The Ministry of Education
Umm Al-Qura University
College of computer Science

وزارة التربية والتعليم
جامعة أم القرى
كلية الحاسبات

# Company database project

1. Abrar Talal Matar 444015304
2. Fanar Alsulami 444001105
3. Raghad Hamid Allqmani 44411421
4. Sarah Abdullah Al-Salami 444004843
5. Sarah Muhammad Alshamrani 444003567

## Class: 1,3,4

## Dr. Leenah Fahd Alagl

# Contents

## Introduction: -

The goal of the Company Management System project is to streamline the internal business activities of the organization by organizing data on departments, employees, projects, and other subjects. The system will have several core tables, each containing unique data on assigned projects, personnel information, and departmental organization. To link these tables, primary and foreign keys will be employed. By implementing this system, the firm will be able to monitor project progress, effectively manage its workforce, and ensure good communication across departments.

## Description: -

### 1.Departments Table

Columns: Department_ID (PK), Department_Name, Manager_ID (FK to Employees), Location_ID (FK to Locations).

Description: Tracks department information and managers.

### 2.Employees Table

Columns: Employee_ID (PK), First_Name, Last_Name, Department_ID (FK to Departments), Manager_ID (selfreferencing FK), etc.

Description: Stores employee details and links them to their departments and managers.

### 3. Dependents Table

Columns: Dependent_ID (PK), Employee_ID (FK to Employees), Dependent_Name, Relationship, Birth_Date.

Description: Tracks employees' dependents (spouses, children, etc.).

### 4. Locations Table

Columns: Location_ID (PK), Location_Name, City, Country.

Description: Stores company location details.

### 5. Projects Table

Columns: Project_ID (PK), Project_Name, Start_Date, End_Date, Department_ID (FK to Departments).

Description: Tracks projects and the departments managing them.

### 6. Project Assignments Table

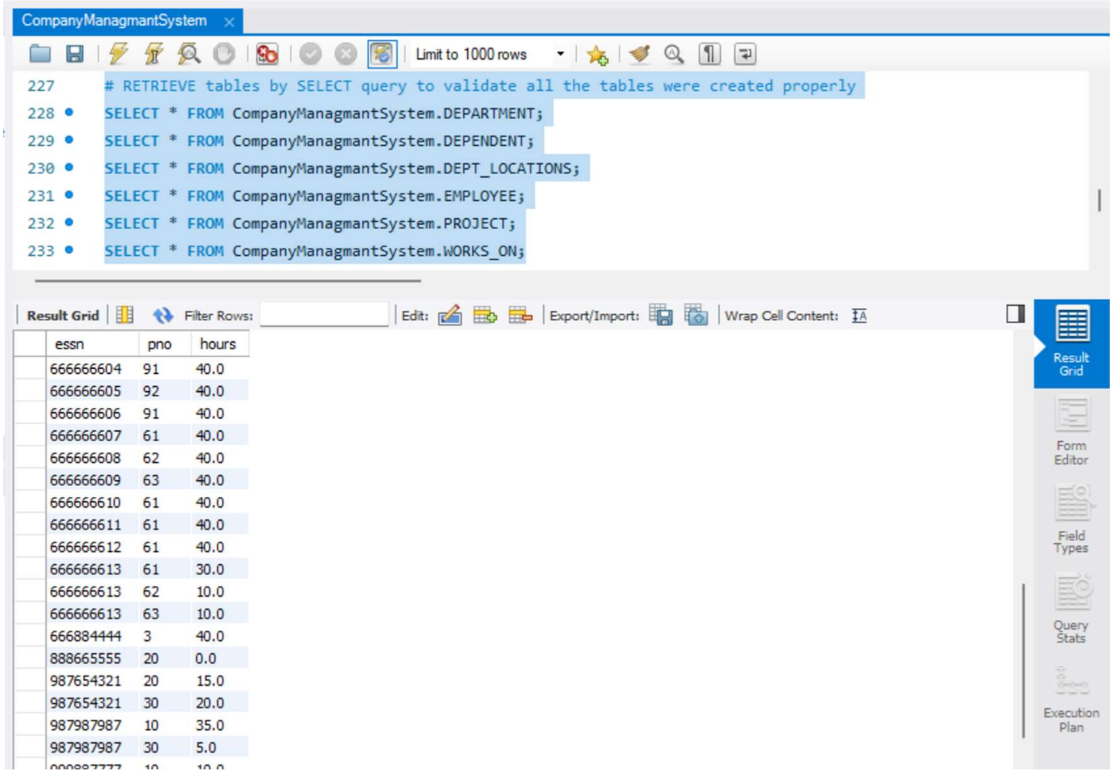
Columns: Assignment_ID (PK), Employee_ID (FK to Employees), Project_ID (FK to Projects), Hours_Worked, Task_Description.

