENT set Dname='Marketing' where Dnumber=1;

1 row updated.

SQL> select\* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Marketing 1 888665555 19-JUN-81

Finance 3 123456789 16-DEC-92

SQL> savepoint a3;

Savepoint created.

SQL> update DEPARTMENT set Dname='HR' where Dnumber=5;

1 row updated.

SQL> select\* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

HR 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Marketing 1 888665555 19-JUN-81

Finance 3 123456789 16-DEC-92

SQL> rollback to a2;

Rollback complete.

SQL> select \* from DEPARTMENT;

DNAME DNUMBER MGR\_SSN MGR\_START

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

Research 5 333445555 22-MAY-88

Administration 4 987654321 01-JAN-95

Headquarters 1 888665555 19-JUN-81

Finance 3 123456789 16-DEC-92

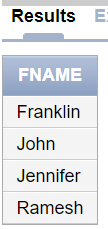
**ASSIGNEMENT-3**

**Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS,**

**UNION,EXCEPT ,INTERSECT,BETWEEN.**

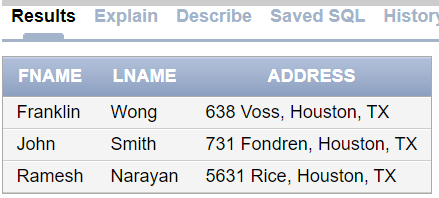
1. **Retrieve the Fname and salary of all employee who is working for department 4 and make over salary 30000 or working for department 5 make over salary 25000.**

select Fname from EMPLOYEE where (Dno=4 and salary>30000 or Dno=5 and salary>25000)



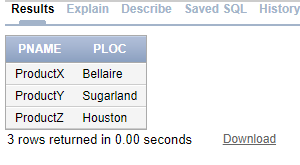
1. **For each employee retrieve the employees first and last name and first and last name of his or her immediate supervisor retrieve the Fname and address of all employee whose salary is in the range of 30000 to 40000**

select Fname,Lname,Address from EMPLOYEE where (salary>=30000 and salary<=40000)



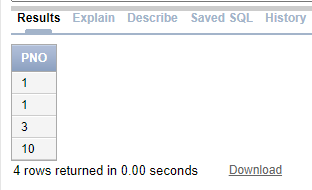
1. **Retrieve the Pname and plocation of all project controlled by department with number 5**

select Pname,Plocation from PROJECT where Dnum=5;



1. **Retrieve pnumber for which employee worked for more than 30 hours**

select Pno from WORKS\_ON where Hours>=30



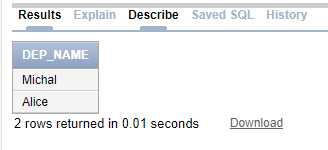
1. **Retrieve the names of employee who are mangers of the department**

select Fname from EMPLOYEE where SSN in( select Mgr\_SSN from DEPARTMENT)



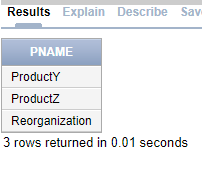
1. **Retrieve the Names of dependent born in the year 1988**

select Dependent\_name from DEPENDENT where extract(year from Bdate)=1988



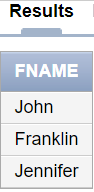
1. **Retrieve the name of project which are located at a place which contains a letter u**

select Pname from Project where Plocation like '%u%'



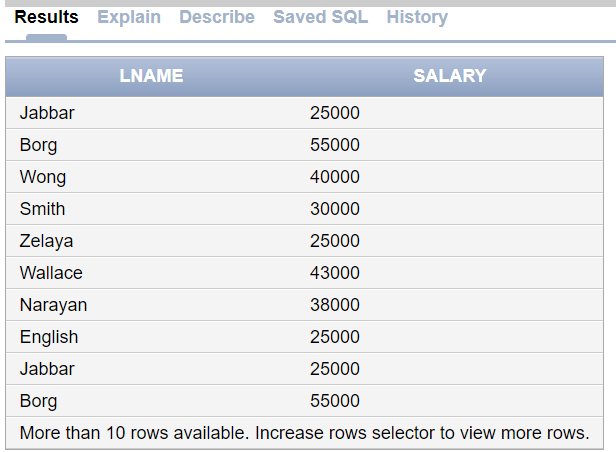
1. **Retrieve the names of each employee who has dependent**

select distinct fname from EMPLOYEE,DEPENDENT where Ssn=ESSN



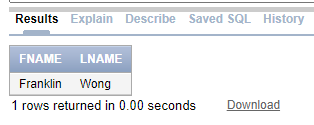
1. **For every department located at Houston retrieve the Manger Lname and salary if Lname of employee contains only four characters.**

select Lname,Salary from EMPLOYEE, DEPT\_LOCATIONS where Dlocation ='Houston'

****

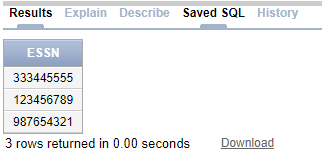
1. **Retrieve the Fname and Lname of employee who is a supervisor for a employee with first name as Ramesh**

select Fname,Lname From EMPLOYEE where Ssn in(select Super\_Ssn from EMPLOYEE where Fname='Ramesh')



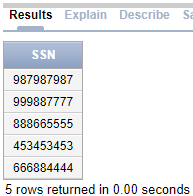
1. **List the ssn of all employee who are manger and also works on project**

Select distinct ESSN from DEPARTMENT,WORKS\_ON where Mgr\_ssn=ESSN



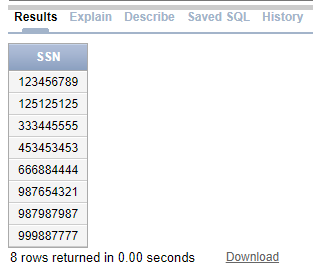
1. **Retrieve the ssn of employee who have no dependent**

Select Ssn from EMPLOYEE where Ssn not in(select ESSN from DEPENDENT)



1. **List the ssn of all employee who either works on project or having the dependent**

Select Ssn from EMPLOYEE where Ssn in(select ESSN from DEPENDENT) or Ssn in(select ESSN from WORKS\_ON)



1. **Retrieve the Fname of employee in the increasing order of their salary and decreasing order of department number**

select Fname from EMPLOYEE order by Salary asc,Dno desc



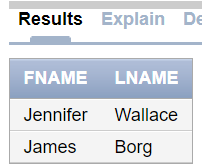
1. **Retrieve the Dependent name in the decreasing order of their birth date**

select Dependent\_name from DEPENDENT order by Bdate desc

****

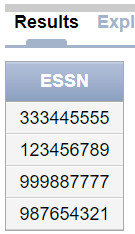
1. **Retrieve the firstname and lastname of all employee whose salary is more than the salary of employees working for department 5.**

select Fname,Lname from EMPLOYEE where Salary >all(Select Salary from EMPLOYEE where Dno=5)



1. **Retrieve the ssn of all employee who work on any of the project controlled by department 4**

select distinct ESSN from WORKS\_ON,PROJECT where Pno=Pnumber AND Dnum=4



1. **Retrieve the name of each employee who has a dependent with same first name and is the same gender as the employee**

select fname from employee,dependent where essn=ssn and fname=dependent\_name and sex=sex

