Practical 1 – ER Diagram and Schema Design

# Create the tables designed in ER diagram and schema and insert the valid values in the table

CREATE DATABASE COMPANY\_DATABASE;

## USE COMPANY\_DATABASE;

CREATE TABLE EMPLOYEE(

FNAME VARCHAR(15) NOT NULL,

## MINIT CHAR,

LNAME VARCHAR(15) NOT NULL,

SSN CHAR(9) NOT NULL,

BDATE DATE,

ADDRESS VARCHAR(30),

## SEX CHAR, SALARY DECIMAL(10,2), SUPER\_SSN CHAR(9),

DNO INT NOT NULL,

PRIMARY KEY(SSN)

);

CREATE TABLE DEPARTMENT(

DNAME VARCHAR(15) NOT NULL,

DNUMBER INT NOT NULL,

MGR\_SSN CHAR(9) NOT NULL,

## MGR\_START\_DATE DATE,

PRIMARY KEY(DNUMBER),

UNIQUE(DNAME),

FOREIGN KEY(MGR\_SSN) REFERENCES EMPLOYEE(SSN) );

## CREATE TABLE DEPT\_LOCATIONS( DLOCATION VARCHAR(15) NOT NULL, DNUMBER INT NOT NULL, PRIMARY KEY(DNUMBER, DLOCATION), FOREIGN KEY(DNUMBER) REFERENCES DEPARTMENT(DNUMBER) );

CREATE TABLE PROJECT(

PNAME VARCHAR(15) NOT NULL,

## PNUMBER INT NOT NULL, PLOCATION VARCHAR(15),

DNUM INT NOT NULL,

PRIMARY KEY(PNUMBER),

UNIQUE(PNAME),

## FOREIGN KEY(DNUM) REFERENCES DEPARTMENT(DNUMBER)

);

CREATE TABLE WORKS\_ON(

ESSN CHAR(9) NOT NULL,

PNO INT NOT NULL,

HOURS DECIMAL(3,1) NOT NULL,

PRIMARY KEY(ESSN, PNO),

FOREIGN KEY(ESSN) REFERENCES EMPLOYEE(SSN),

FOREIGN KEY(PNO) REFERENCES PROJECT(PNUMBER)

);

CREATE TABLE DEPENDENT(

ESSN CHAR(9) NOT NULL,

## DEPENDENT\_NAME VARCHAR(15) NOT NULL, SEX CHAR, BDATE DATE, RELATIONSHIP VARCHAR(8), PRIMARY KEY(ESSN, DEPENDENT\_NAME),

FOREIGN KEY(ESSN) REFERENCES EMPLOYEE(SSN)

);

(Contd.) Insertion of valid values in the tables:

## USE COMPANY\_DATABASE;

INSERT INTO EMPLOYEE VALUES

('John','B','Smith','123456789','1965-01-09','731 Fondren, Housten, TX','M',30000,'333445555',5),

('Franklin','T','Wong','333445555','1955-12-08','638 Voss, Housten,

TX','M',40000,'888665555',5),

('Alicia','J','Zolaya','999887777','1968-01-19','3321 Castlo, Spring,

TX','F',25000,'987654321',4),

('Jennifer','S','Wallace','987654321','1941-06-20','291 Berry, Beilaire,

TX','F',43000,'888665555',4),

('Ramesh','K','Narayan','666884444','1962-09-15','975 Fire Oak, Humble,

TX','M',38000,'333445555',5),

('Joyce','A','English','453453453','1972-07-31','5631 Rice, Housten, TX','F',25000,'333445555',5),

('Ahmad','V','Jabbar','987987987','1969-03-29','980 Dallas, Housten,

TX','M',25000,'987654321',4),

('James','E','Borg','888665555','1937-11-10','450 Stone, Housten,

TX','M',55000,NULL,1);

INSERT INTO DEPARTMENT VALUES

('Research',5,'333445555','1988-05-22'),

('Administration',4,'987654321','1995-01-01'),

('Headquaters',1,'888665555','1981-06-19');

## INSERT INTO DEPT\_LOCATIONS VALUES

('Houston',1),

('Stafford',4),

('Bellaire',5),

('Sugarland',5),

('Houston',5);

INSERT INTO PROJECT VALUES

('ProductX',1,'Bellaire',5),

('ProductY',2,'Sugarland',5),

('ProductZ',3,'Houston',5),

('Computerization',10,'Stafford',4),

('Reorganization',20,'Houston',1),

('Newbenefits',30,'Stafford',4);

INSERT INTO WORKS\_ON VALUES

('123456789',1,32.5),

('123456789',2,7.5),

('666884444',3,40),

('453453453',1,20), ('453453453',2,20), ('333445555',2,10),

('333445555',3,10),

('333445555',10,10),

('333445555',20,10),

('999887777',30,30),

('999887777',10,10),

('987987987',10,35), ('987987987',30,5),

('987654321',30,20),

('987654321',20,15),

('888665555',20,0);

INSERT INTO DEPENDENT VALUES

('333445555','Alice','F','1986-04-05','Daughter'),

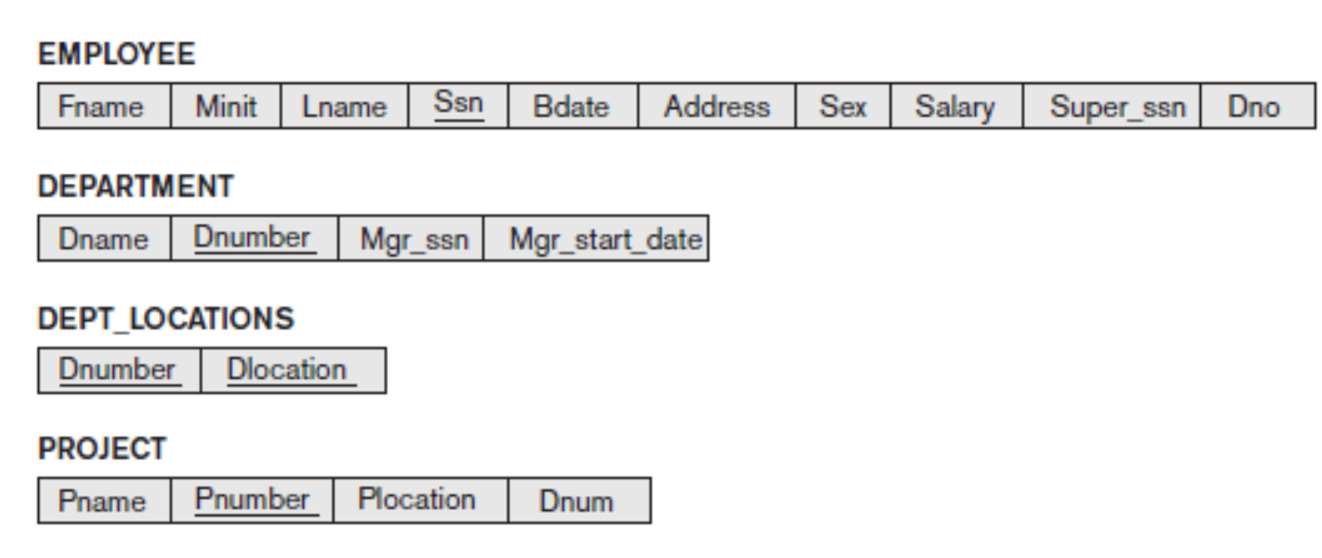
('333445555','Theodore','M','1983-10-25','Son'), ('333445555','Joy','F','1958-05-03','Spouse'),

('987654321','Abner','M','1942-02-28','Spouse'),

('123456789','Michael','M','1988-01-04','Son'),

('123456789','Alice','F','1988-12-30','Daughter'),

('123456789','Elizabeth','F','1967-05-05','Spouse');



# **Practical 2** - Selection and projection-based queries **2**

1. Display all the tables created in the database.

SELECT \* FROM DEPARTMENT

SELECT \* FROM DEPENDENT

## SELECT \* FROM DEPT\_LOCATIONS SELECT \* FROM EMPLOYEE

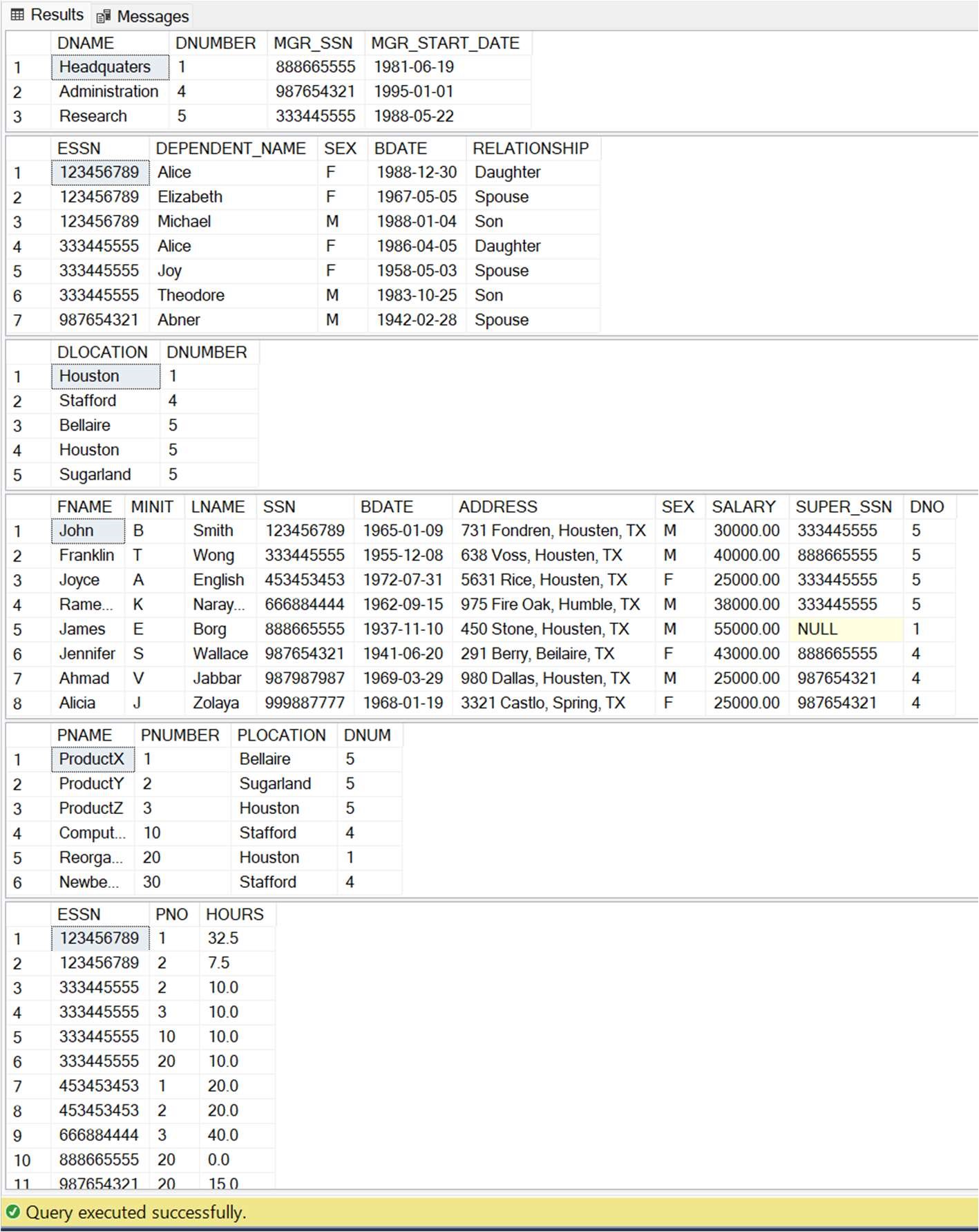
SELECT \* FROM PROJECT

SELECT

\*

FROM

WORKS\_ON



1. Display all Employee’s SSNs

SELECT SSN FROM EMPLOYEE;