Description: Records employees' tasks and hours on projects.

# Key Relationships: -

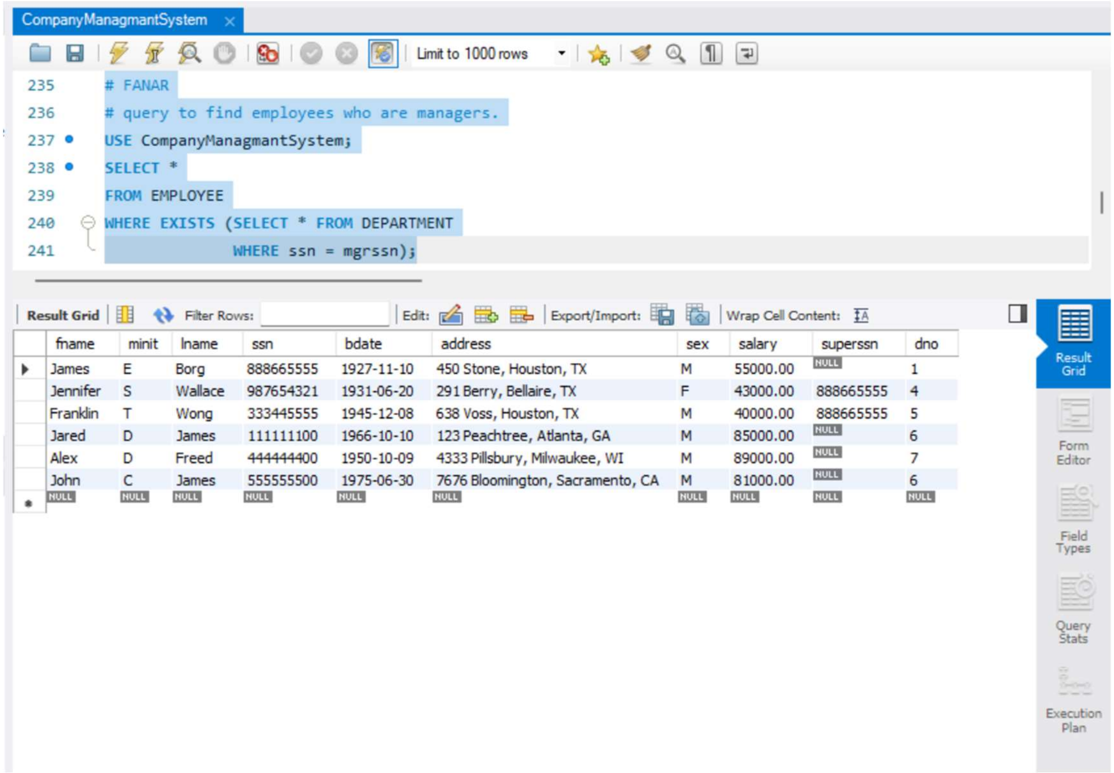
1. Employees & Departments: Linked via Department_ID (each employee belongs to a department).
2. Employees & Dependents: Linked via Employee_ID (tracks employees' family).
3. Departments & Locations: Linked via Location_ID (tracks where departments are located).
4. Projects & Departments: Linked via Department_ID (defines department responsibility for each project).
5. Project Assignments: Links employees to projects through Employee_ID and Project_ID**.**