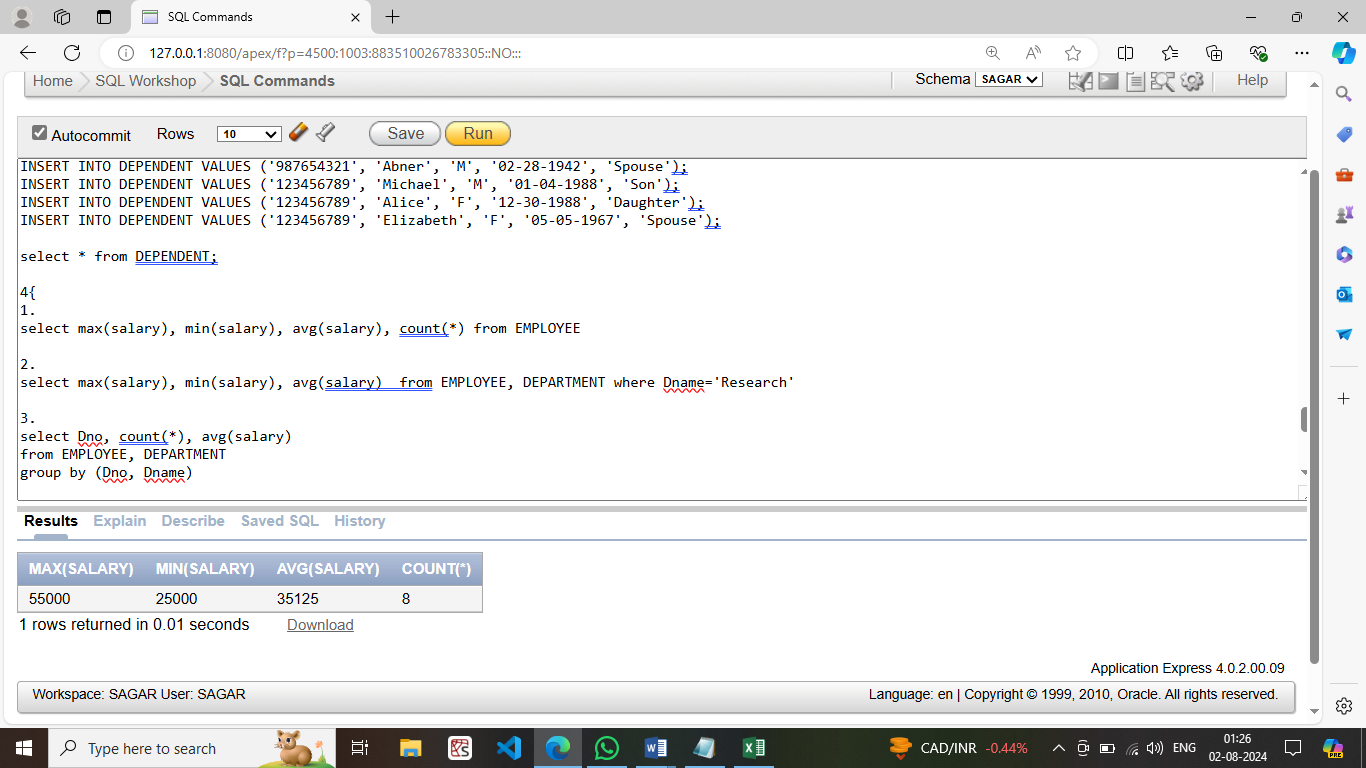


### ASSIGNMNENT :4

**Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING**

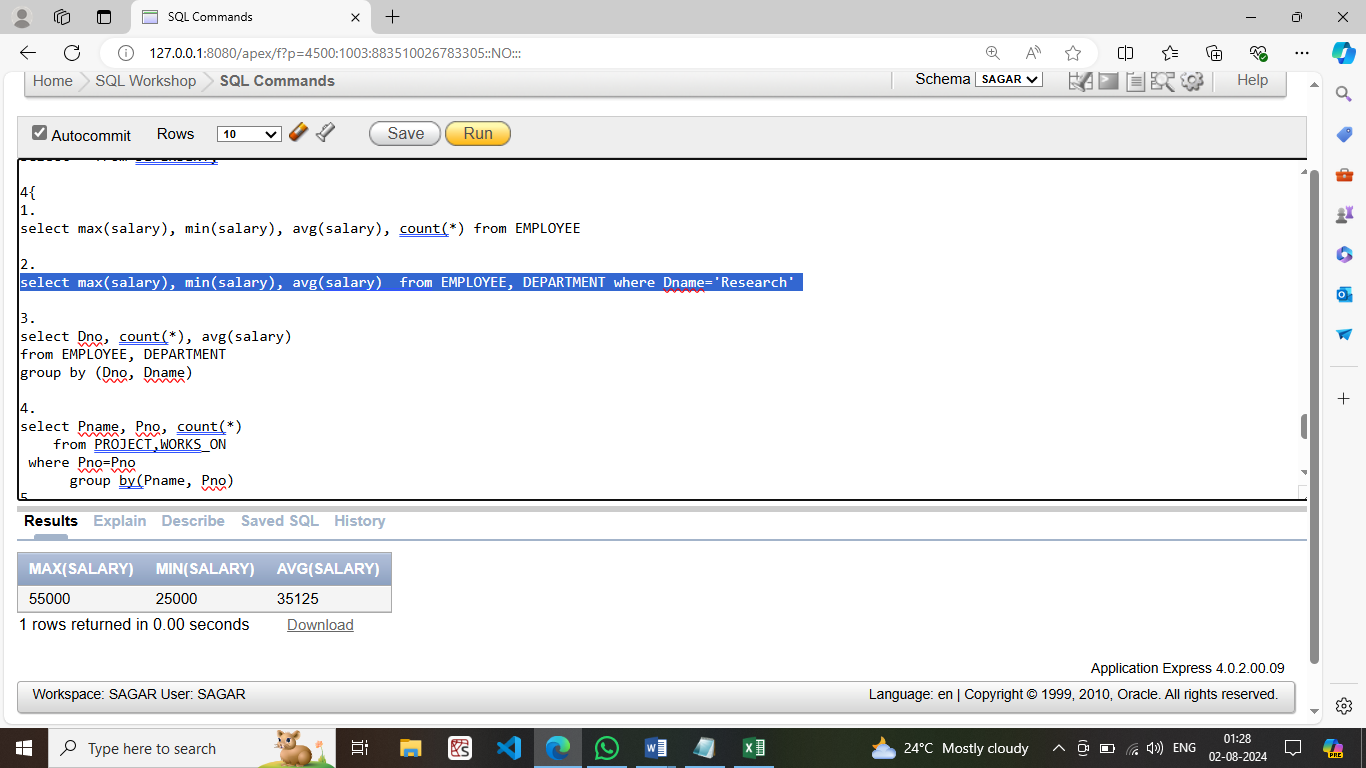
1. **Retrieve the maximum salary, the minimum salary and the average salary among all the employees also total number of employees in the company**

select max(salary), min(salary), avg(salary), count(\*) from EMPLOYEE



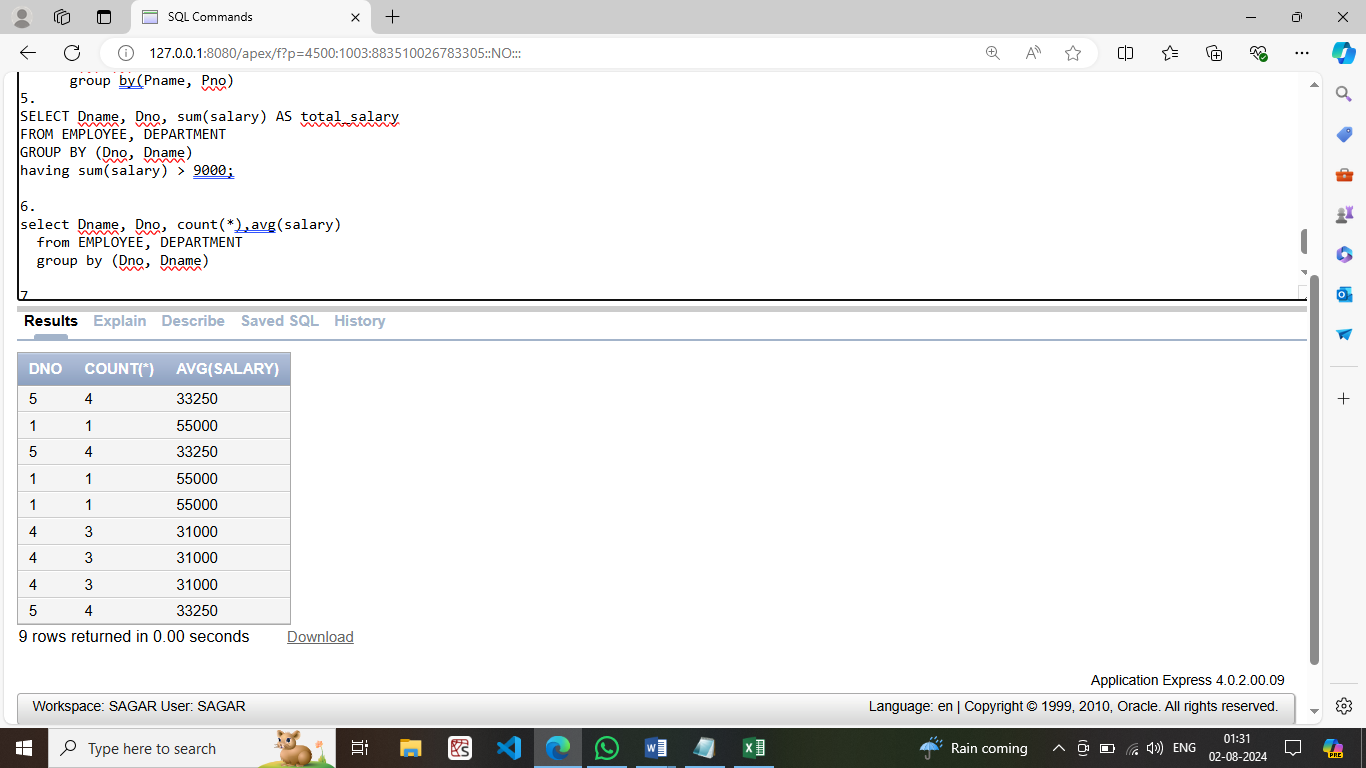
1. **Find the maximum salary, the minimum salary, and the average salary among employees who work for the 'Research' department.**

select max(salary), min(salary), avg(salary) from EMPLOYEE, DEPARTMENT where Dname='Research'



1. **For each department, retrieve the department number, the number of employees in the department, and their average salary.**

select Dno, count(\*), avg(salary) from EMPLOYEE, DEPARTMENT group by (Dno, Dname)



1. **For each project, retrieve the project number, project name, and the number of employees who work on that project.**