1. Display Employee Name, Employee Number working in department number 5.

SELECT FNAME, LNAME, SSN FROM EMPLOYEE WHERE DNO = 5;

1. Retrieve the birth date and address of the employee(s) whose name is ‘John B. Smith’.

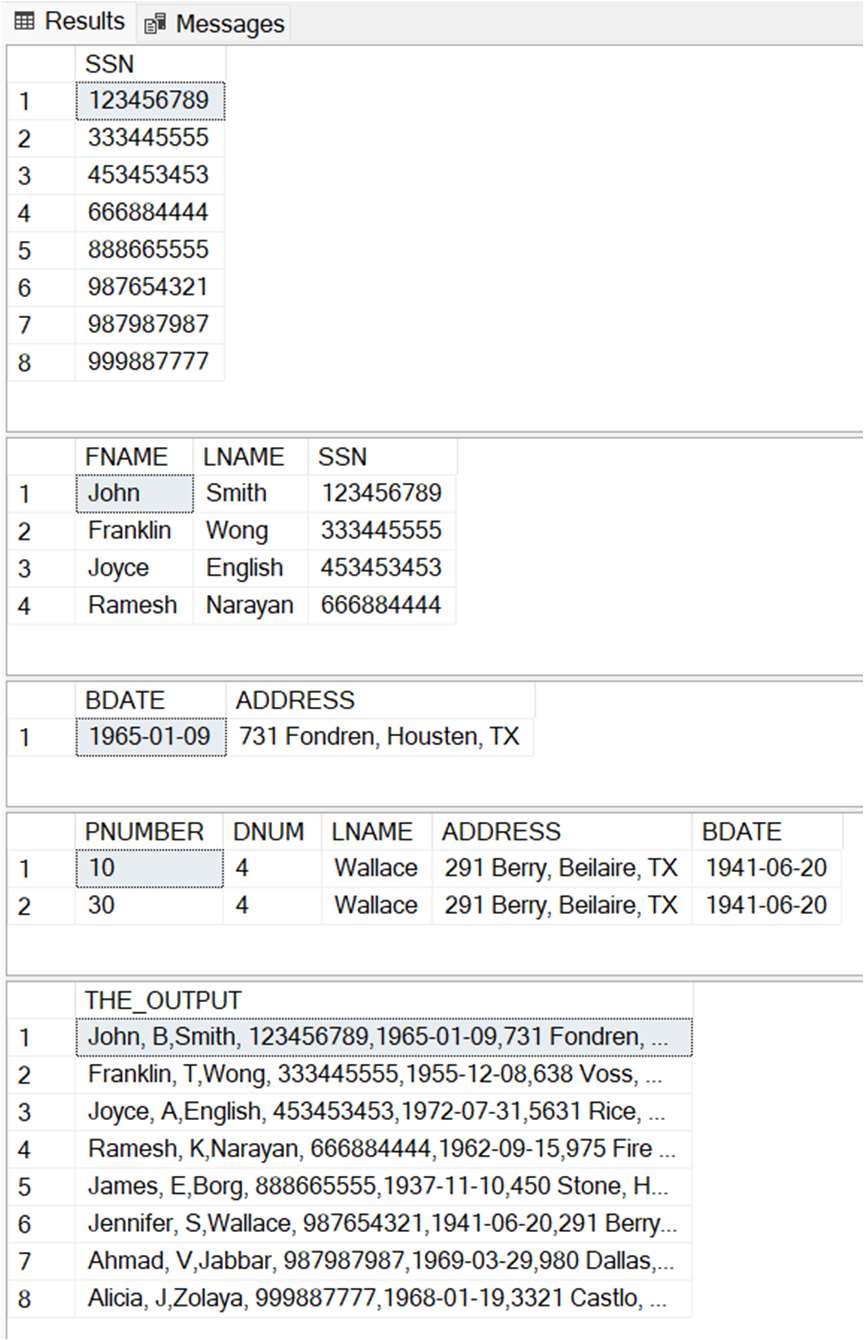
## SELECT BDATE, ADDRESS FROM EMPLOYEE WHERE FNAME = 'John' AND MINIT = 'B' AND LNAME = 'Smith';

5. For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

## SELECT P.PNUMBER, P.DNUM, E.LNAME, E.ADDRESS, E.BDATE FROM PROJECT P, DEPARTMENT D, EMPLOYEE E WHERE P.PLOCATION = 'Stafford' AND P.DNUM = D.DNUMBER AND D.MGR\_SSN = E.SSN;

6. Display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE\_OUTPUT.

## SELECT CONCAT(FNAME, ', ', MINIT, ',', LNAME, ', ', SSN, ',' ,BDATE, ',' , ADDRESS, ',' , SEX, ',' , SALARY, ',' , SALARY, ',' , SUPER\_SSN, ',' , DNO) AS THE\_OUTPUT FROM EMPLOYEE;



1. For each employee, retrieve the employee’s first and last name and the first and last name of his or her immediate supervisor.

SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE as E, EMPLOYEE as S WHERE E.SUPER\_SSN=S.SSN;

1. Display Name of all the employees where the third letter of their name is ‘A’.

## SELECT FNAME FROM EMPLOYEE WHERE FNAME LIKE '\_\_A%';

9. Display Name of all employees either have two R‘s or have two A‘s in their name and are either in Dept No = 7 or their Manger‘s Employee No = ‘123456789’.

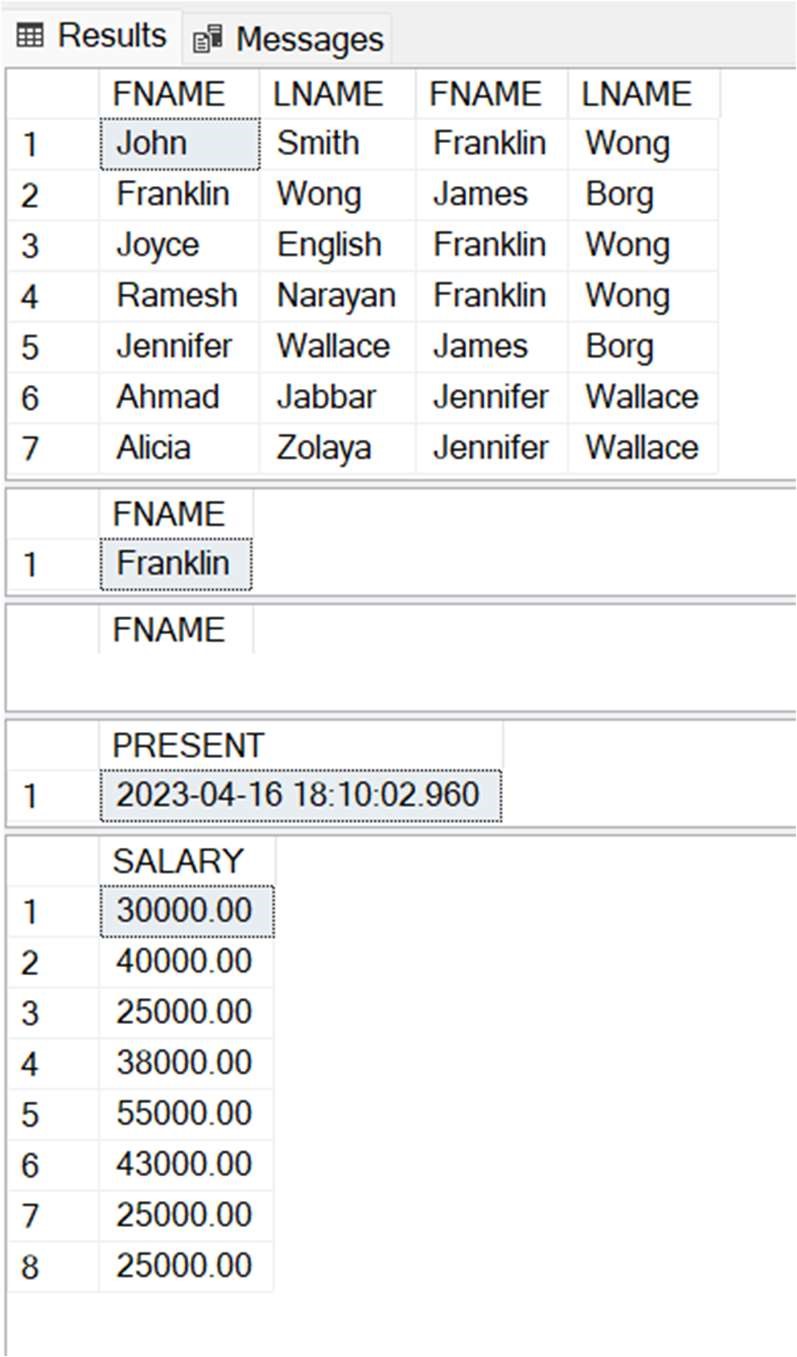
## SELECT FNAME FROM EMPLOYEE WHERE (FNAME LIKE '%A%A%' OR FNAME LIKE '%R%R%') AND (DNO = 5 OR SUPER\_SSN = 123456789);

1. Display the Current Date.

SELECT GETDATE();

1. Retrieve the salary of every employee

## SELECT SALARY from EMPLOYEE;



# Practical 3 - Selection and projection-based queries

1. Display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with ‘J‘, ‘A‘ and ‘M‘.

SELECT CONCAT(UPPER(SUBSTRING(FNAME,1,1)),LOWER(SUBSTRING(FNAME,2,14))) AS NAME,

LEN(FNAME) FROM EMPLOYEE WHERE (FNAME LIKE 'J%' OR FNAME LIKE 'A%' OR FNAME LIKE 'M%');

1. Display all valid combinations of Employee’s Ssn and Department name in the database.

SELECT E.SSN, D.DNAME FROM EMPLOYEE AS E, DEPARTMENT AS D WHERE E.DNO = D.DNUMBER;

3. Display the Employee Name and Salary of all the employees earning more than 25000.

SELECT FNAME, LNAME, SALARY FROM EMPLOYEE WHERE SALARY > 25000;