## Query: -



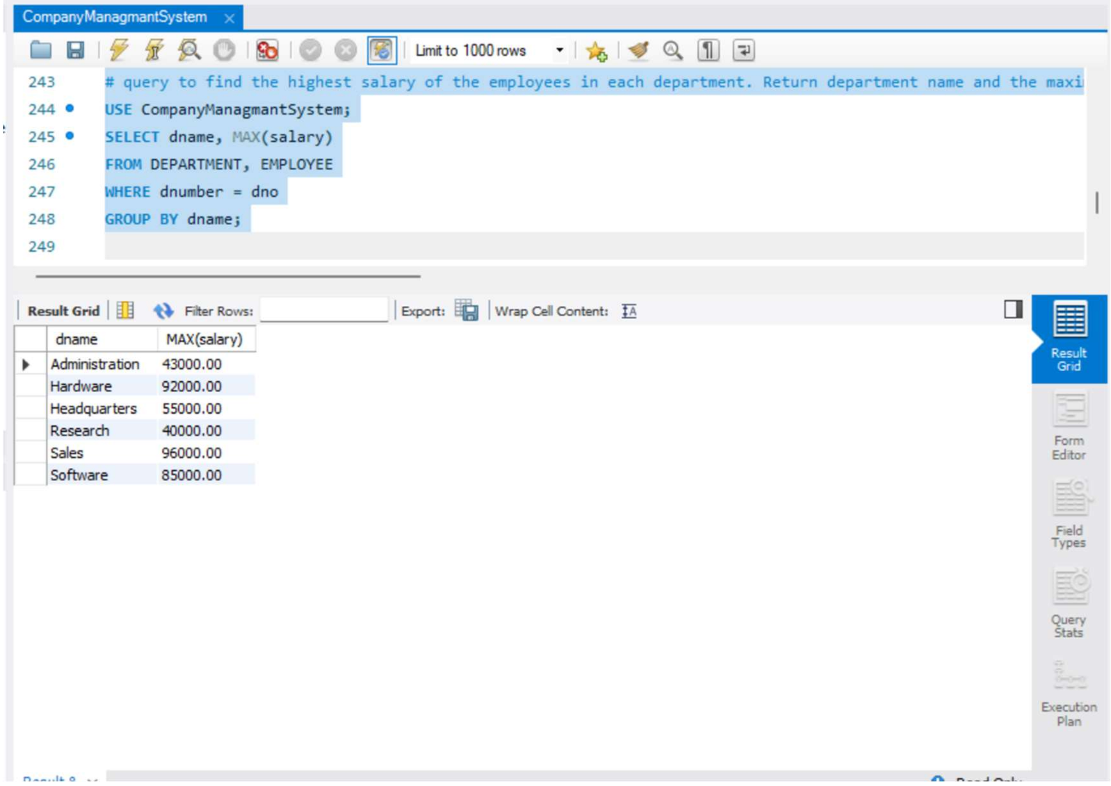
```
# RETRIEVE tables by SELECT query to validate all the tables were created properly
SELECT * FROM CompanyManagmantSystem.DEPARTMENT;
SELECT * FROM CompanyManagmantSystem.DEPENDENT;
SELECT * FROM CompanyManagmantSystem.DEPT_LOCATIONS;
SELECT * FROM CompanyManagmantSystem.EMPLOYEE;
SELECT * FROM CompanyManagmantSystem.PROJECT;
SELECT * FROM CompanyManagmantSystem.WORKS_ON;
```

| essn | pno | hours |
|---|---|---|
| 666666604 | 91 | 40.0 |
| 666666605 | 92 | 40.0 |
| 666666606 | 91 | 40.0 |
| 666666607 | 61 | 40.0 |
| 666666608 | 62 | 40.0 |
| 666666609 | 63 | 40.0 |
| 666666610 | 61 | 40.0 |
| 666666611 | 61 | 40.0 |
| 666666612 | 61 | 40.0 |
| 666666613 | 61 | 30.0 |
| 666666613 | 62 | 10.0 |
| 666666613 | 63 | 10.0 |
| 666884444 | 3 | 40.0 |
| 888665555 | 20 | 0.0 |
| 987654321 | 20 | 15.0 |
| 987654321 | 30 | 20.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 999887777 | 10 | 10.0 |