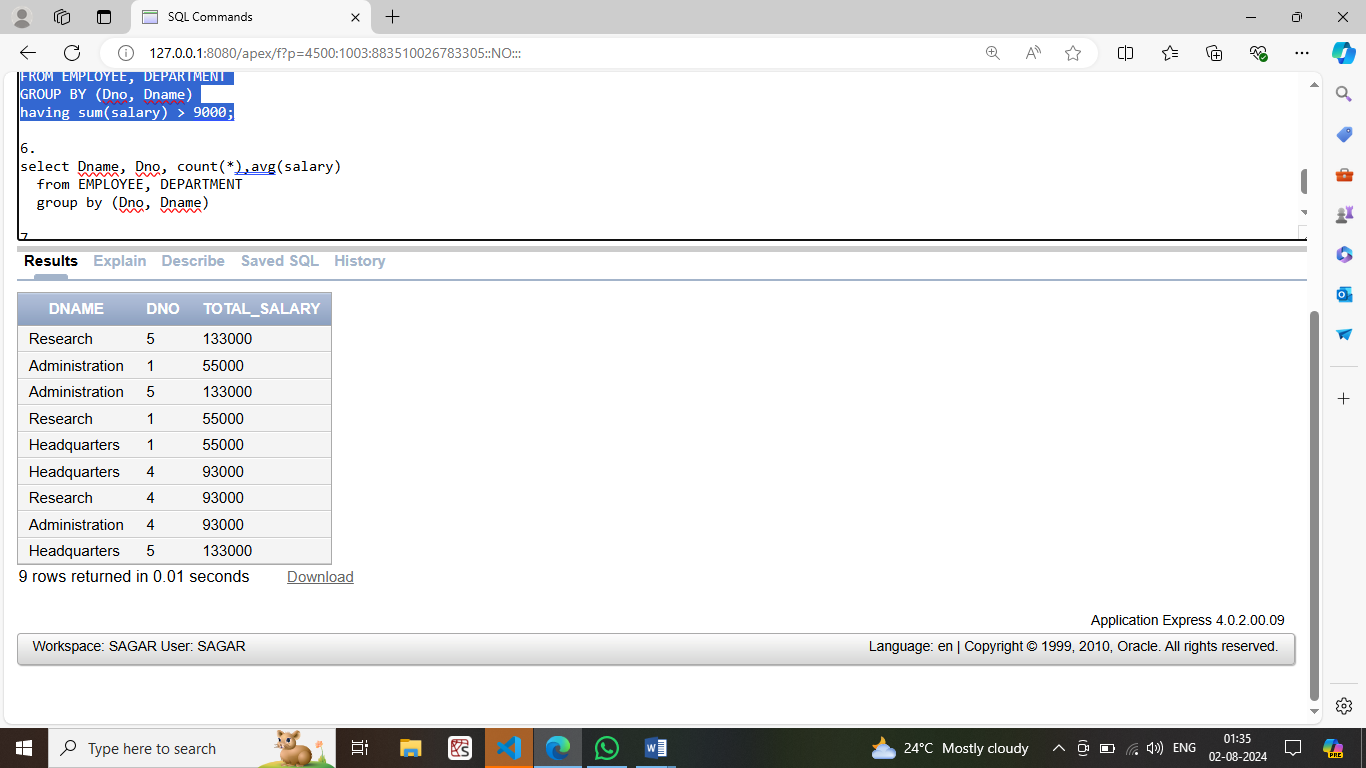
select Pname, Pno, count(\*) from PROJECT,WORKS\_ON where Pno=Pno group by(Pname, Pno)



1. **For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.**

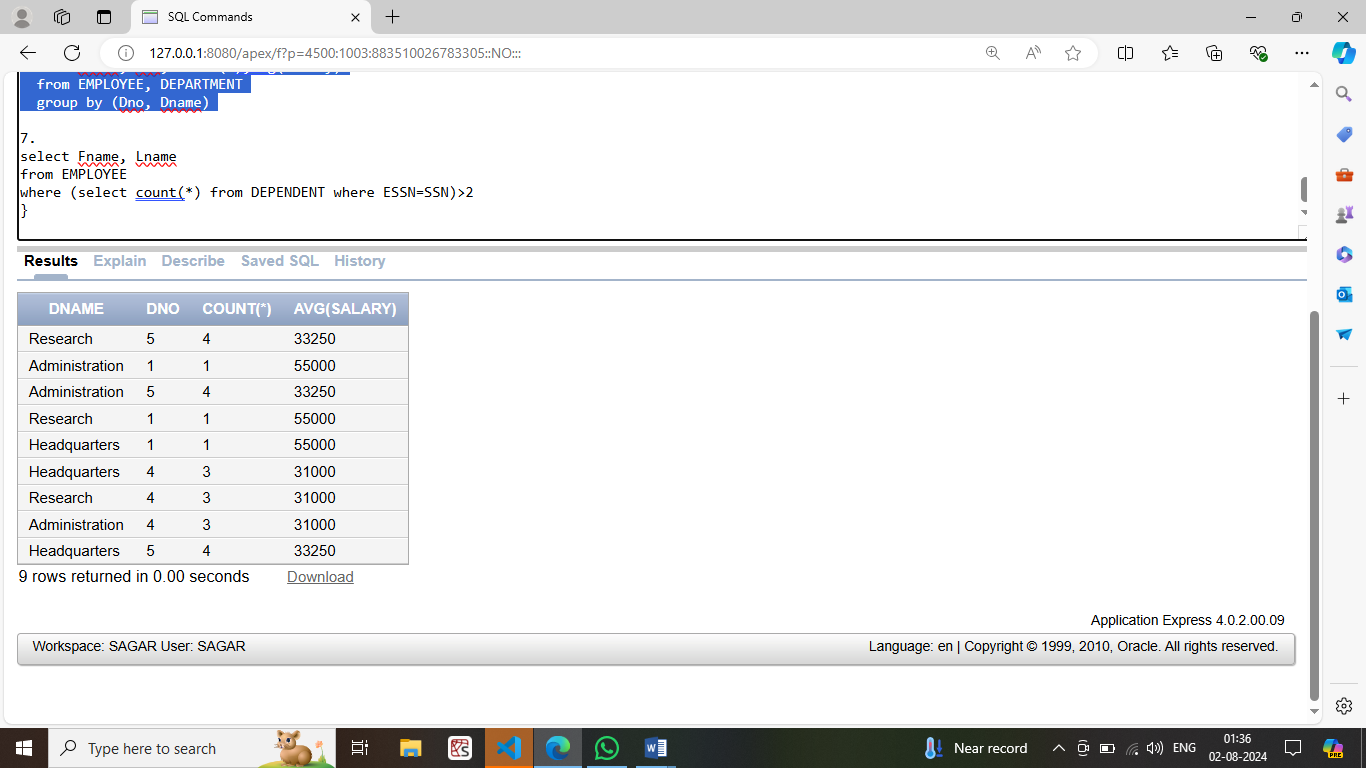
SELECT Dname, Dno, sum(salary) AS total\_salary FROM EMPLOYEE, DEPARTMENT

GROUP BY (Dno, Dname) having sum(salary) > 9000;



1. **For each department whose average employee salary is more than Rs 30000, retrieve the department name and number of employees working for that department.**

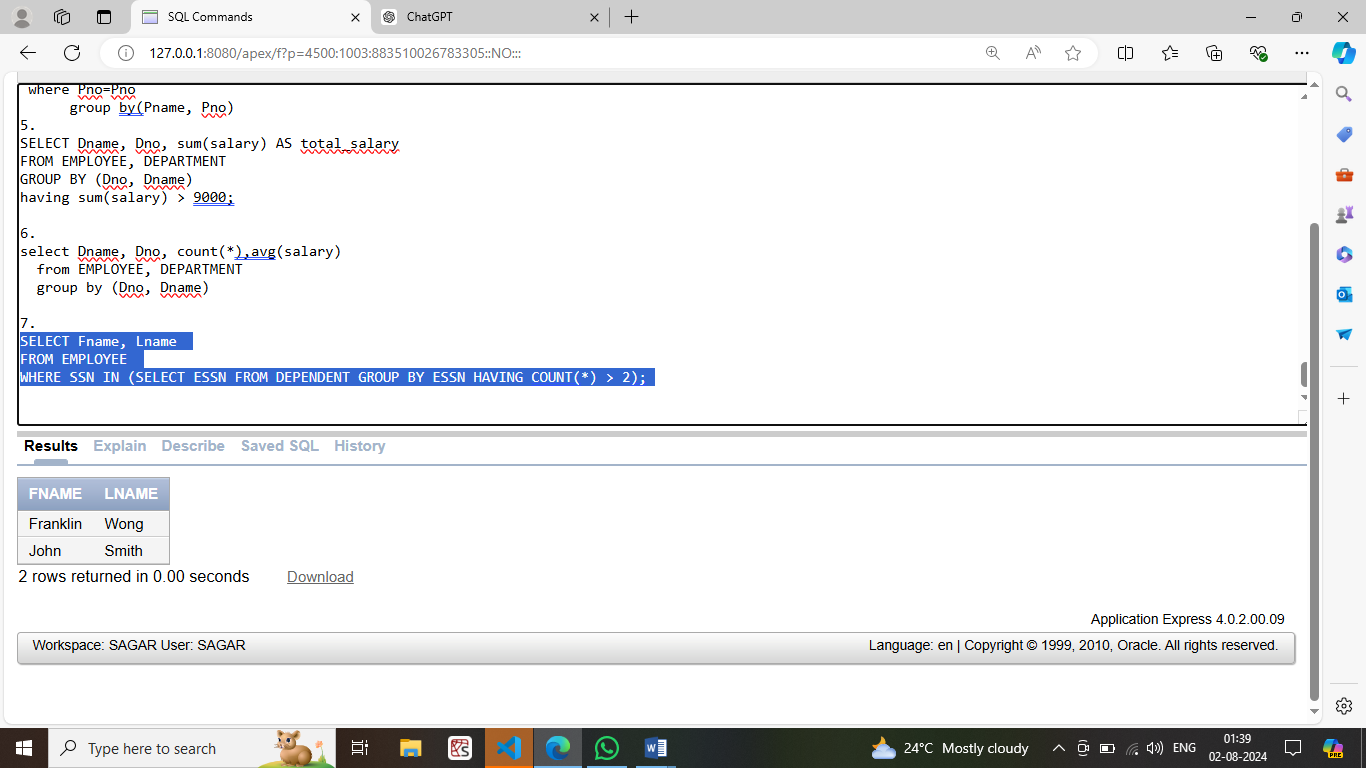
select Dname, Dno, count(\*), avg(salary) from EMPLOYEE, DEPARTMENT group by (Dno, Dname)