4. Retrieve the name and address of all employees who work for the ‘Research’ department.

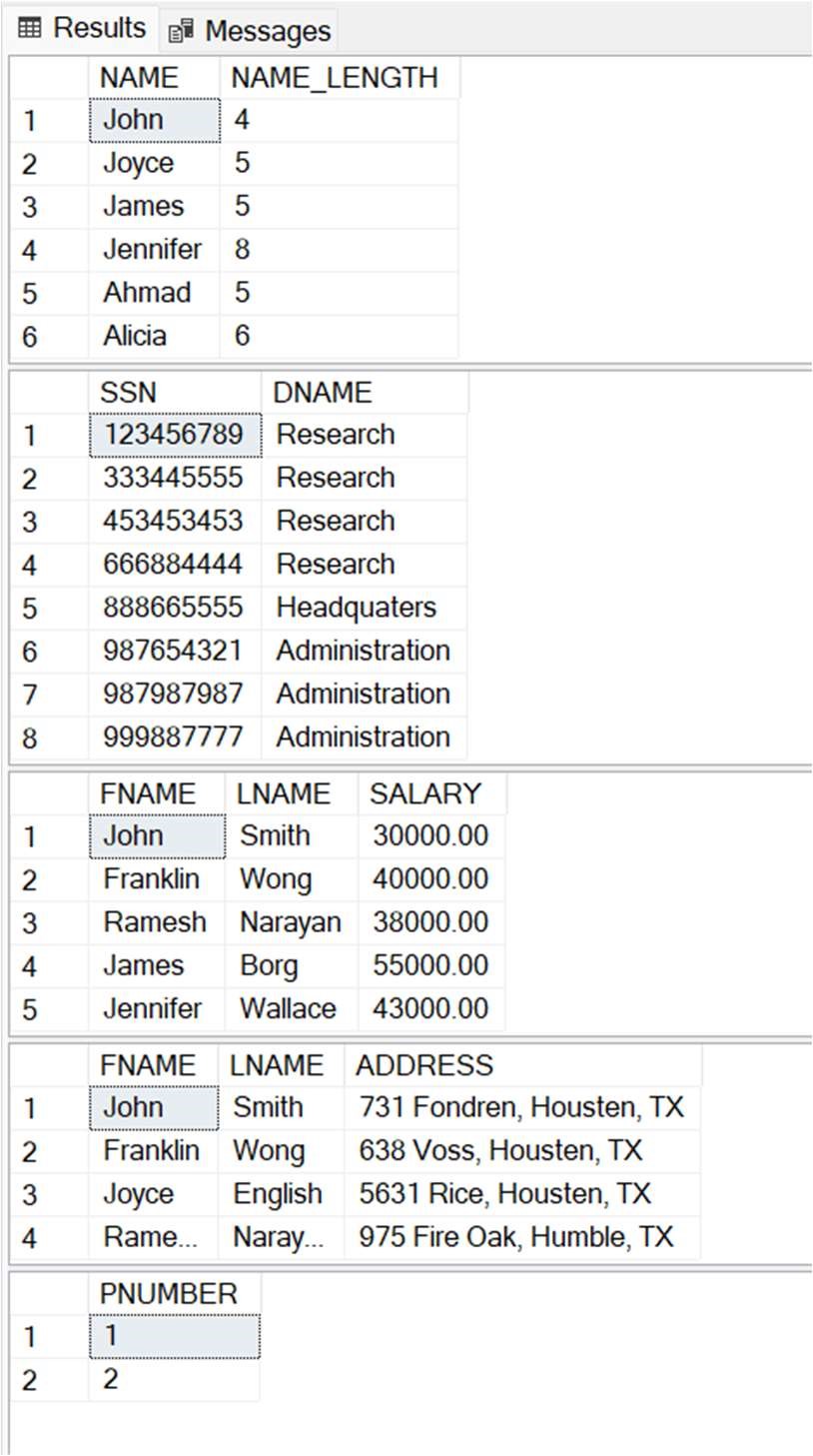
## SELECT E.FNAME, E.LNAME, E.ADDRESS FROM EMPLOYEE AS E, DEPARTMENT AS D WHERE E.DNO = D.DNUMBER AND D.DNAME = 'Research';

5. Make a list of all project numbers for projects that involve an employee whose last name is ‘Smith’, either as a worker or as a manager of the department that controls the project.

(SELECT DISTINCT PNUMBER FROM PROJECT, DEPARTMENT, EMPLOYEE WHERE DNUM = DNUMBER AND MGR\_SSN = SSN AND LNAME = 'Smith')

UNION

## (SELECT DISTINCT PNUMBER FROM PROJECT, WORKS\_ON, EMPLOYEE WHERE PNUMBER = PNO AND ESSN = SSN AND LNAME = 'Smith');



1. Retrieve all employees whose address is in Houston, Texas

SELECT \* FROM EMPLOYEE WHERE ADDRESS LIKE '%Housten, TX%';

1. Find all employees who were born during the 1950s.

## SELECT \* FROM EMPLOYEE WHERE BDATE >= '1950-01-01' AND BDATE < '1960-01-01';

1. Show the resulting salaries if every employee working on the ‘ProductX’ project is given a 10% raise.

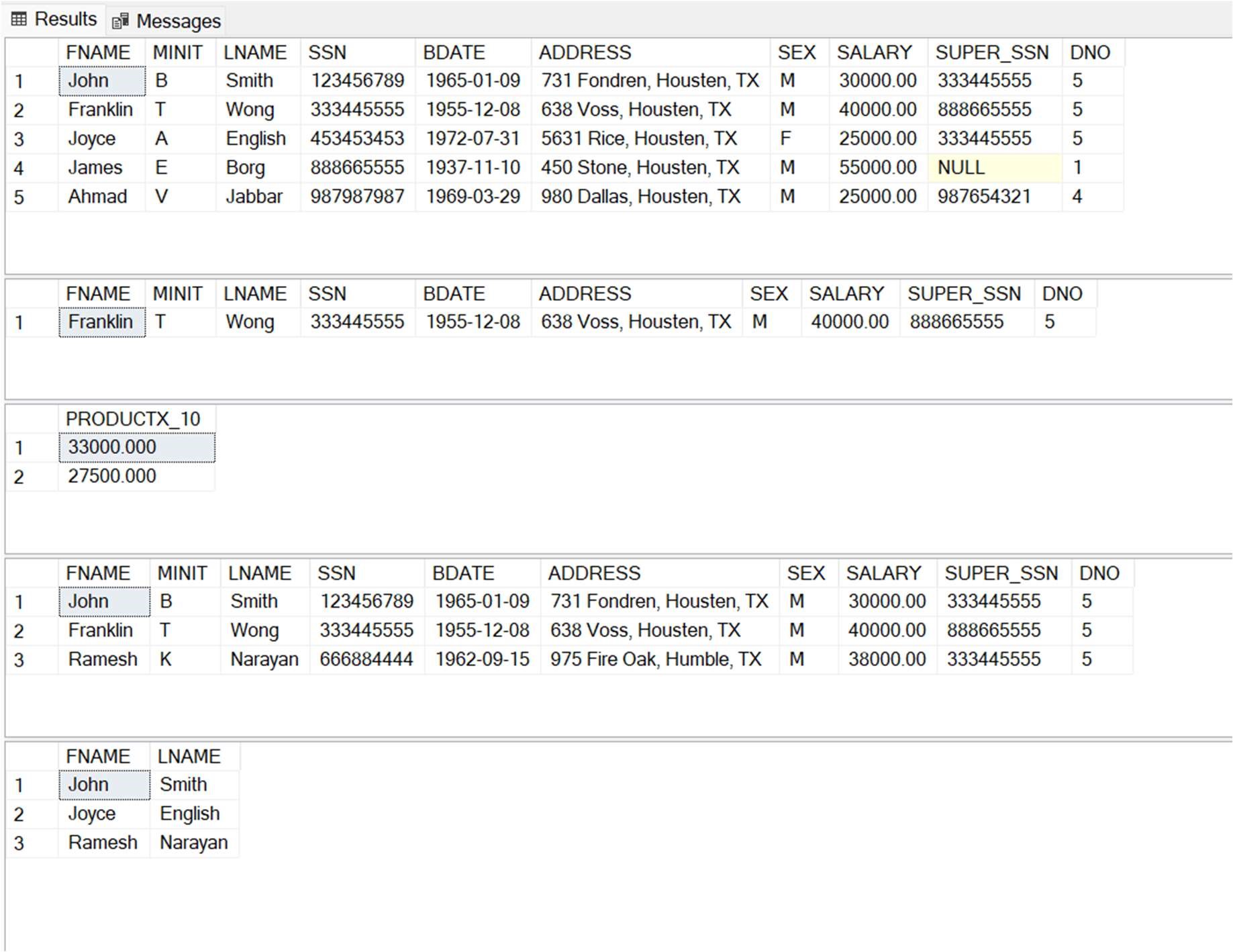
SELECT E.SALARY\*1.1 AS PRODUCTX\_10 FROM EMPLOYEE AS E, PROJECT AS P, WORKS\_ON AS W WHERE P.PNAME = 'ProductX' AND W.PNO = P.PNUMBER AND W.ESSN = E.SSN;

1. Retrieve all employees in department 5 whose salary is between $30,000 and $40,000.

SELECT \* FROM EMPLOYEE WHERE DNO = 5 AND SALARY >= 30000 AND SALARY <= 40000;

1. Find the names of all employees who are directly supervised by ‘Franklin Wong’.

## SELECT E1.FNAME, E1.LNAME FROM EMPLOYEE AS E1, EMPLOYEE AS E2 WHERE E1.SUPER\_SSN = E2.SSN AND E2.FNAME = 'Franklin' AND E2.LNAME = 'Wong';



# Practical 4 – Some Complex queries

1. Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, then first name.

## SELECT D.DNAME, E.LNAME, E.FNAME, P.PNAME FROM DEPARTMENT AS D, EMPLOYEE AS E, WORKS\_ON AS W, PROJECT AS P WHERE D.DNUMBER = E.DNO AND E.SSN = W.ESSN AND W.PNO = P.PNUMBER ORDER BY D.DNAME, E.LNAME, E.FNAME;

1. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.

SELECT FNAME, LNAME FROM EMPLOYEE WHERE DNO = 5 AND SSN IN (SELECT W.ESSN FROM WORKS\_ON AS W WHERE W.HOURS > 10 AND W.PNO = (SELECT P.PNUMBER FROM PROJECT AS P WHERE P.PNAME = 'ProductX'));

1. List the names of all employees who have a dependent with the same first name as themselves.

SELECT E.FNAME, E.LNAME FROM EMPLOYEE AS E WHERE E.SSN IN (SELECT ESSN FROM DEPENDENT WHERE E.FNAME = DEPENDENT\_NAME);