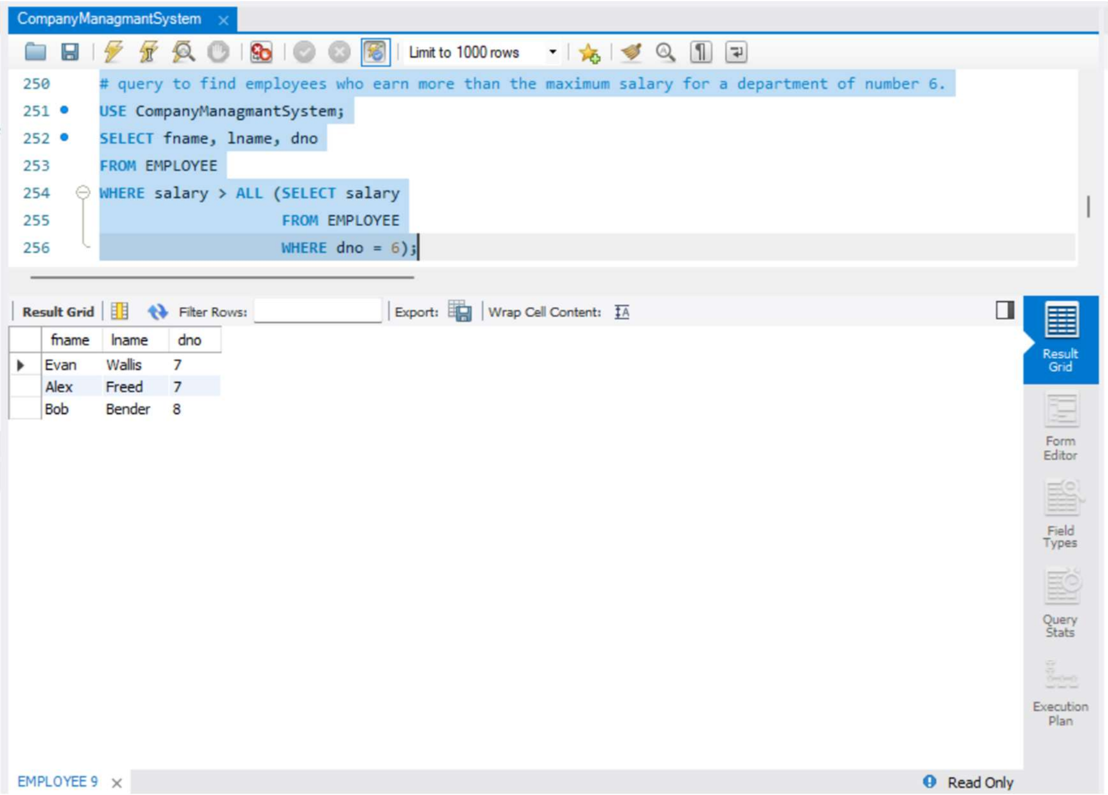
```sql
# FANAR
# query to find employees who are managers.
USE CompanyManagmantSystem;
SELECT *
FROM EMPLOYEE
WHERE EXISTS (SELECT * FROM DEPARTMENT
              WHERE ssn = mgrssn);
```

| fname | minit | lname | ssn | bdate | address | sex | salary | superssn | dno |
|---|---|---|---|---|---|---|---|---|---|
| James | E | Borg | 888665555 | 1927-11-10 | 450 Stone, Houston, TX | M | 55000.00 | NULL | 1 |
| Jennifer | S | Wallace | 987654321 | 1931-06-20 | 291 Berry, Bellaire, TX | F | 43000.00 | 888665555 | 4 |
| Franklin | T | Wong | 333445555 | 1945-12-08 | 638 Voss, Houston, TX | M | 40000.00 | 888665555 | 5 |
| Jared | D | James | 111111100 | 1966-10-10 | 123 Peachtree, Atlanta, GA | M | 85000.00 | NULL | 6 |
| Alex | D | Freed | 444444400 | 1950-10-09 | 4333 Pillsbury, Milwaukee, WI | M | 89000.00 | NULL | 7 |
| John | C | James | 555555500 | 1975-06-30 | 7676 Bloomington, Sacramento, CA | M | 81000.00 | NULL | 6 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |



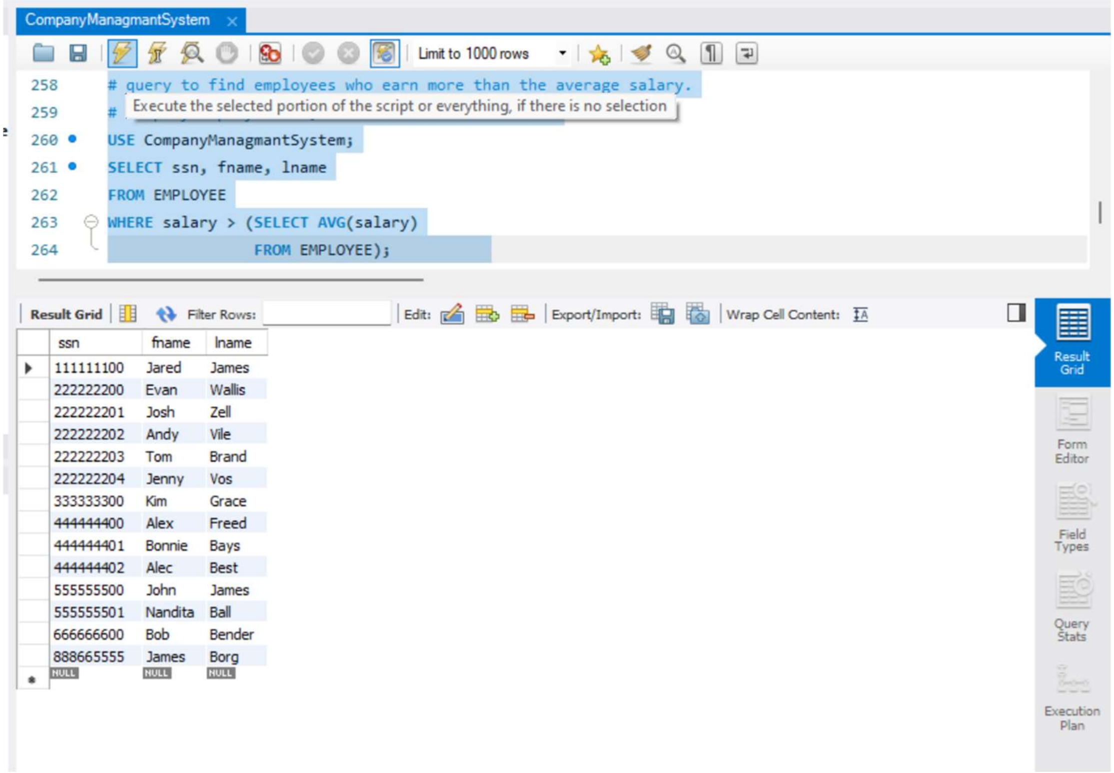
```sql
# query to find the highest salary of the employees in each department. Return department name and the maxi
USE CompanyManagmantSystem;
SELECT dname, MAX(salary)
FROM DEPARTMENT, EMPLOYEE
WHERE dnumber = dno
GROUP BY dname;
```

| dname | MAX(salary) |
|---|---|
| Administration | 43000.00 |
| Hardware | 92000.00 |
| Headquarters | 55000.00 |
| Research | 40000.00 |
| Sales | 96000.00 |
| Software | 85000.00 |



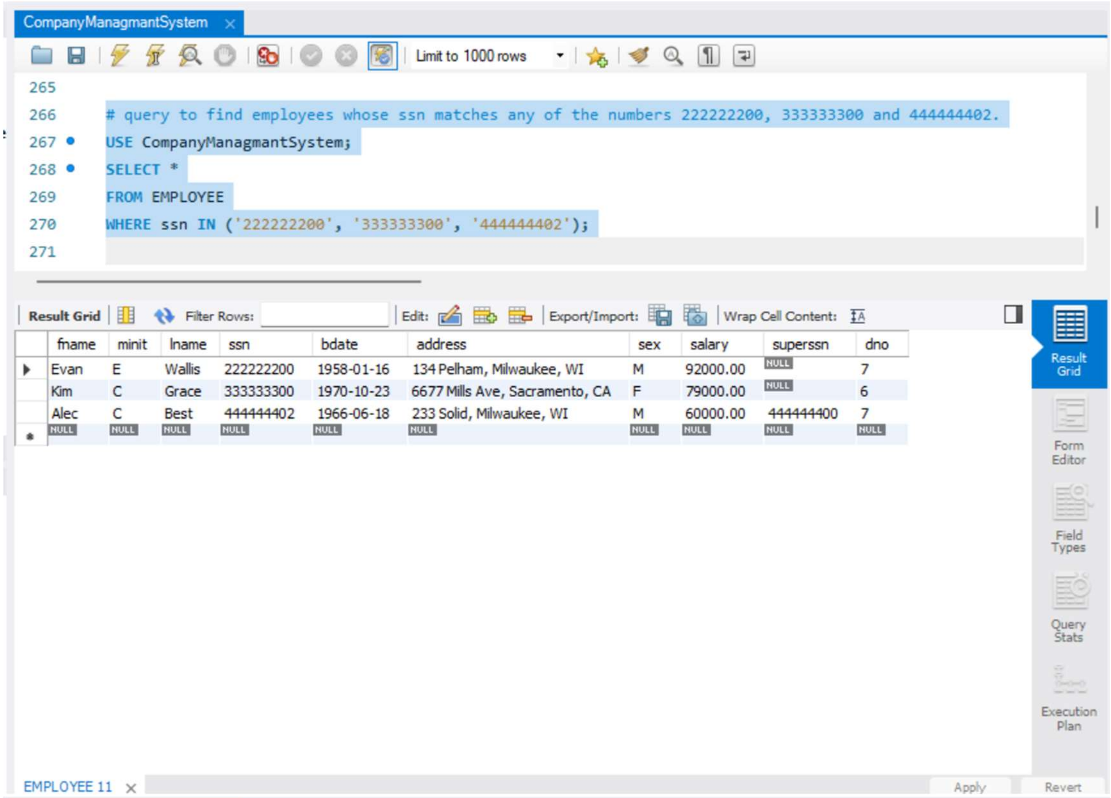
```sql
# query to find employees who earn more than the maximum salary for a department of number 6.
USE CompanyManagmantSystem;
SELECT fname, lname, dno
FROM EMPLOYEE
WHERE salary > ALL (SELECT salary
                    FROM EMPLOYEE
                    WHERE dno = 6);
```