1. **For each employee who have more than two dependents, retrieve the name of employee and number of dependents**

select Fname, Lname from EMPLOYEE where

(select count(\*) from DEPENDENT where ESSN=SSN)>2 }



### ASSIGNMNENT :5

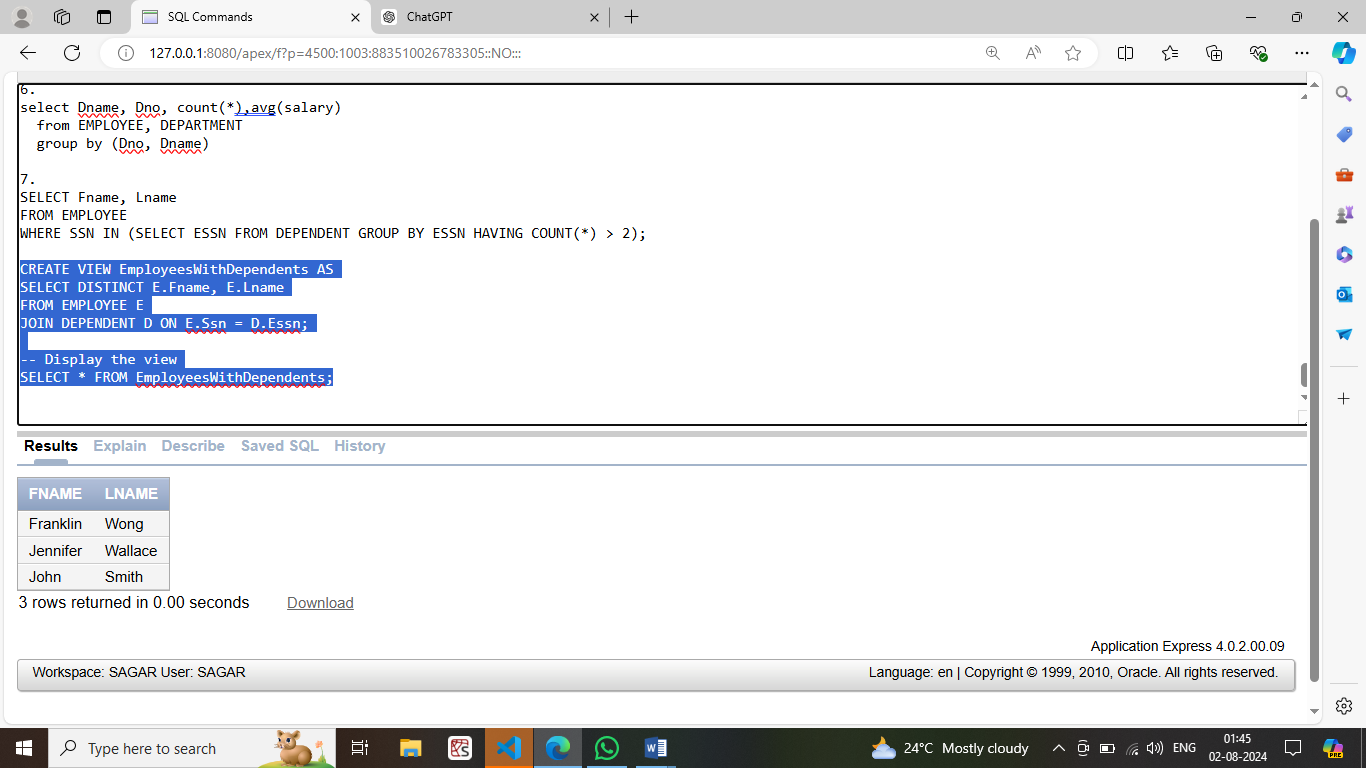
**TO CREATE AND DROPPING OF VIEW**

1. **Create view that has the list of names of employees who have dependent and also display the view;**

CREATE VIEW Employees With Dependents AS SELECT DISTINCT E.Fname, E.Lname

FROM EMPLOYEE E JOIN DEPENDENT D ON E.Ssn = D.Essn;

SELECT \* FROM EmployeesWithDependents;



1. **Create a view that has the employee name, supervisor name and employee salary for each employee who works for administration department and also display the view**

CREATE VIEW AdminEmployeeDetails AS

SELECT E.Fname AS EmployeeFirstName, E.Lname AS EmployeeLastName,

S.Fname AS SupervisorFirstName, S.Lname AS SupervisorLastName,

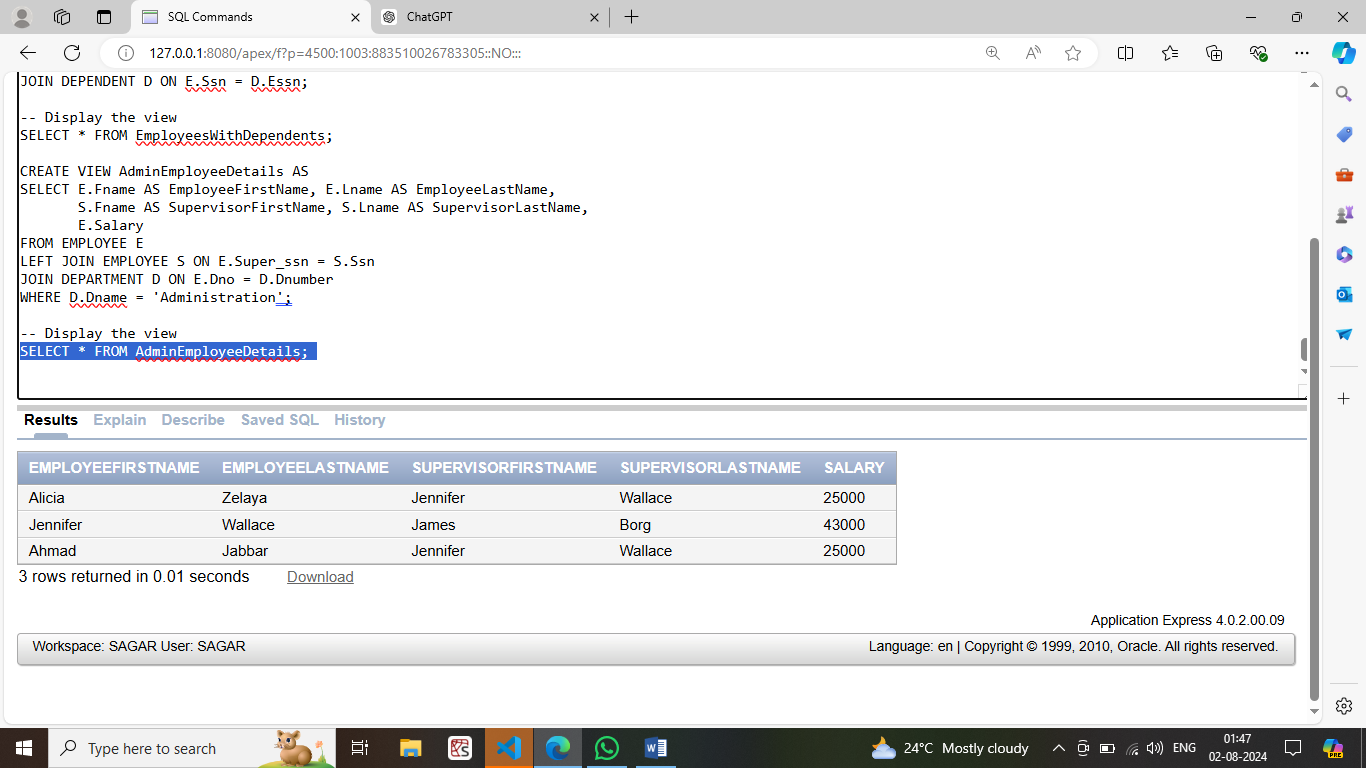
E.Salary

FROM EMPLOYEE E

LEFT JOIN EMPLOYEE S ON E.Super\_ssn = S.Ssn

JOIN DEPARTMENT D ON E.Dno = D.Dnumber

WHERE D.Dname = 'Administration';

SELECT \* FROM AdminEmployeeDetails;