1. Create a table EMPLOYEE\_BACKUP to back up the EMPLOYEE table

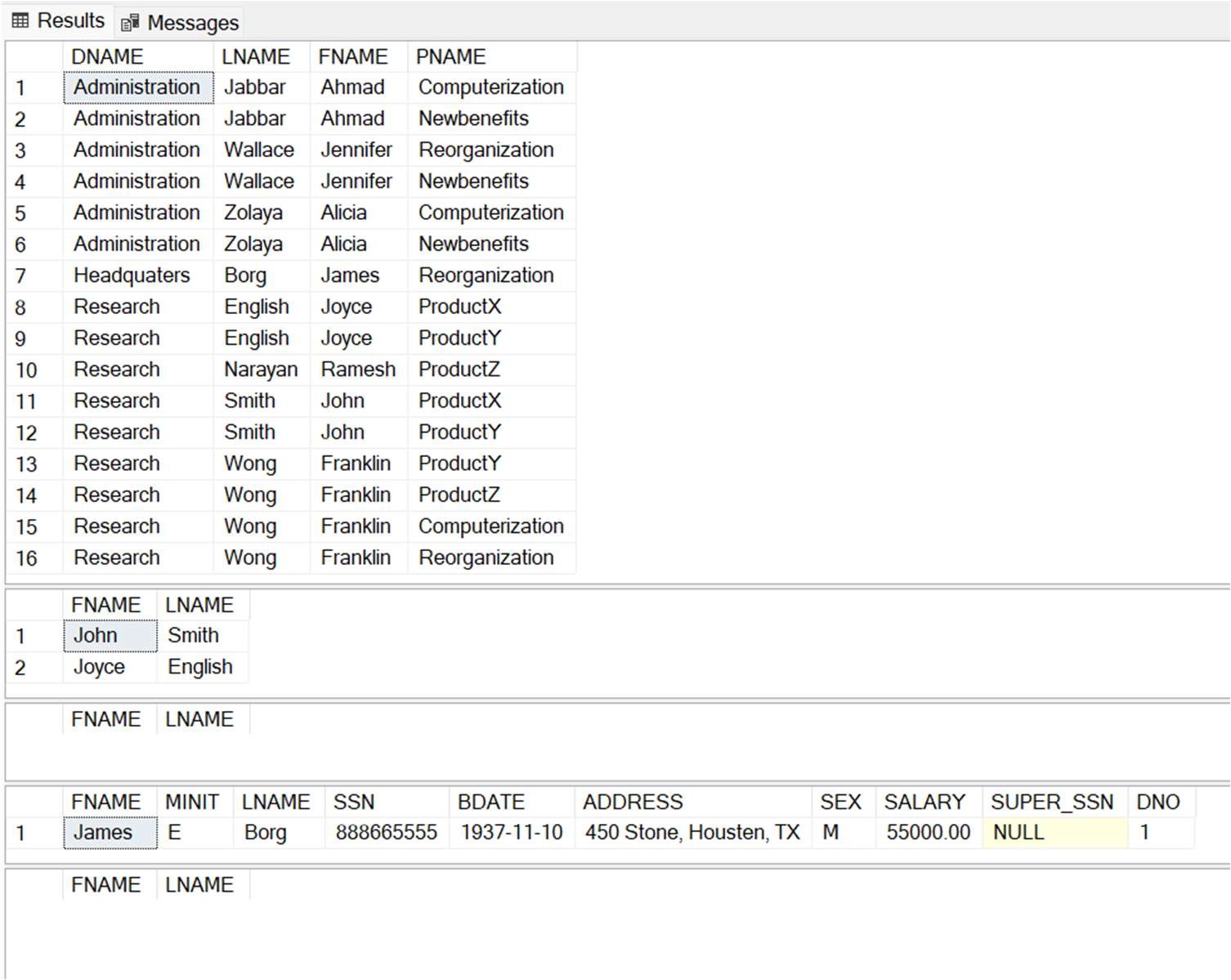
SELECT \* INTO EMPLOYEE\_BACKUP FROM EMPLOYEE;

1. Retrieve the names of all employees who do not have supervisors.

## SELECT \* FROM EMPLOYEE WHERE SUPER\_SSN IS NULL;

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

## SELECT E.FNAME, E.LNAME FROM EMPLOYEE AS E WHERE E.SSN IN (SELECT ESSN FROM DEPENDENT WHERE E.FNAME = DEPENDENT\_NAME AND E.SEX = SEX);



1. For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project.

## SELECT PNUMBER, PNAME, COUNT(\*) AS NO\_OF\_EMPLOYEE FROM PROJECT, WORKS\_ON WHERE PNUMBER = PNO AND ESSN IN (SELECT E.SSN FROM EMPLOYEE AS E WHERE E.DNO = 5) GROUP BY PNUMBER, PNAME;

1. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than $40,000.

## SELECT DNAME, COUNT(\*) AS NO\_OF\_EMPLOYEE FROM DEPARTMENT, EMPLOYEE WHERE DNUMBER = DNO AND SALARY>40000 AND DNO IN (SELECT DNO FROM EMPLOYEE GROUP BY DNO HAVING COUNT(\*)>5) GROUP BY DNAME;

1. Retrieve the names of employees who make at least $10,000 more than the employee who is paid the least in the company.

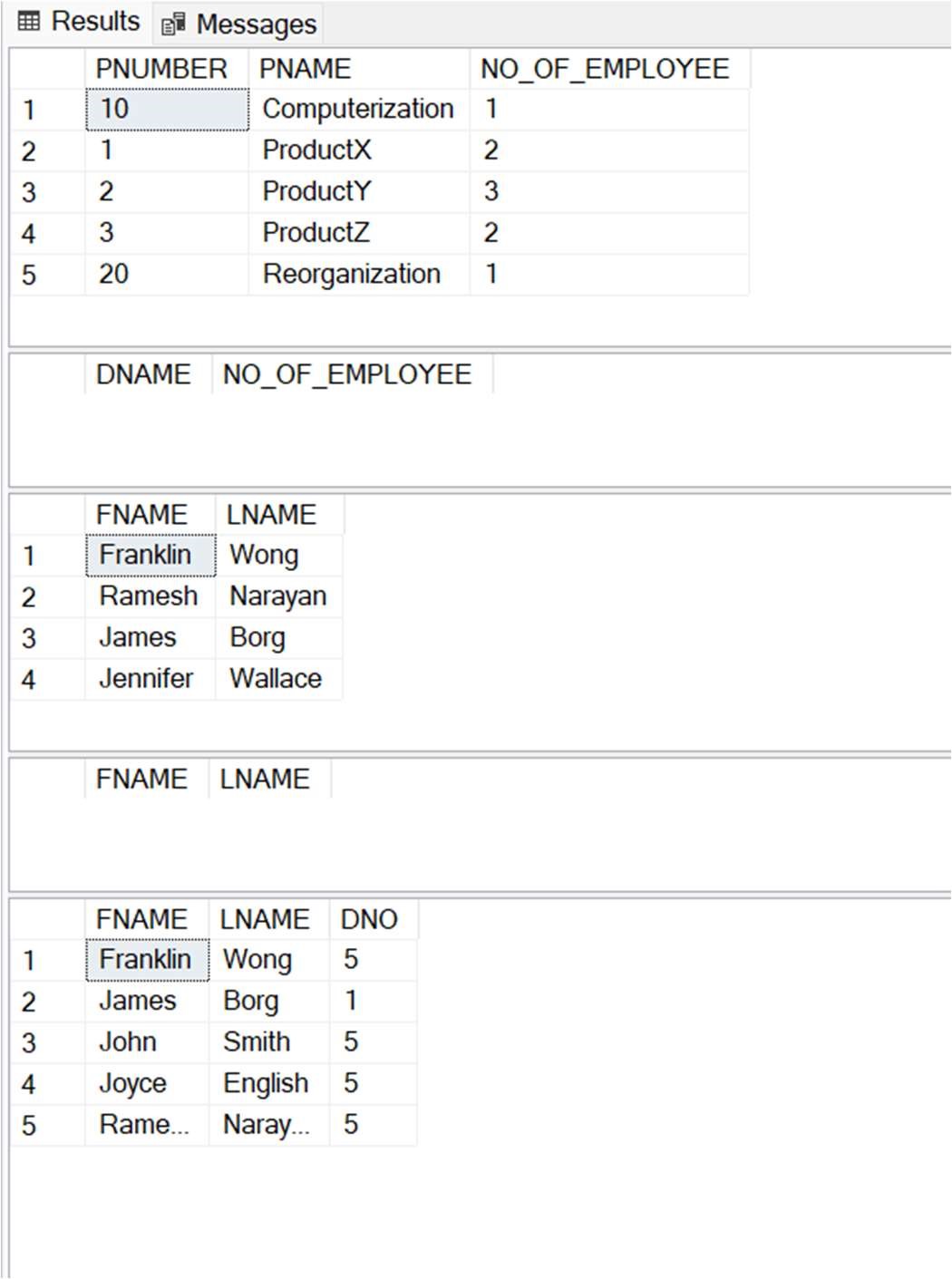
SELECT FNAME, LNAME FROM EMPLOYEE WHERE SALARY >= (SELECT MIN(SALARY) FROM EMPLOYEE)+10000;

1. Display Employee Name and Department Number for the Employee No= 7900.

## SELECT FNAME, LNAME FROM EMPLOYEE WHERE SSN='%7900%';

11. Display Employee Name and Department No. of all the employees in Dept 5 and Dept 1 in the alphabetical order by name.

## SELECT FNAME, LNAME, DNO FROM EMPLOYEE WHERE DNO=5 OR DNO=1 ORDER BY FNAME;



# Practical 5 – Aggregation function-based queries

1. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary.

## SELECT SUM(SALARY) AS SUM\_SALARY, MIN(SALARY) AS MIN\_SALARY, MAX(SALARY) AS MAX\_SALARY, AVG(SALARY) AS AVG\_SALARY FROM EMPLOYEE;

1. Find the sum of the salaries of all employees of the ‘Research’ department, as well as the maximum salary, the minimum salary, and the average salary in this department.

SELECT SUM(SALARY) AS SUM\_SALARY, MIN(SALARY) AS MIN\_SALARY, MAX(SALARY) AS MAX\_SALARY, AVG(SALARY) AS AVG\_SALARY FROM EMPLOYEE WHERE DNO = (SELECT DNUMBER FROM DEPARTMENT WHERE DNAME = 'Research');

1. Retrieve the total number of employees in the company

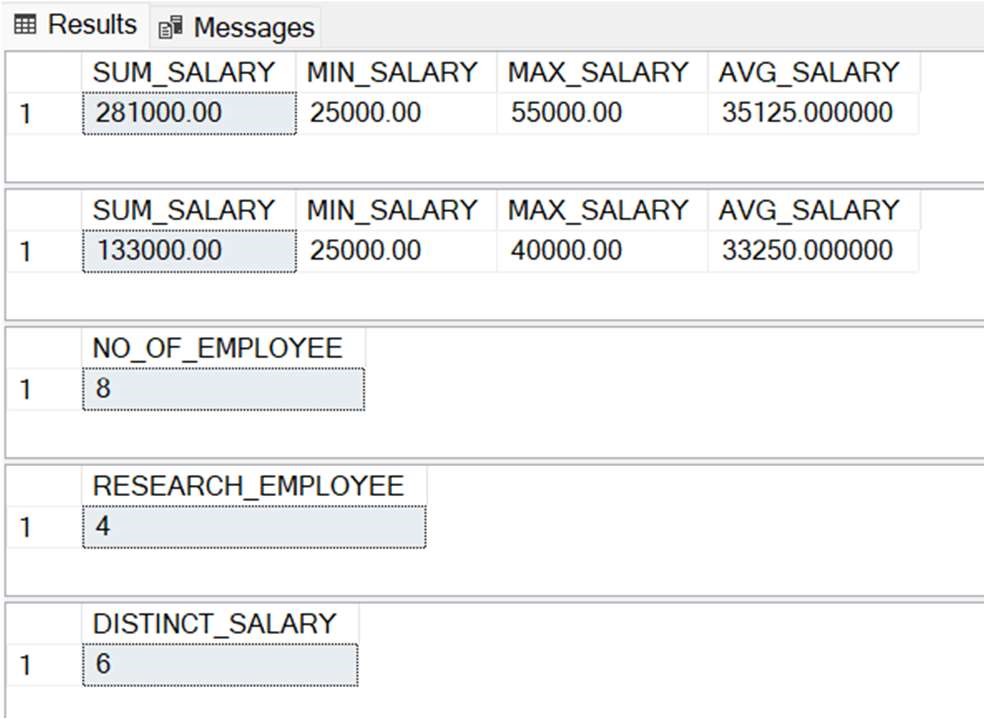
SELECT COUNT(\*) AS NO\_OF\_EMPLOYEE FROM EMPLOYEE;

1. Retrieve the number of employees in the ‘Research’ department.

SELECT COUNT(\*) AS RESEARCH\_EMPLOYEE FROM EMPLOYEE WHERE DNO = (SELECT DNUMBER FROM DEPARTMENT WHERE DNAME = 'Research');