| fname | lname | dno |
|---|---|---|
| Evan | Wallis | 7 |
| Alex | Freed | 7 |
| Bob | Bender | 8 |



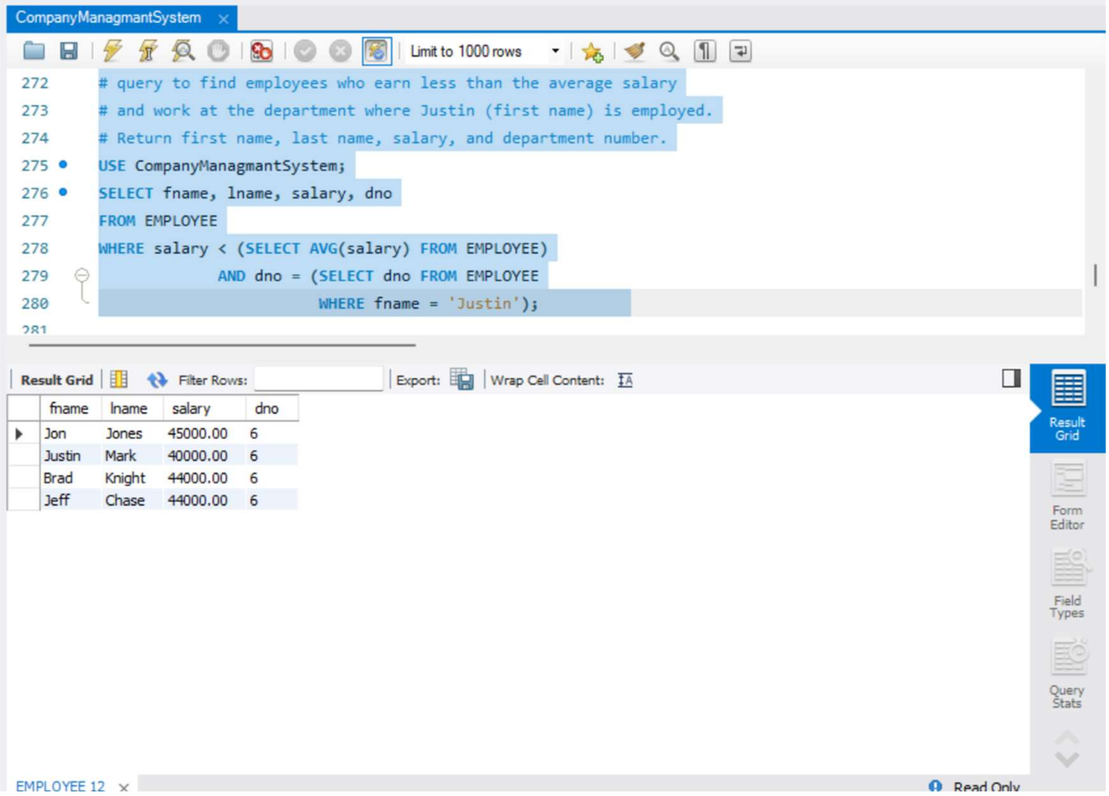
```sql
# query to find employees who earn more than the average salary.
#
USE CompanyManagmantSystem;
SELECT ssn, fname, lname
FROM EMPLOYEE
WHERE salary > (SELECT AVG(salary)
                FROM EMPLOYEE);
```