1. **Create a view that has employee name, salary and project name which are located at Bellarie, Stafford use explicit set values.**

CREATE VIEW EmployeesOnSpecificProjects AS

SELECT E.Fname, E.Lname, E.Salary, P.Pname

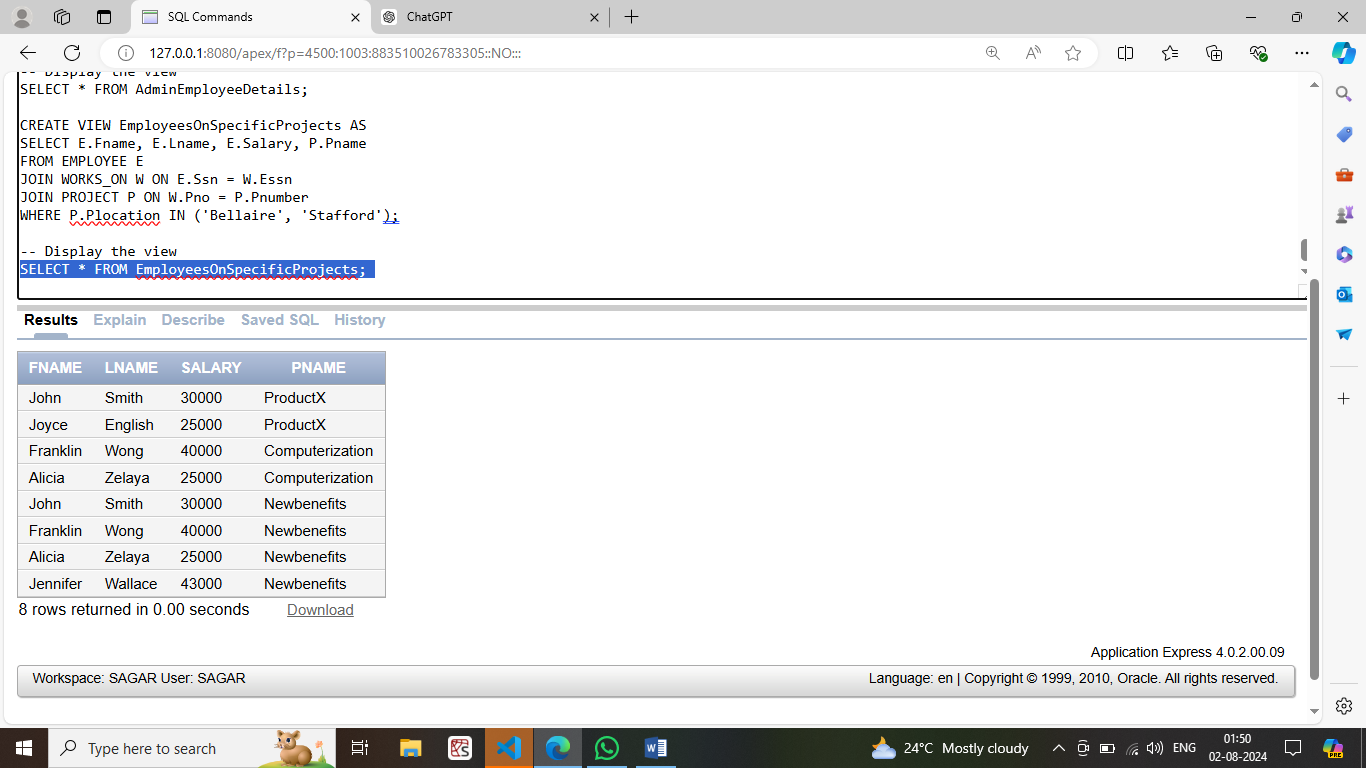
FROM EMPLOYEE E

JOIN WORKS\_ON W ON E.Ssn = W.Essn

JOIN PROJECT P ON W.Pno = P.Pnumber

WHERE P.Plocation IN ('Bellaire', 'Stafford');

SELECT \* FROM EmployeesOnSpecificProjects;



1. **Create view that has department name, manager name, project name and salary for every project and also display the view.**

CREATE VIEW DepartmentProjectDetails AS

SELECT D.Dname, M.Fname AS ManagerFirstName, M.Lname AS ManagerLastName,

P.Pname, E.Salary

FROM DEPARTMENT D

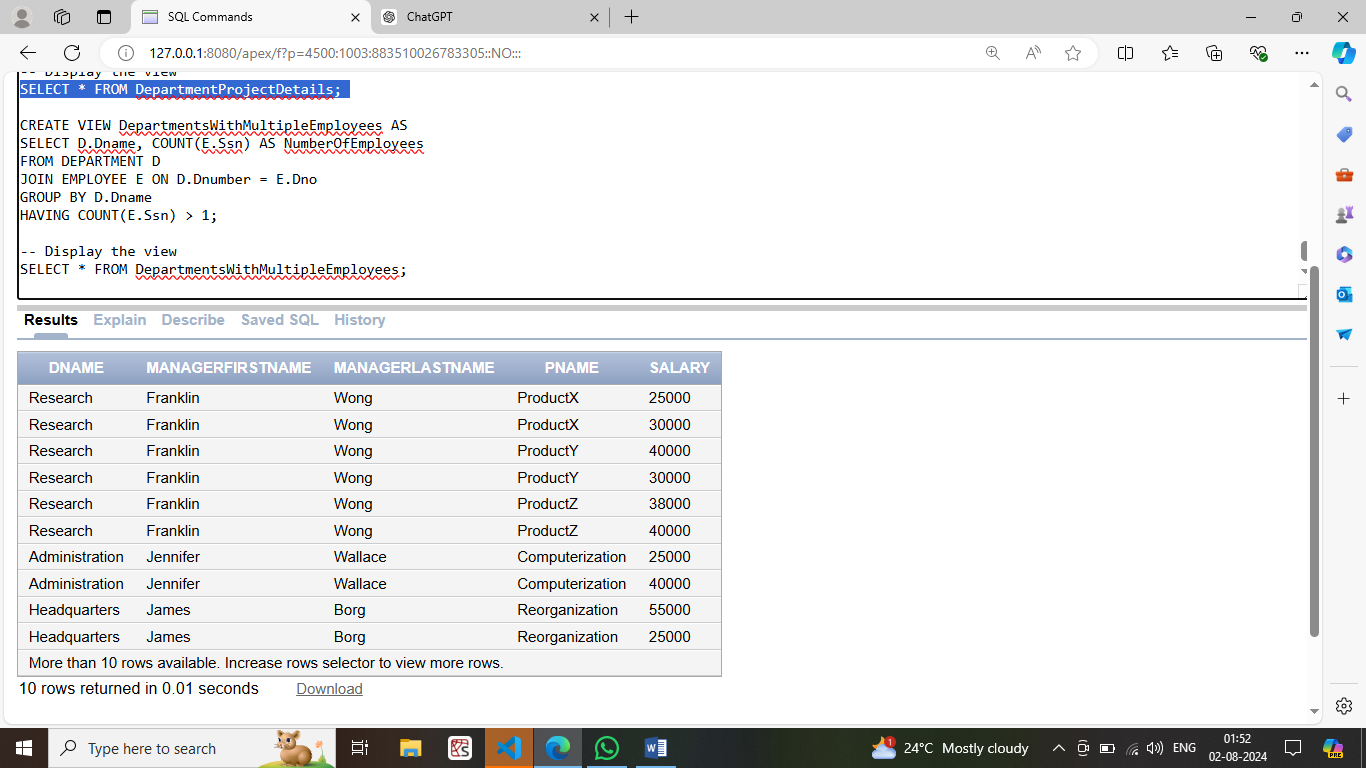
JOIN EMPLOYEE M ON D.Mgr\_ssn = M.Ssn

JOIN PROJECT P ON D.Dnumber = P.Dnum

JOIN WORKS\_ON W ON P.Pnumber = W.Pno

JOIN EMPLOYEE E ON W.Essn = E.Ssn;

SELECT \* FROM DepartmentProjectDetails;



1. **Create view that has the department name, number of employees working in the department, for each department with more than one employee working in the department display the view.**