1. Count the number of distinct salary values in the database

## SELECT COUNT(DISTINCT SALARY) AS DISTINCT\_SALARY FROM EMPLOYEE;



1. Retrieve the names of all employees who have two or more dependents

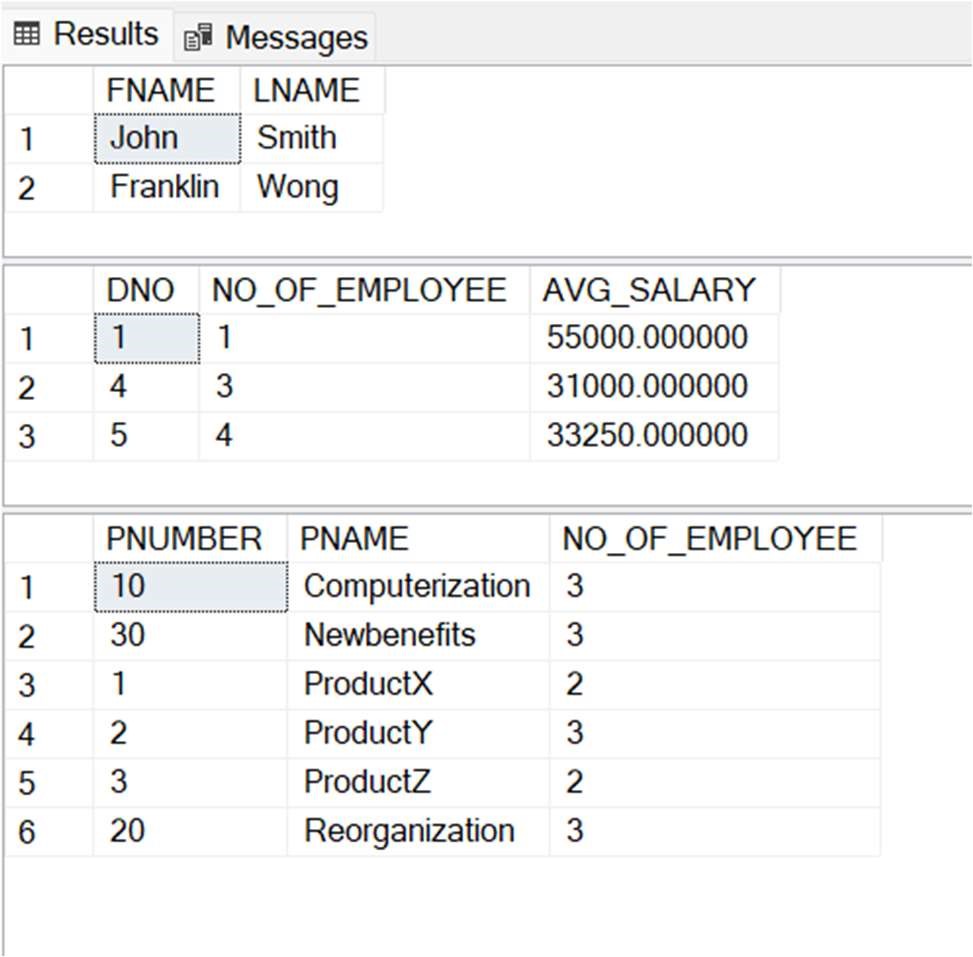
SELECT FNAME, LNAME FROM EMPLOYEE WHERE(SELECT COUNT(\*) FROM DEPENDENT WHERE SSN=ESSN)>=2;

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

## SELECT DNO, COUNT(\*) AS NO\_OF\_EMPLOYEE, AVG(SALARY) AS AVG\_SALARY FROM EMPLOYEE GROUP BY DNO;

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

## SELECT PNUMBER, PNAME, COUNT(\*) AS NO\_OF\_EMPLOYEE FROM PROJECT, WORKS\_ON WHERE PNUMBER = PNO GROUP BY PNUMBER, PNAME;



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

SELECT P.PNUMBER, P.PNAME, COUNT(W.PNO) AS NO\_OF\_EMPLOYEE FROM PROJECT AS P, WORKS\_ON AS W WHERE P.PNUMBER = W.PNO GROUP BY P.PNUMBER, P.PNAME, W.PNO HAVING COUNT(W.PNO)>2;

1. List the names of managers who have at least one dependent.

## SELECT FNAME, LNAME FROM EMPLOYEE, DEPARTMENT WHERE SSN = MGR\_SSN AND (SELECT COUNT(\*) FROM DEPENDENT WHERE SSN=ESSN)>=1;

1. Retrieve all distinct salary values in COMPANY.

SELECT DISTINCT SALARY FROM EMPLOYEE;

1. Display Name and Salaries represented by asterisks, where each asterisk (\*) signifies $100.

SELECT FNAME, REPLICATE('\*',SALARY/100) AS STAR\_REPRESENTATION FROM EMPLOYEE;

1. Display the no. of managers in the COMPANY.

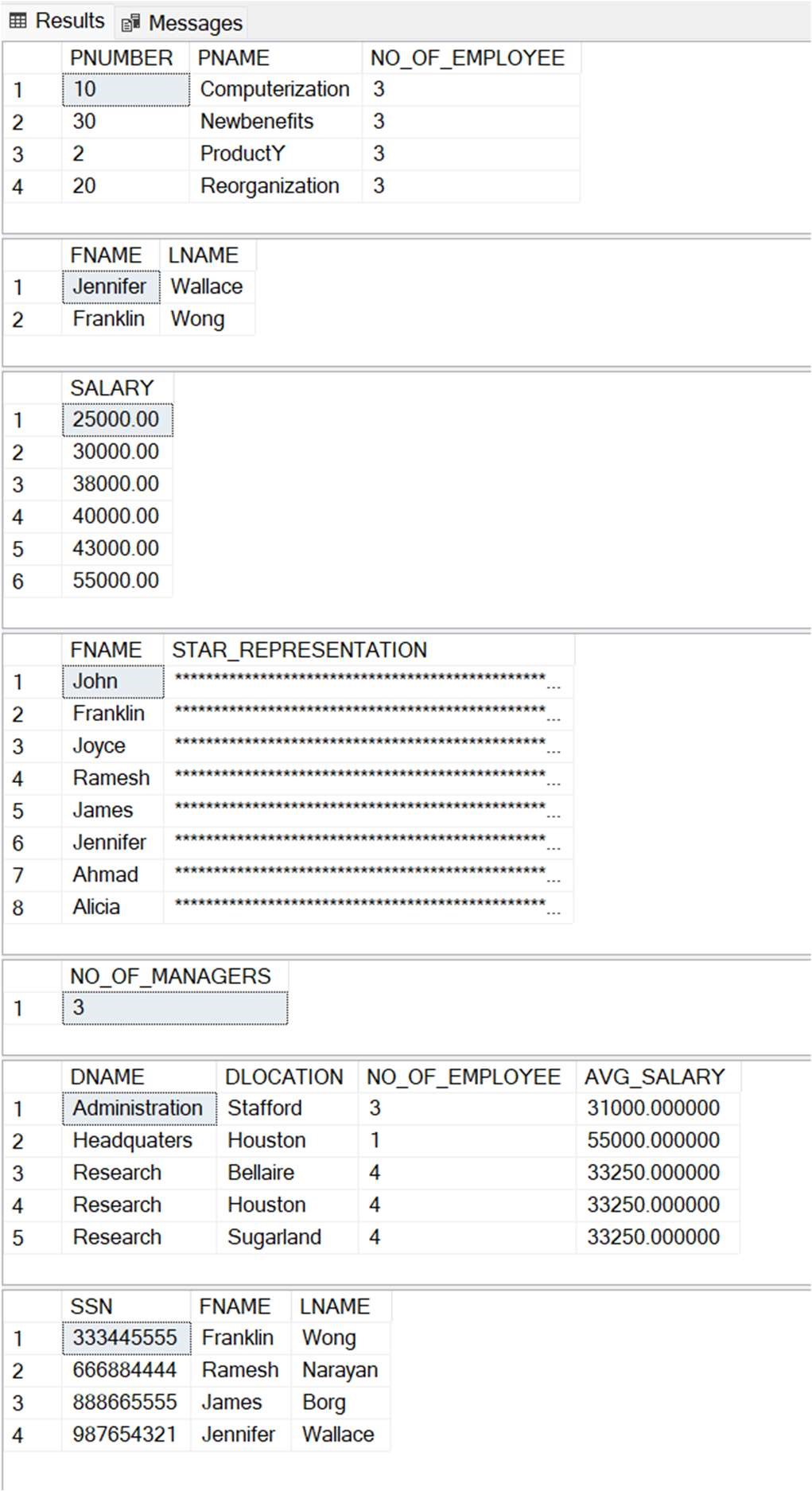
## SELECT COUNT(DISTINCT MGR\_SSN) FROM DEPARTMENT;

1. Display the Department Name, Location Name, No. of Employees and the average salary in that department.

SELECT D.DNAME, LOC.DLOCATION, COUNT(E.DNO) AS NO\_OF\_EMPLOYEE, AVG(E.SALARY) AS AVG\_SALARY FROM DEPARTMENT AS D, EMPLOYEE AS E, DEPT\_LOCATIONS AS LOC WHERE D.DNUMBER = E.DNO AND D.DNUMBER = LOC.DNUMBER GROUP BY D.DNAME, LOC.DLOCATION, E.DNO;

1. Display the Employee No. And Name for all employees who earn more than the average salary.

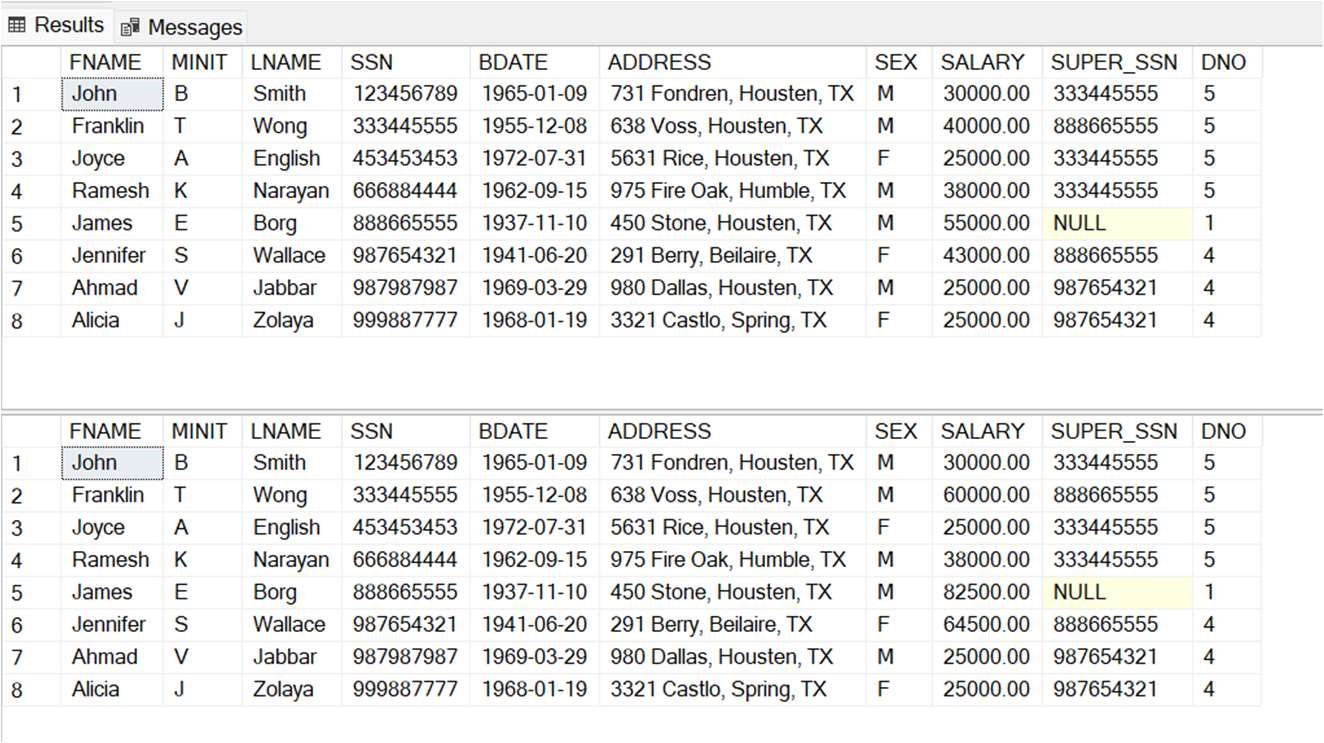
## SELECT SSN, FNAME, LNAME FROM EMPLOYEE WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEE);