| ssn | fname | lname |
|---|---|---|
| 111111100 | Jared | James |
| 222222200 | Evan | Wallis |
| 222222201 | Josh | Zell |
| 222222202 | Andy | Vile |
| 222222203 | Tom | Brand |
| 222222204 | Jenny | Vos |
| 333333300 | Kim | Grace |
| 444444400 | Alex | Freed |
| 444444401 | Bonnie | Bays |
| 444444402 | Alec | Best |
| 555555500 | John | James |
| 555555501 | Nandita | Ball |
| 666666600 | Bob | Bender |
| 888665555 | James | Borg |
| NULL | NULL | NULL |



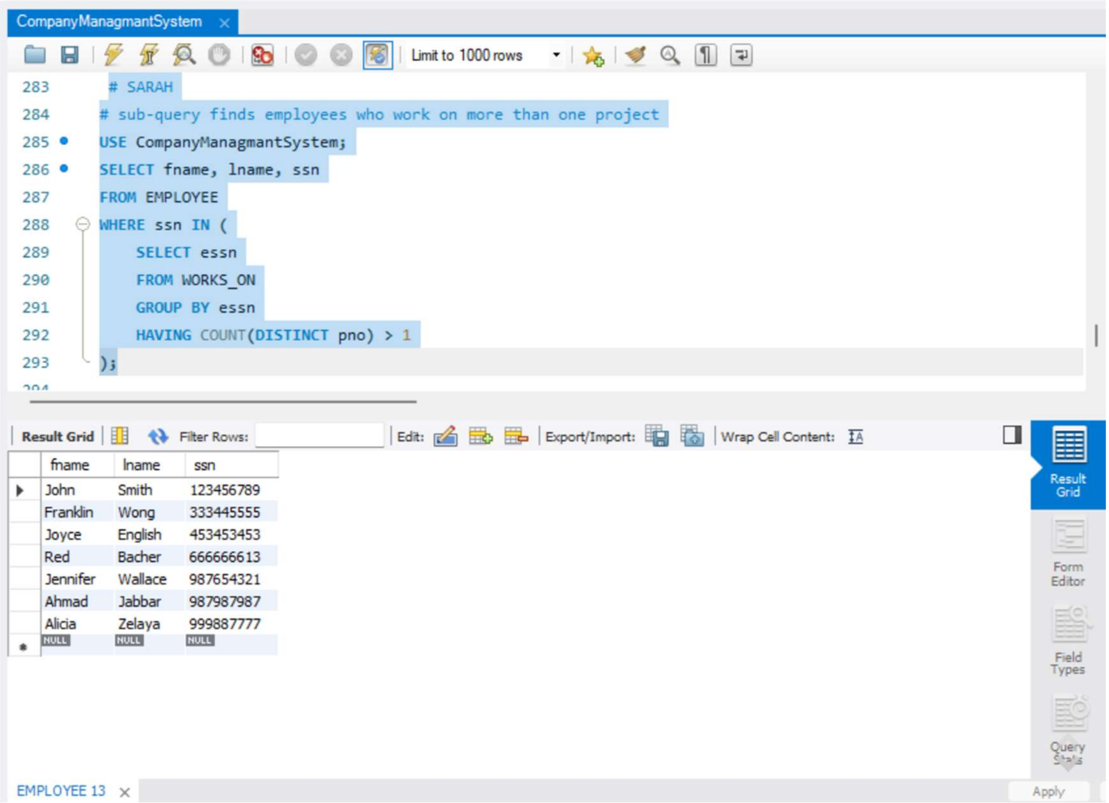
```
# query to find employees whose ssn matches any of the numbers 222222200, 333333300 and 444444402.
USE CompanyManagmantSystem;
SELECT *
FROM EMPLOYEE
WHERE ssn IN ('222222200', '333333300', '444444402');
```