CREATE VIEW DepartmentsWithMultipleEmployees AS

SELECT D.Dname, COUNT(E.Ssn) AS NumberOfEmployees

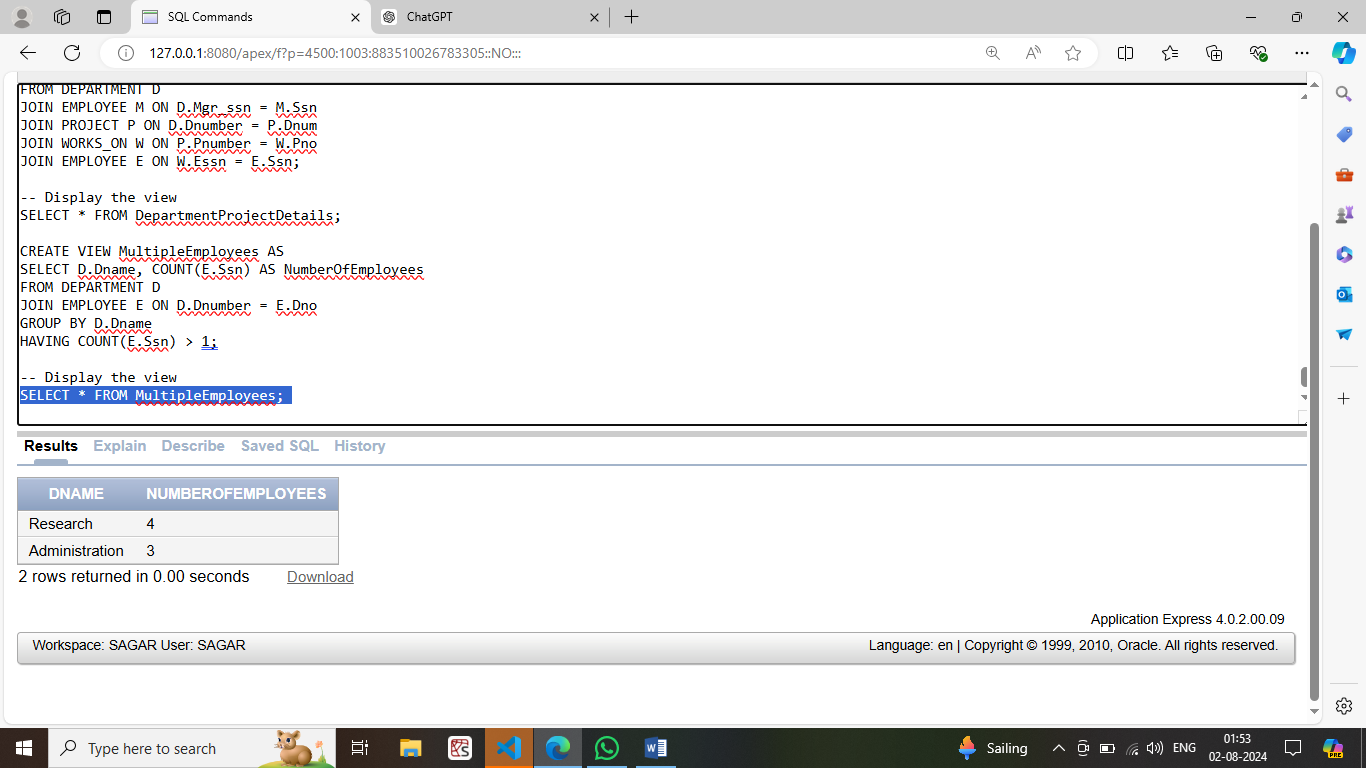
FROM DEPARTMENT D

JOIN EMPLOYEE E ON D.Dnumber = E.Dno

GROUP BY D.Dname

HAVING COUNT(E.Ssn) > 1;

SELECT \* FROM DepartmentsWithMultipleEmployees;



**Assignment 6:**

**Develop PL/SQL program using PROCEDURES.**

1. **Write a stored procedure to add new department record accept the input from user.**

CREATE OR REPLACE PROCEDURE addDepartment(

nname IN DEPARTMENT.Dname%TYPE,

ndno IN DEPARTMENT.Dnumber%TYPE,

ass IN DEPARTMENT.Mgr\_ssn%TYPE,

md IN DEPARTMENT.Mgr\_start\_date%TYPE

)

AS

BEGIN

INSERT INTO DEPARTMENT (Dname, Dnumber, Mgr\_ssn, Mgr\_start\_date)

VALUES (nname, ndno, ass, md);

END;

/

------------------------------------- Test the procedure------------------------------------------

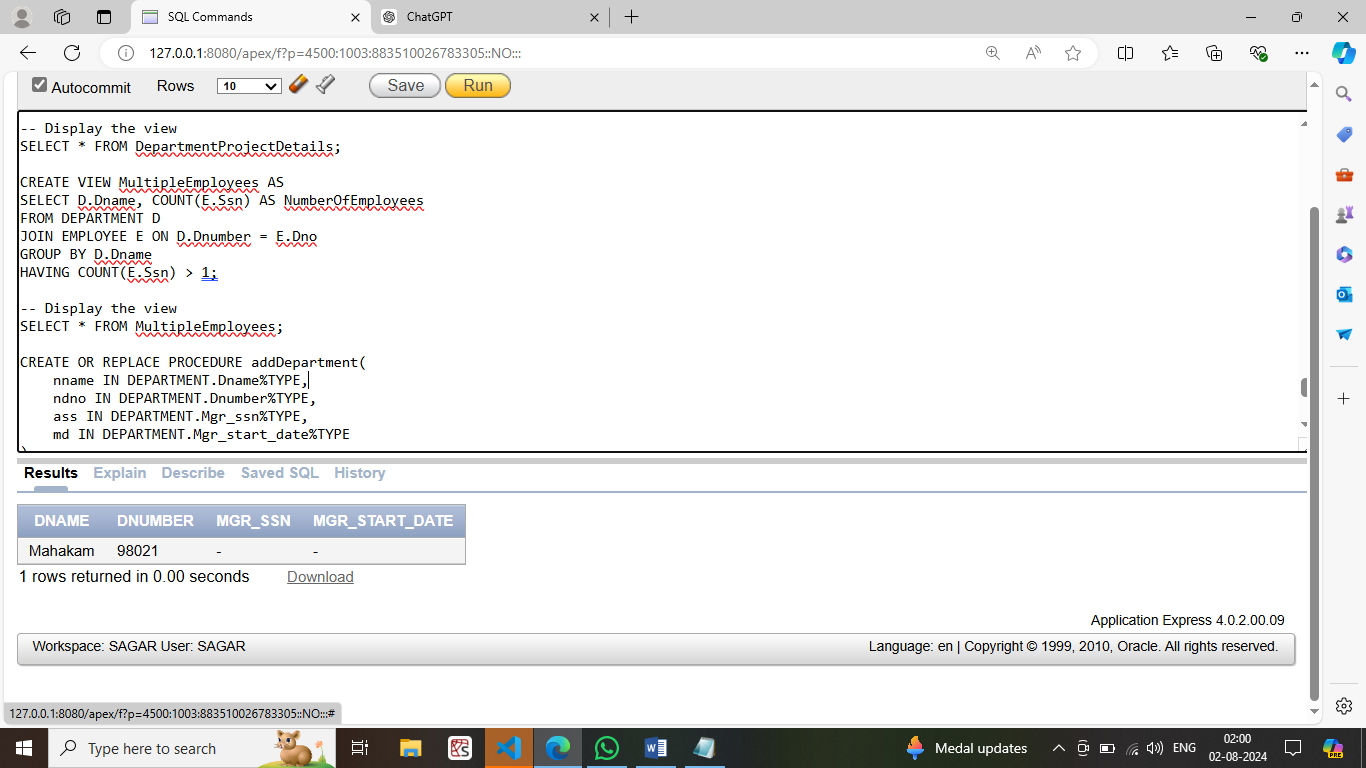
BEGIN

addDepartment('Mahakam', 98021, NULL, NULL);

END;

/

SELECT \* FROM DEPARTMENT WHERE Dname = 'Mahakam';



1. **Write a stored procedure to display the employee name and salary if name of the working department and ssn of the employee is passed**

CREATE OR REPLACE PROCEDURE displayEmployee(

mssn IN EMPLOYEE.Ssn%TYPE,

mDname IN DEPARTMENT.Dname%TYPE,

mname OUT EMPLOYEE.Fname%TYPE,

msal OUT EMPLOYEE.Salary%TYPE

)

IS

BEGIN

SELECT E.Fname, E.Salary

INTO mname, msal

FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.Dno = D.Dnumber

WHERE D.Dname = mDname AND E.Ssn = mssn;

END;

DECLARE

mname EMPLOYEE.Fname%TYPE;

msal EMPLOYEE.Salary%TYPE;