# Practical 6– Update/Delete-based queries

1. Increment the salary of managers in the COMPANY by 50%.

## SELECT \* FROM EMPLOYEE; UPDATE EMPLOYEE SET SALARY = SALARY\*1.5 WHERE SSN IN (SELECT MGR\_SSN FROM DEPARTMENT); SELECT \* FROM EMPLOYEE;



2. Decrement the salary of manager of project number 10 by 10%.

## SELECT \* FROM EMPLOYEE; UPDATE EMPLOYEE SET SALARY = SALARY\*0.9 WHERE SSN IN (SELECT MGR\_SSN FROM DEPARTMENT WHERE DNUMBER = (SELECT DNUM FROM PROJECT WHERE PNUMBER = 10));

SELECT

\*

FROM

EMPLOYEE;



1. Update the name of project to ‘XYZ’ lead in department no 5.

## UPDATE PROJECT SET PNAME = 'XYZ' WHERE DNUM = 5;

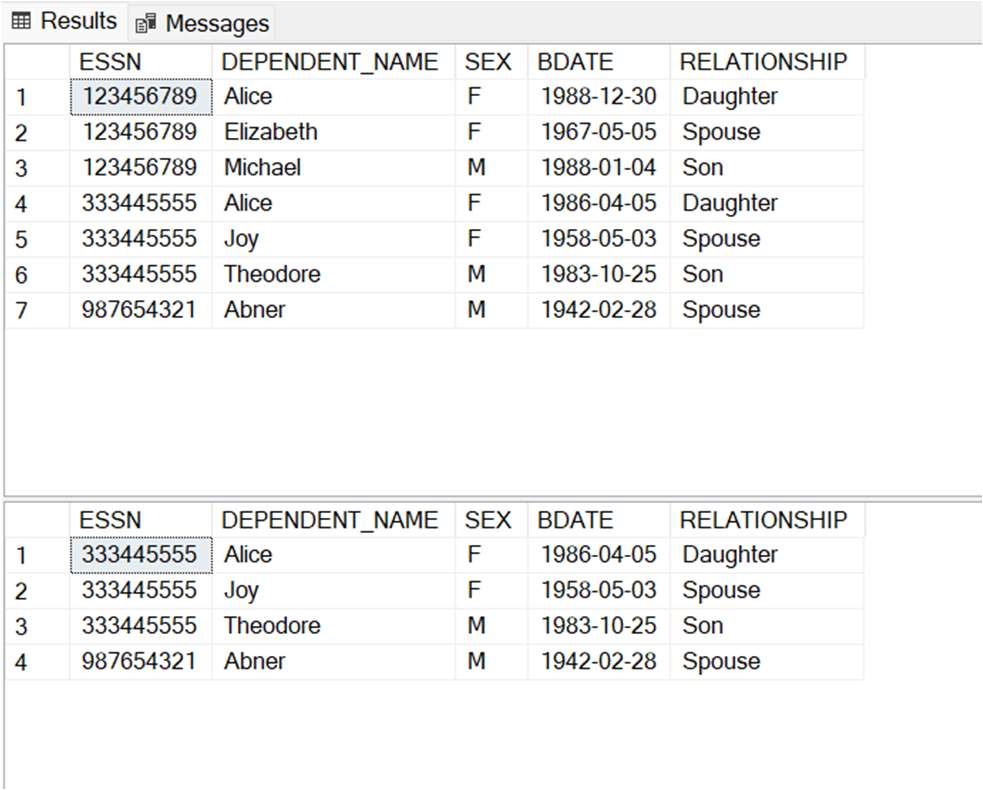


1. Remove the dependents of employee with SNN = ‘123456789’

SELECT \* FROM DEPENDENT;

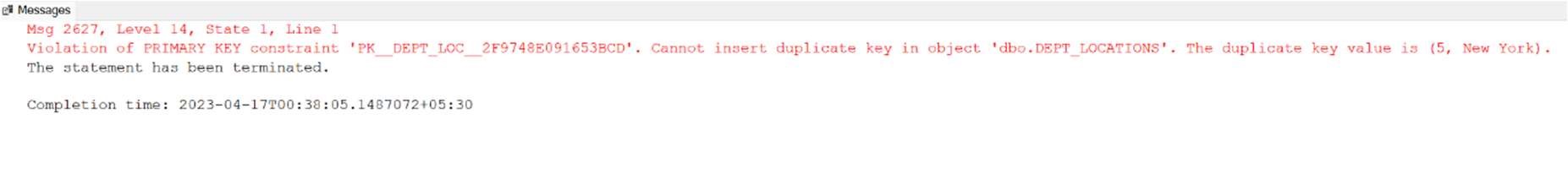
DELETE FROM DEPENDENT WHERE ESSN = 123456789;

SELECT \* FROM DEPENDENT;



1. Change the location of research department to ‘New York’.

## UPDATE DEPT\_LOCATIONS SET DLOCATION = 'New York' WHERE DNUMBER = (SELECT D.DNUMBER FROM DEPARTMENT D WHERE D.DNAME = 'Research');



6. Increment the salary of all the employees working on ‘ProductX’.

## SELECT \* FROM EMPLOYEE; UPDATE EMPLOYEE SET SALARY = 2\*SALARY WHERE SSN IN (SELECT ESSN FROM WORKS\_ON WHERE PNO = (SELECT PNUMBER FROM PROJECT WHERE PNAME = 'ProductX'));

SELECT

\*

FROM

EMPLOYEE;



7. Assign all the employees working under manager 333445555 to manager 987654321.

## SELECT \* FROM EMPLOYEE; UPDATE EMPLOYEE SET SUPER\_SSN = 987654321 WHERE SUPER\_SSN = 333445555; SELECT \* FROM EMPLOYEE;



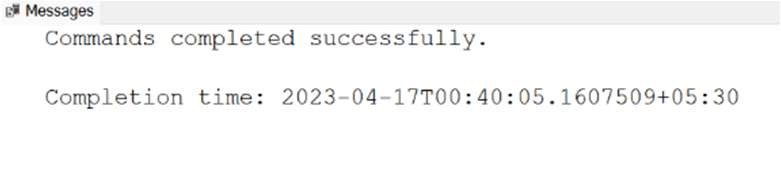
1. Delete all the employees having age greater than 65.

DELETE FROM EMPLOYEE WHERE (SELECT DATEDIFF(YEAR,BDATE,GETDATE()) + 0)>65;



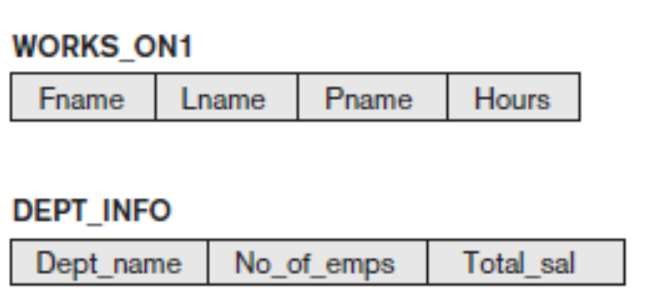
1. Make ‘333445555’ as the default manager for any department.

## ALTER TABLE DEPARTMENT ADD DEFAULT 333445555 FOR MGR\_SSN;



# Practical 7 – Some more Update/Delete-based, View-based and Alter-based queries

1. Create the virtual tables(views) for the schemas given below.



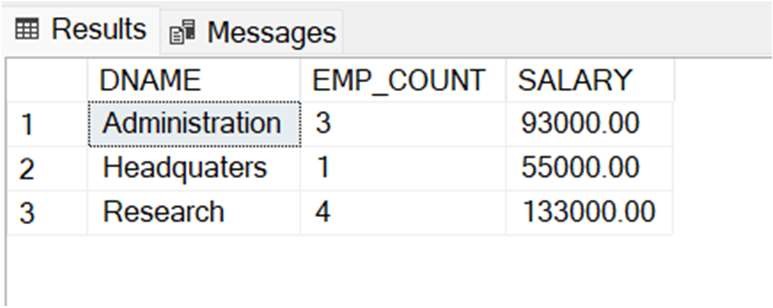
CREATE VIEW WORKS\_ON1 AS

## SELECT E.FNAME AS FNAME, E.LNAME AS LNAME, P.PNAME AS PNAME, W.HOURS AS HOURS FROM EMPLOYEE E, PROJECT P, WORKS\_ON W WHERE E.SSN = W.ESSN AND P.PNUMBER = W.PNO;



CREATE VIEW DEPT\_INFO AS

## SELECT DNAME, COUNT(\*) EMP\_COUNT, SUM(SALARY) AS SALARY FROM DEPARTMENT, EMPLOYEE WHERE DNO = DNUMBER GROUP BY DNAME;



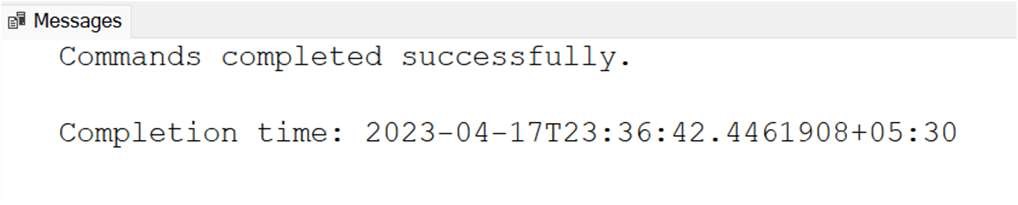
2. Consider the WORKS\_ON1 view, and issue the command to update the PNAME attribute of ‘John Smith’ from ‘ProductX’ to ‘ProductY’.

## UPDATE WORKS\_ON1 SET PNAME = 'ProductY' WHERE FNAME = 'John' AND PNAME = 'ProductX';



3. Get rid of the view WORKS\_ON1.

## DROP VIEW WORKS\_ON1;



1. Create a view that has the department name, manager name, and manager salary for every department.

CREATE VIEW DEPT AS

## SELECT D.DNAME AS DNAME, E.FNAME AS FNAME, E.LNAME AS LNAME, E.SALARY AS SALARY FROM EMPLOYEE E, DEPARTMENT D

WHERE

E

.

SSN

=

D

.

MGR\_SSN

;

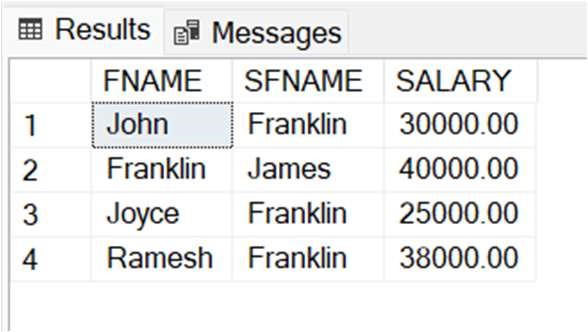


1. Create a view that has the employee name, supervisor name, and employee salary for each employee who works in the ‘Research’ department.

CREATE VIEW RESEARCH AS

## SELECT E1.FNAME AS FNAME, E2.FNAME AS SFNAME, E1.SALARY AS SALARY FROM EMPLOYEE E1,EMPLOYEE E2, DEPARTMENT D

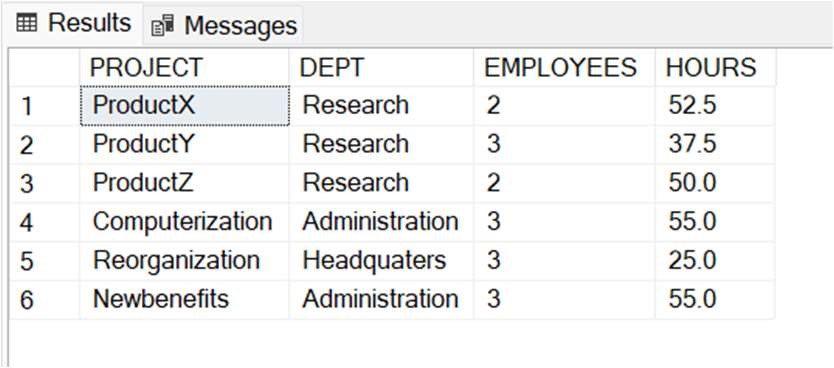
## WHERE E1.SUPER\_SSN = E2.SSN AND E1.DNO = (SELECT D.DNUMBER WHERE DNAME = 'Research');



1. Create a view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.

CREATE VIEW PROJ AS

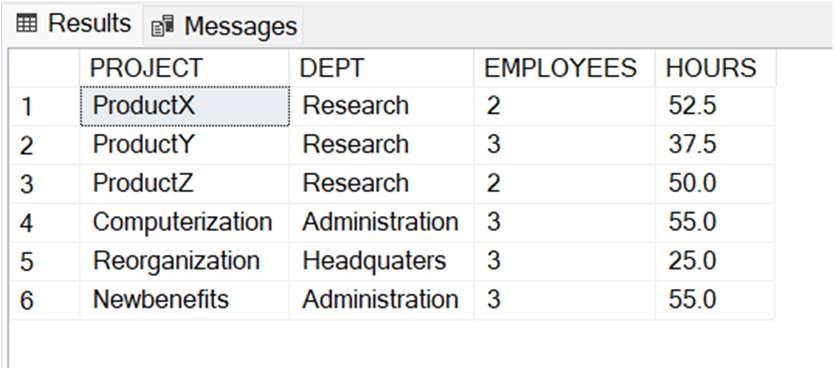
## SELECT PNAME AS PROJECT, DNAME AS DEPT, (SELECT COUNT(\*) FROM WORKS\_ON W WHERE W.PNO = P.PNUMBER) AS EMPLOYEES, (SELECT SUM(W.Hours) FROM WORKS\_ON W WHERE W.PNO = P.PNUMBER GROUP BY PNO) AS HOURS FROM PROJECT P, DEPARTMENT D WHERE P.DNUM = D.DNUMBER;



1. Create a view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

CREATE VIEW NEW\_PROJ AS

## SELECT PNAME AS PROJECT, DNAME AS DEPT, (SELECT COUNT(\*) FROM WORKS\_ON W WHERE W.PNO = P.PNUMBER) AS EMPLOYEES, (SELECT SUM(W.Hours) FROM WORKS\_ON W WHERE W.PNO = P.PNUMBER GROUP BY PNO) AS HOURS FROM PROJECT P, DEPARTMENT D WHERE P.DNUM = D.DNUMBER AND (SELECT COUNT(\*) FROM WORKS\_ON W WHERE W.PNO = P.PNUMBER GROUP BY W.PNO)>1;



1. Create a view for a certain user who is only allowed to see employee information for employees who work for department 5.

CREATE VIEW DEPT5 AS

SELECT

\*

FROM

EMPLOYEE

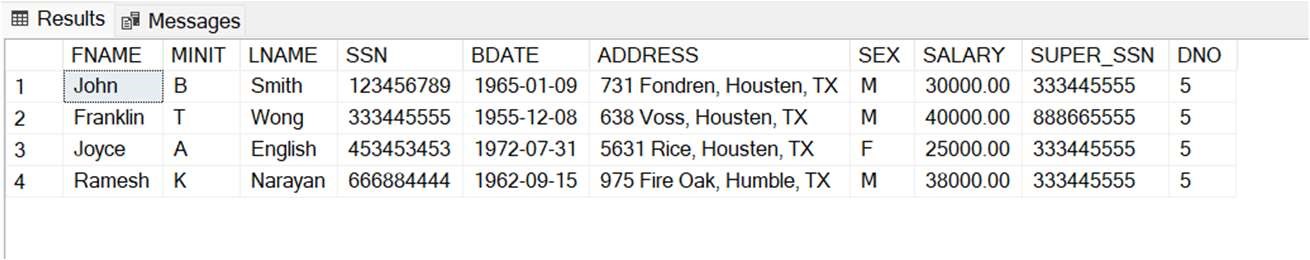
WHERE

DNO

=

5

;



1. Remove the attribute Address from the EMPLOYEE base table.

SELECT \* FROM EMPLOYEE;

## ALTER TABLE EMPLOYEE

DROP COLUMN ADDRESS;

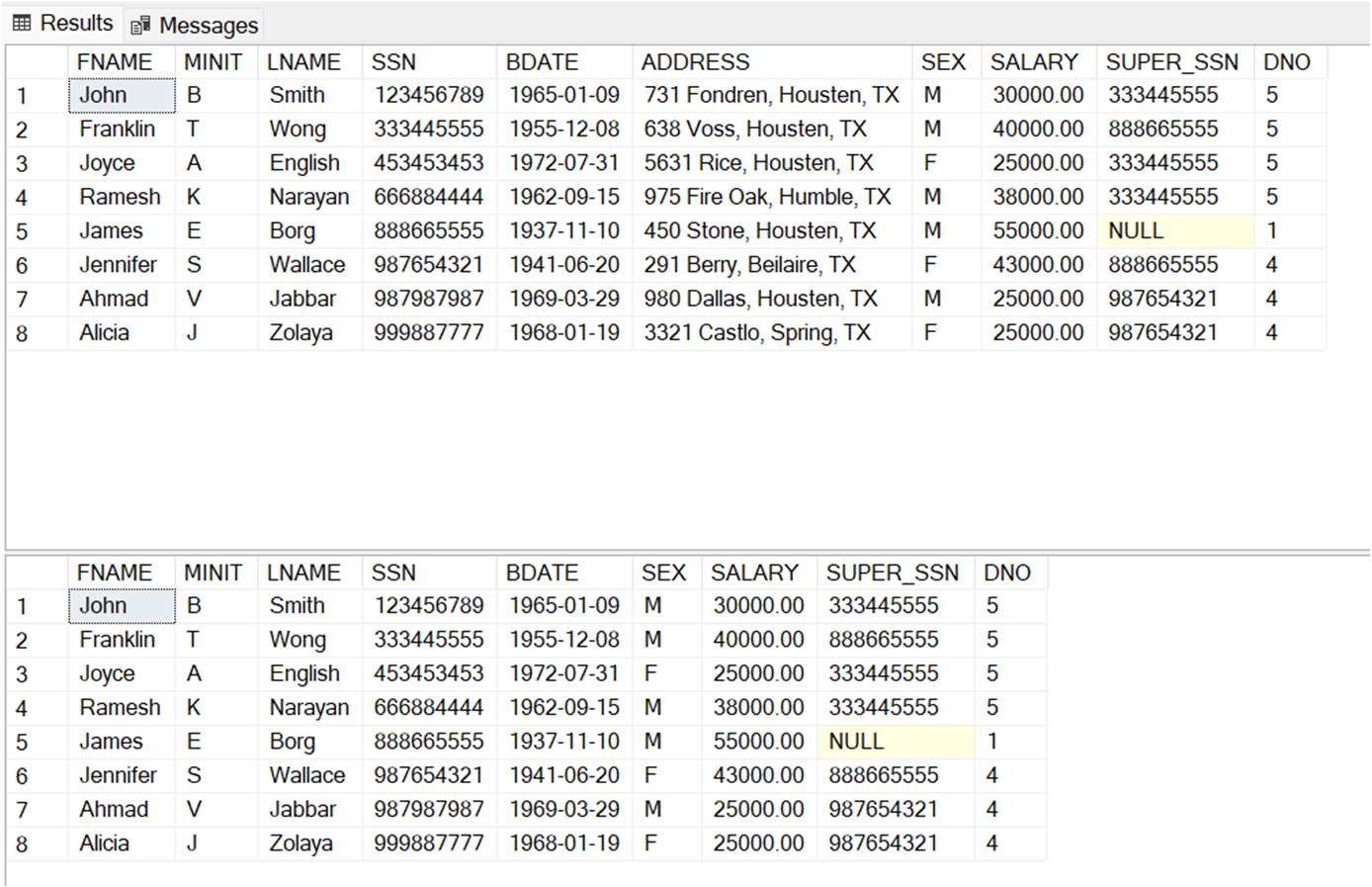
SELECT

\*

FROM

EMPLOYEE

;



# Practical 8 – Some more Nested Queries and complex subqueries

1. Retrieve the names of employees who have no dependents.

SELECT \* FROM EMPLOYEE WHERE SSN NOT IN (SELECT ESSN FROM DEPENDENT);



1. Retrieve all supervisees of a supervisory employee e at all levels—that is, all employees e′ directly supervised by e, all employees e′’ directly supervised by each employee e′, all employees e″′ directly supervised by each employee e″, and so on.

WITH RECURSIVE SUP\_EMP(SUPER\_SSN, SSN) AS( SELECT SUPER\_SSN, SSN FROM EMPLOYEE

UNION

## SELECT E.SSN, S.SUPER\_SSN FROM EMPLOYEE AS E , SUP\_EMP AS S WHERE E.SUPER\_SSN = S.SSN)

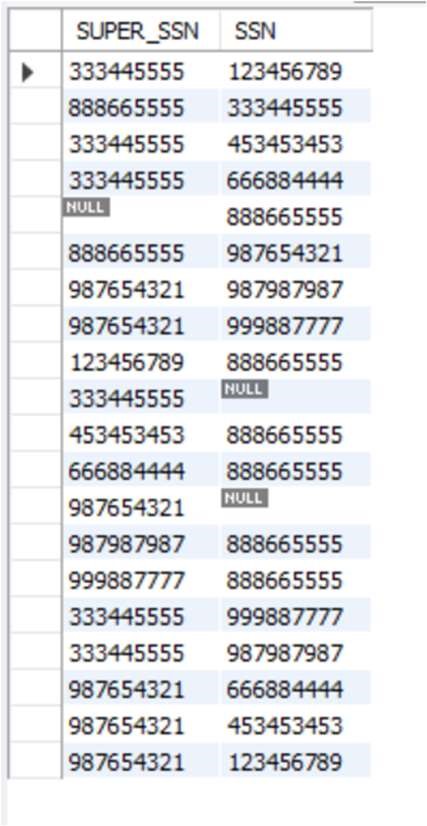
SELECT

\*

FROM

SUP\_EMP

;



1. For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

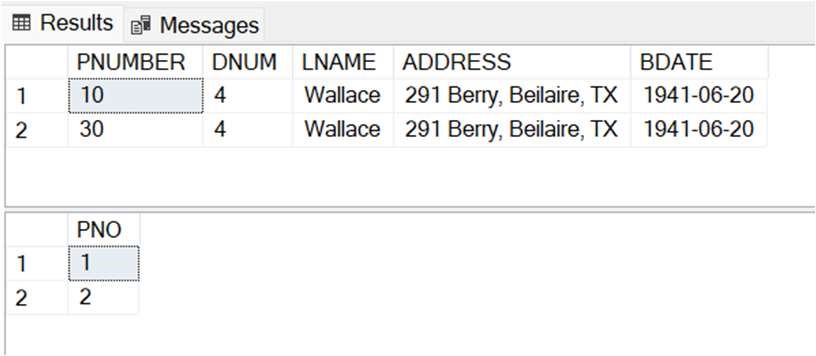
## SELECT P.PNUMBER, P.DNUM, E.LNAME, E.ADDRESS, E.BDATE FROM PROJECT AS P, DEPARTMENT AS D, EMPLOYEE AS E WHERE P.PLOCATION = 'Stafford' AND P.DNUM = D.DNUMBER AND D.MGR\_SSN = E.SSN;

1. Make a list of all project numbers for projects that involve an employee whose last name is ‘Smith’, either as a worker or as a manager of the department that controls the project.