BEGIN

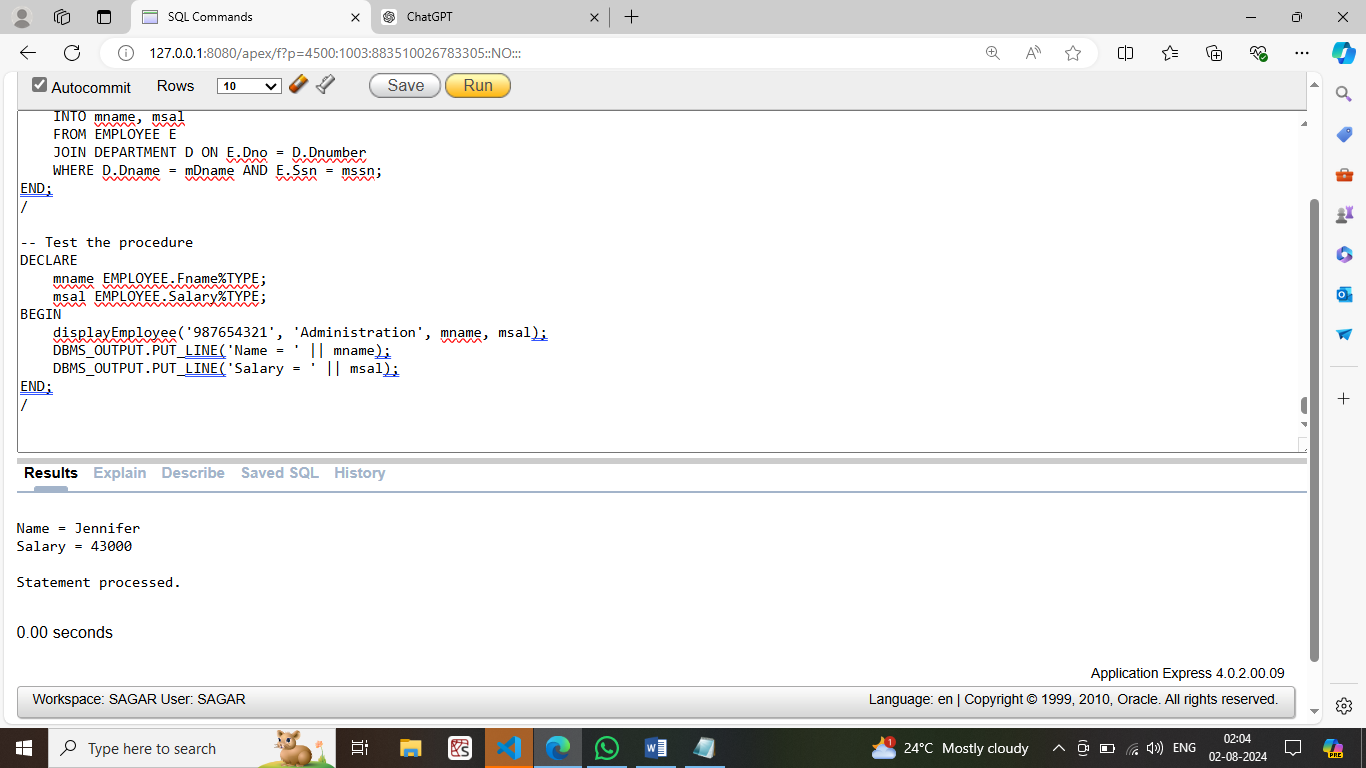
displayEmployee('987654321', 'Administration', mname, msal);

DBMS\_OUTPUT.PUT\_LINE('Name = ' || mname);

DBMS\_OUTPUT.PUT\_LINE('Salary = ' || msal);

END;

/



**3.Write a stored procedure to update the salary (by 10%) of the employee who is working on the project named ProductX and hours more than 30.**

create or replace procedure emp\_sal as

begin

update EMPLOYEE set Salary=Salary+Salary\*0.1 where Ssn in(select Ssn from EMPLOYEE,PROJECT,WORKS\_ON where Pname='ProductX' and Pno=Pnumber and Hours>30 and Ssn=ESSN);

end;

/

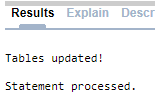
begin

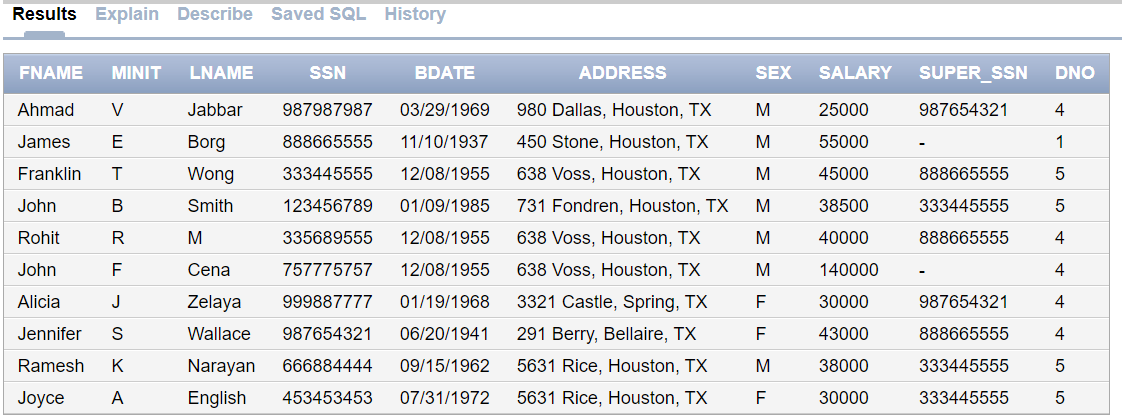
emp\_sal();

dbms\_output.put\_line('Tables updated!');

end;

/





**Assignment 7:**

**Develop PL/SQL program using FUNCTIONS.**

1. **Write a stored function to display salary of employee if ssn of employee is passed as an argument.**

CREATE OR REPLACE FUNCTION getEmployeeSalary(emessn IN EMPLOYEE.Ssn%TYPE) RETURN DECIMAL

AS

tor DECIMAL(10, 2);

BEGIN

SELECT Salary INTO tor FROM EMPLOYEE WHERE Ssn = emessn;

RETURN tor;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN NULL; -- Handle case where SSN does not exist

END;

/

DECLARE

c DECIMAL(10, 2);

BEGIN

c := getEmployeeSalary('123456789');

IF c IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Salary: ' || c);

ELSE

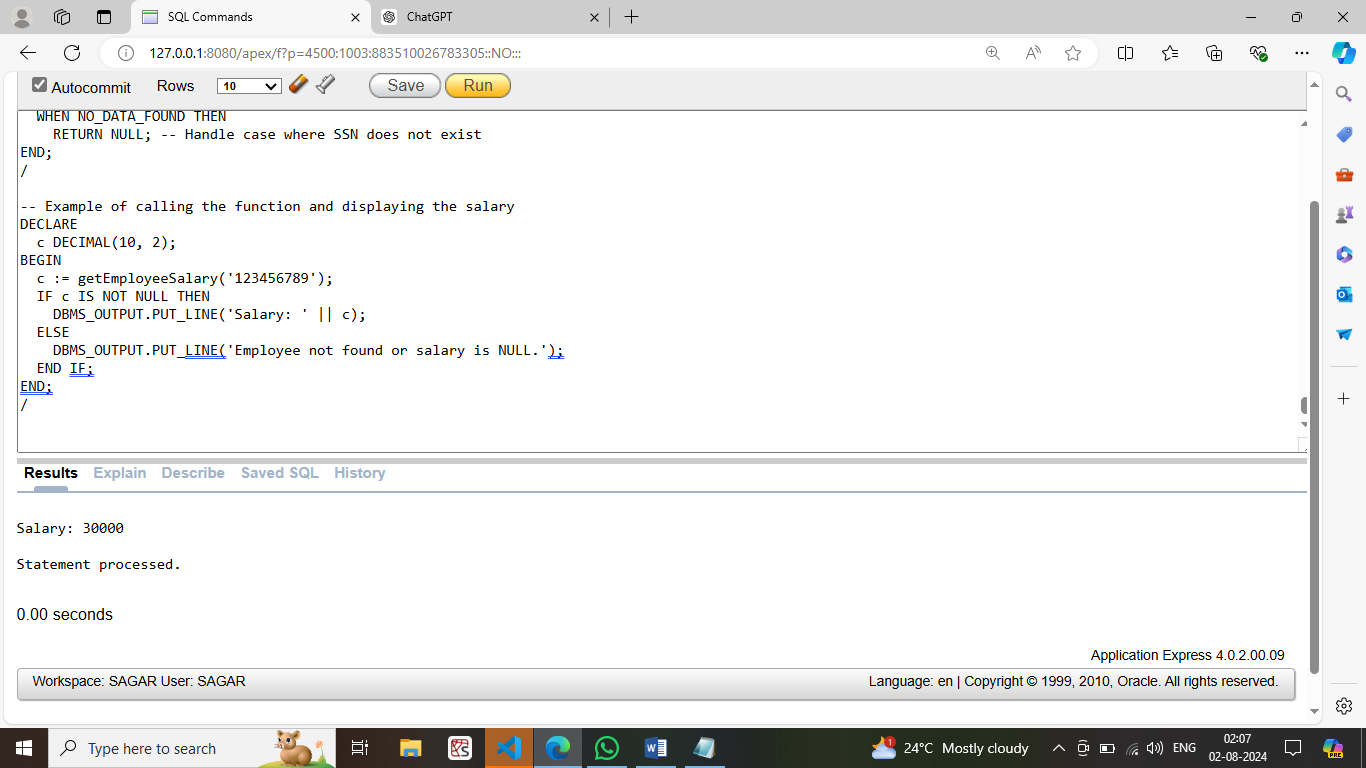
DBMS\_OUTPUT.PUT\_LINE('Employee not found or salary is NULL.');

END IF;

END;

/

**Output:**



### ASSIGNMNENT :8

**Develop PL/SQL programs using CURSOR.**

1. **Using implicit cursor update the salary of the employee are working on project number (1,10) and also display the number employees salary updated. If number of rows updated more than 5 then print many employees are benefited otherwise few benifited.**

DECLARE

v\_num\_updated INTEGER;

BEGIN

UPDATE EMPLOYEE

SET Salary = Salary + 5000

WHERE Ssn IN (SELECT Essn FROM WORKS\_ON WHERE Pno IN (1, 10));

v\_num\_updated := SQL%ROWCOUNT;

IF v\_num\_updated > 5 THEN

DBMS\_OUTPUT.PUT\_LINE('Many employees are benefitted. Number of employees: ' || v\_num\_updated);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Few employees are benefitted. Number of employees: ' || v\_num\_updated);

END IF;

END;

/

**Output:**

Few employees are benefitted. Number of employees: 4

1 row(s) updated.