(SELECT W.PNO FROM WORKS\_ON AS W, EMPLOYEE AS E WHERE E.LNAME = 'Smith' AND E.SSN = W.ESSN)

UNION

## (SELECT W.PNO FROM WORKS\_ON AS W, PROJECT AS P, EMPLOYEE AS E, DEPARTMENT AS D WHERE E.LNAME = 'Smith' AND W.PNO = P.PNUMBER AND P.DNUM = D.DNUMBER AND D.MGR\_SSN = E.SSN);



1. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.

SELECT DISTINCT(E.FNAME) FROM EMPLOYEE AS E, WORKS\_ON AS W, PROJECT AS P WHERE P.PNAME = 'ProductX' AND P.PNUMBER = W.PNO AND W.HOURS > 10 AND W.ESSN = E.SSN AND E.DNO = 5;

1. List the names of all employees who have a dependent with the same first name as themselves.

## SELECT E.FNAME FROM EMPLOYEE AS E, DEPENDENT AS D WHERE E.FNAME = D.DEPENDENT\_NAME AND E.SSN = D.ESSN;

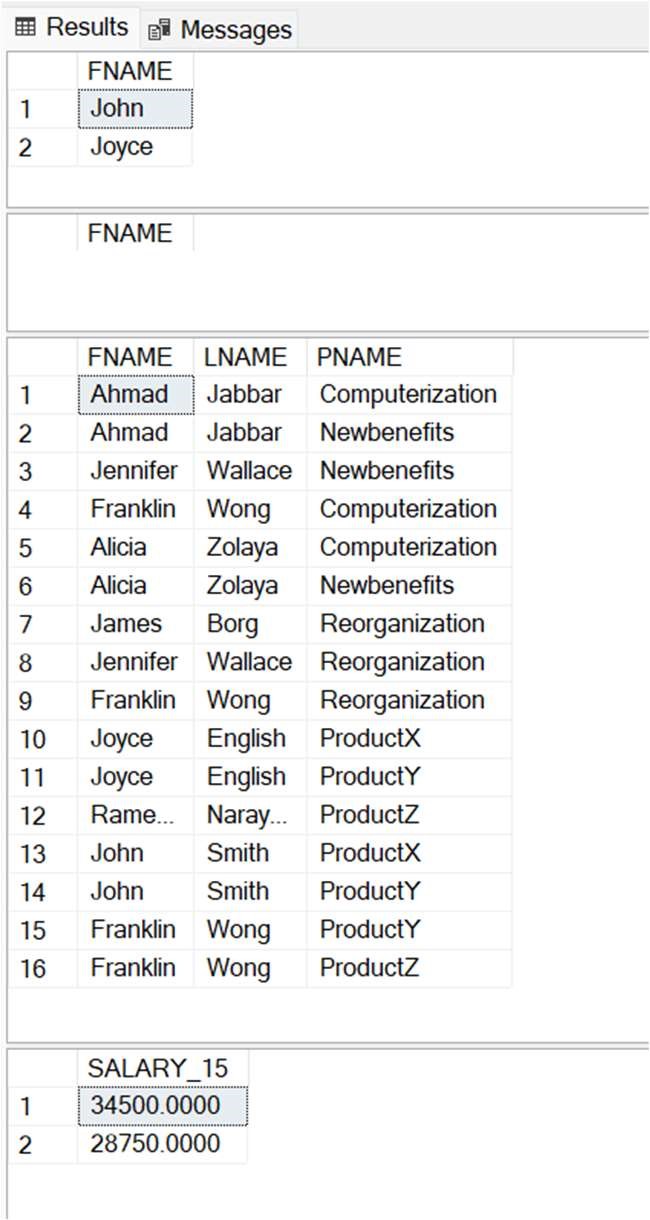
1. Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, then first name.

## SELECT E.FNAME, E.LNAME, P.PNAME FROM EMPLOYEE AS E, DEPARTMENT AS D, WORKS\_ON AS W, PROJECT AS P

## WHERE W.ESSN = E.SSN AND W.PNO = P.PNUMBER AND P.DNUM = D.DNUMBER ORDER BY D.DNAME, E.LNAME, E.FNAME;

1. Show the resulting salaries if every employee working on the ‘ProductX’ project is given a 15 percent raise.

## SELECT E.SALARY\*1.15 AS SALARY\_15 FROM PROJECT AS P, WORKS\_ON AS W, EMPLOYEE AS E WHERE P.PNAME = 'ProductX' AND P.PNUMBER = W.PNO and W.ESSN = E.SSN;



# Practical 9 – Some more Nested Queries and complex subqueries

1. For each department whose average employee salary is more than $30,000, retrieve the department name and the number of employees working for that department.

## SELECT DNAME, COUNT(\*) Emp\_Count FROM DEPARTMENT, EMPLOYEE WHERE DNO = DNUMBER GROUP BY DNAME HAVING AVG(SALARY)>30000;

1. Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.

SELECT FNAME FROM EMPLOYEE WHERE DNO = (SELECT DNO FROM EMPLOYEE WHERE SALARY = (SELECT Max(SALARY) FROM EMPLOYEE));

1. Retrieve the names of all employees whose supervisor’s supervisor has ‘888665555’ for Ssn.

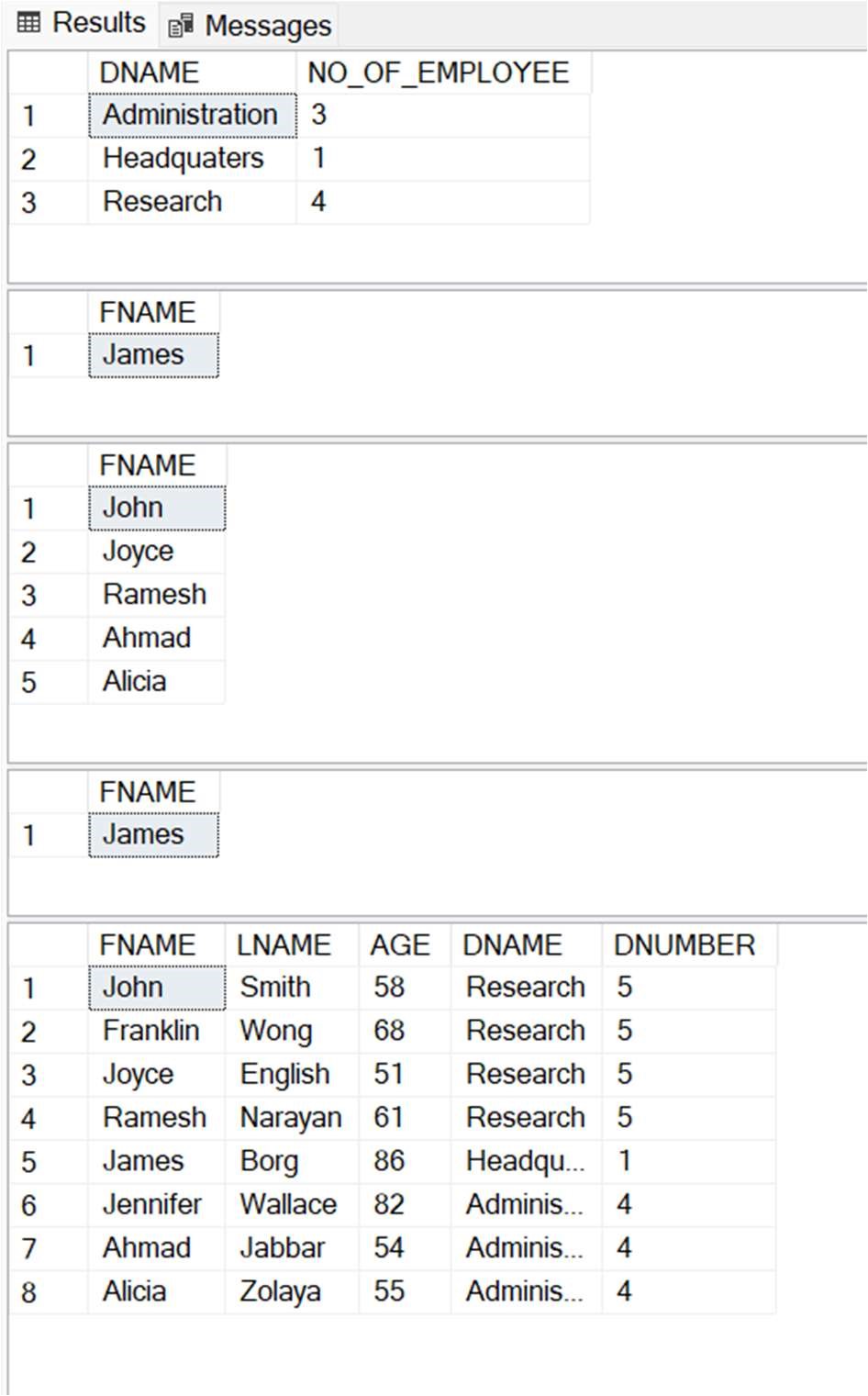
SELECT E1.FNAME FROM EMPLOYEE AS E1, EMPLOYEE AS E2 WHERE E1.SUPER\_SSN = E2.SSN AND E2.SUPER\_SSN = 888665555;

1. Display Name of all employees who don‘t have Manager.

SELECT FNAME FROM EMPLOYEE WHERE SUPER\_SSN IS NULL;

1. Display Employee full name, Age, Department Name and Department No for all the employees.

## SELECT E.FNAME, E.LNAME, (SELECT DATEDIFF(YEAR,E.BDATE,GETDATE()) + 0) AS AGE, D.DNAME, D.DNUMBER FROM EMPLOYEE AS E, DEPARTMENT AS D WHERE D.DNUMBER = E.DNO;



1. Display Name and Employee no. Along with their Manger‘s Name and the Manager‘s employee no; along with the Employees‘ Name who do not have a Manager.

## SELECT E1.FNAME AS ENAME, E1.SSN AS ESSN, E2.FNAME AS MNAME, E2.SSN AS MSSN FROM EMPLOYEE AS E1 LEFT OUTER JOIN EMPLOYEE AS E2 ON E1.SUPER\_SSN = E2.SSN;

1. Display the following for each employee <FName> earns < Salary> monthly but wants < 3 \* Current Salary >. Practicalel the Column as Dream Salary.

## SELECT CONCAT(FNAME,' earns ',SALARY,' monthly but wants ',3\*SALARY) AS DREAM\_SALARY FROM EMPLOYEE;