1. **Define a EXPLICIT cursor to display the ssn, fname and dependent names of employees.**

DECLARE

CURSOR display IS

SELECT E.Ssn, E.Fname, D.Dependent\_name

FROM EMPLOYEE E

JOIN DEPENDENT D ON E.Ssn = D.Essn;

c\_SSN EMPLOYEE.Ssn%TYPE;

c\_Fname EMPLOYEE.Fname%TYPE;

c\_Dname DEPENDENT.Dependent\_name%TYPE;

BEGIN

OPEN display;

LOOP

FETCH display INTO c\_SSN, c\_Fname, c\_Dname;

EXIT WHEN display%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('SSN: ' || c\_SSN);

DBMS\_OUTPUT.PUT\_LINE('First Name: ' || c\_Fname);

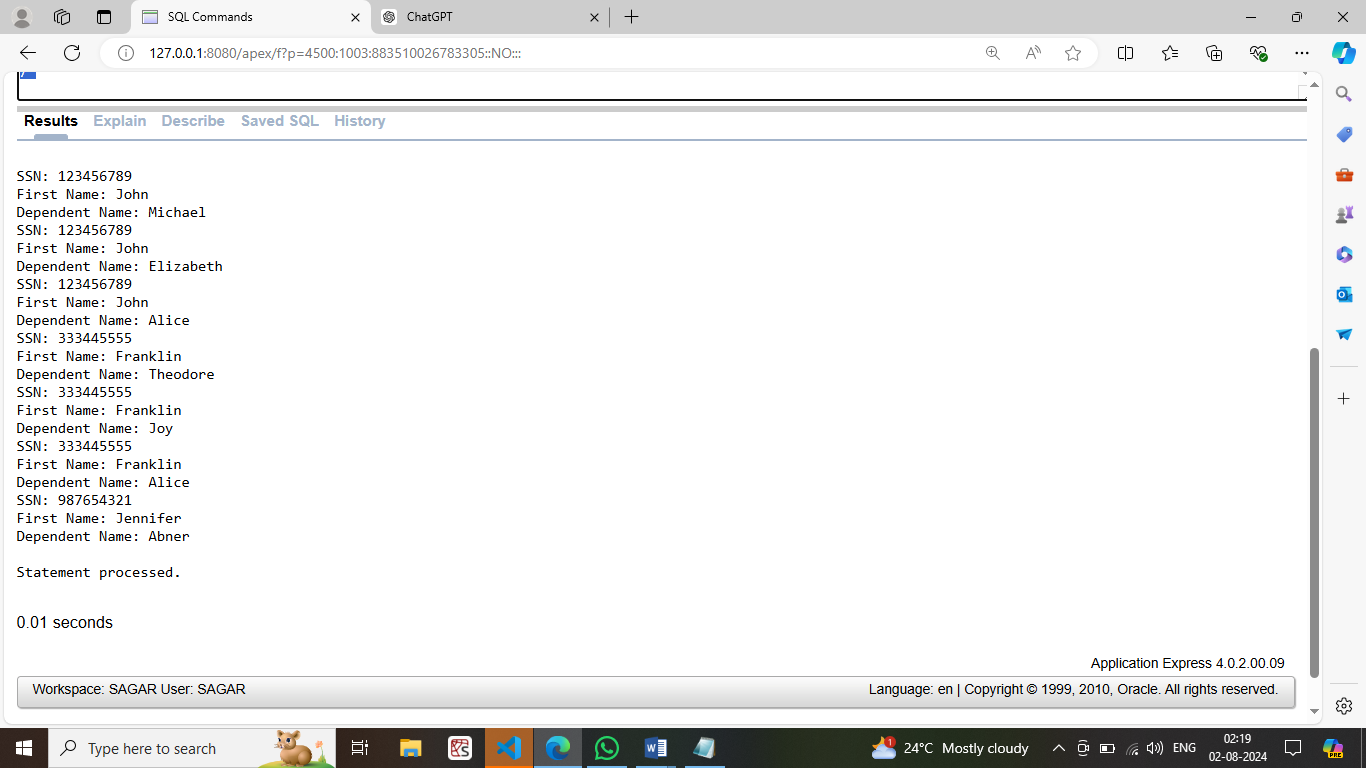
DBMS\_OUTPUT.PUT\_LINE('Dependent Name: ' || c\_Dname);

END LOOP;

CLOSE display;

END;

/



**ASSIGNMNENT :9**

**Develop PL/SQL programs using Triggers**

**1. Create trigger to display total number of employees working for department research in organization/company before inserting new record .**

create or replace trigger sal\_vibha

before update of salary ON EMPLOYEE

for each row

declare

diff EMPLOYEE.salary%type;

BEGIN

diff:=:new.salary-:old.salary;

DBMS\_OUTPUT.PUT\_LINE('old is ==' || :old.salary);

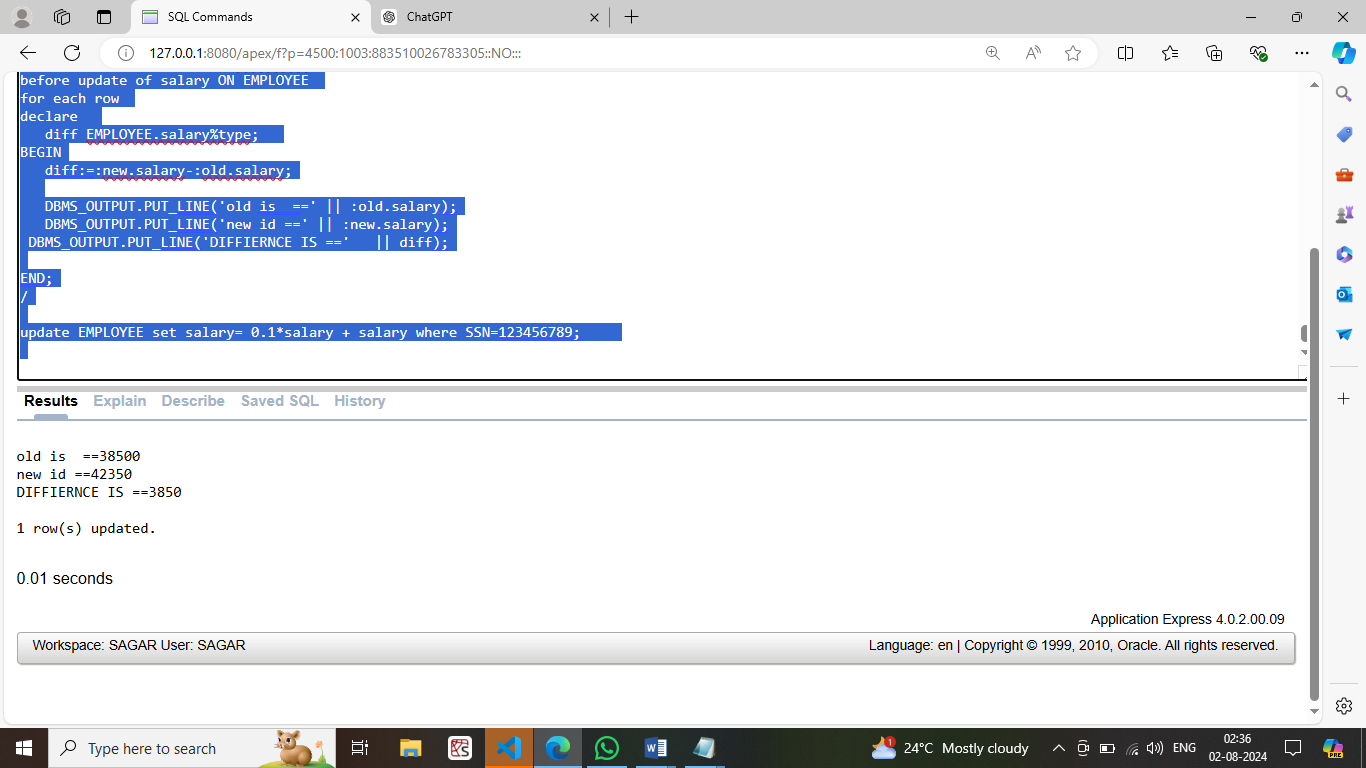
DBMS\_OUTPUT.PUT\_LINE('new id ==' || :new.salary);

DBMS\_OUTPUT.PUT\_LINE('DIFFIERNCE IS ==' || diff);

END;

/

update EMPLOYEE set salary= 0.1\*salary + salary where SSN=123456789;



**2. Write a trigger to display the salary difference between the old and new salary before updating.**

CREATE OR REPLACE TRIGGER simpi

BEFORE INSERT ON EMPLOYEE

FOR EACH ROW

DECLARE

c int;

v\_count int;

BEGIN

SELECT Dnumber INTO c FROM DEPARTMENT WHERE Dname = 'Admin';

SELECT COUNT(\*) as ss INTO v\_count from EMPLOYEE WHERE Dno= c;

DBMS\_OUTPUT.PUT\_LINE('Total number of employees in administration department: ' || v\_count);

END;

/

INSERT INTO EMPLOYEE VALUES ('John', 'F', 'Cena', '757775757', '12-08-1955', '638 Voss, Houston, TX', 'M', 140000, NULL, 4);

**Output:**