| fname | minit | lname | ssn | bdate | address | sex | salary | superssn | dno |
|---|---|---|---|---|---|---|---|---|---|
| Evan | E | Wallis | 222222200 | 1958-01-16 | 134 Pelham, Milwaukee, WI | M | 92000.00 | NULL | 7 |
| Kim | C | Grace | 333333300 | 1970-10-23 | 6677 Mills Ave, Sacramento, CA | F | 79000.00 | NULL | 6 |
| Alec | C | Best | 444444402 | 1966-06-18 | 233 Solid, Milwaukee, WI | M | 60000.00 | 444444400 | 7 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |



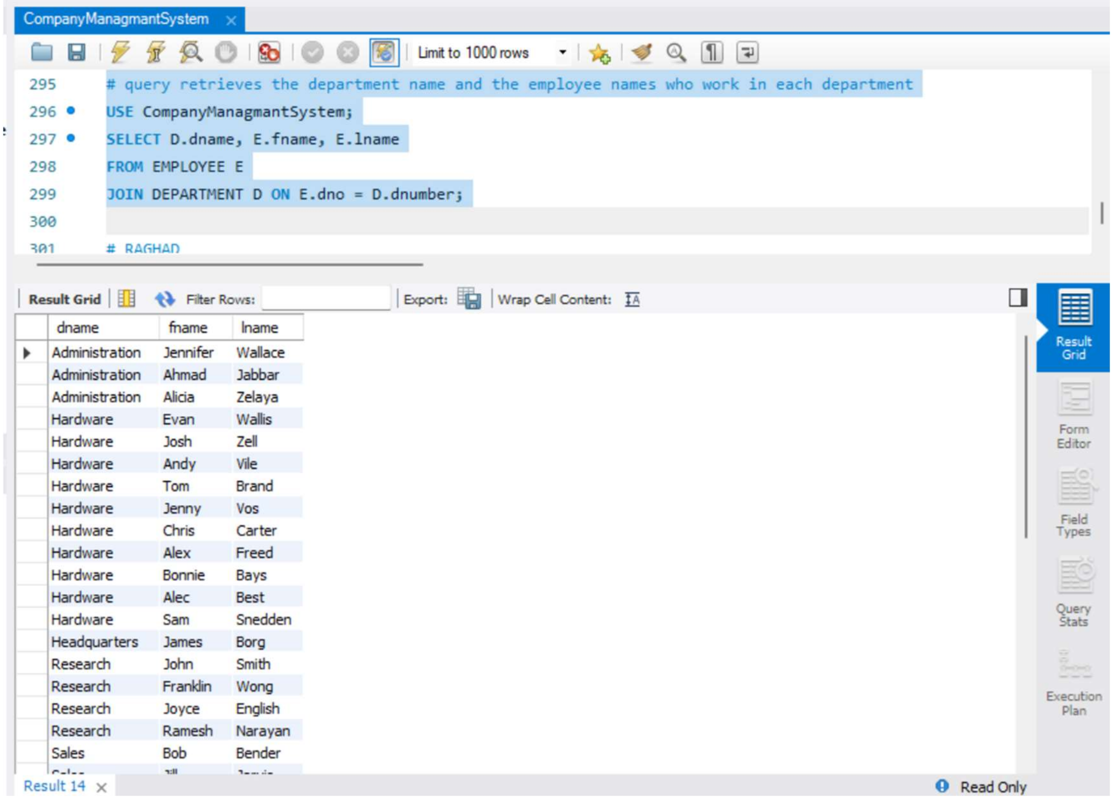
```
# query to find employees who earn less than the average salary
# and work at the department where Justin (first name) is employed.
# Return first name, last name, salary, and department number.
USE CompanyManagmantSystem;
SELECT fname, lname, salary, dno
FROM EMPLOYEE
WHERE salary < (SELECT AVG(salary) FROM EMPLOYEE)
            AND dno = (SELECT dno FROM EMPLOYEE
                       WHERE fname = 'Justin');
```

| fname | lname | salary | dno |
|---|---|---|---|
| Jon | Jones | 45000.00 | 6 |
| Justin | Mark | 40000.00 | 6 |
| Brad | Knight | 44000.00 | 6 |
| Jeff | Chase | 44000.00 | 6 |



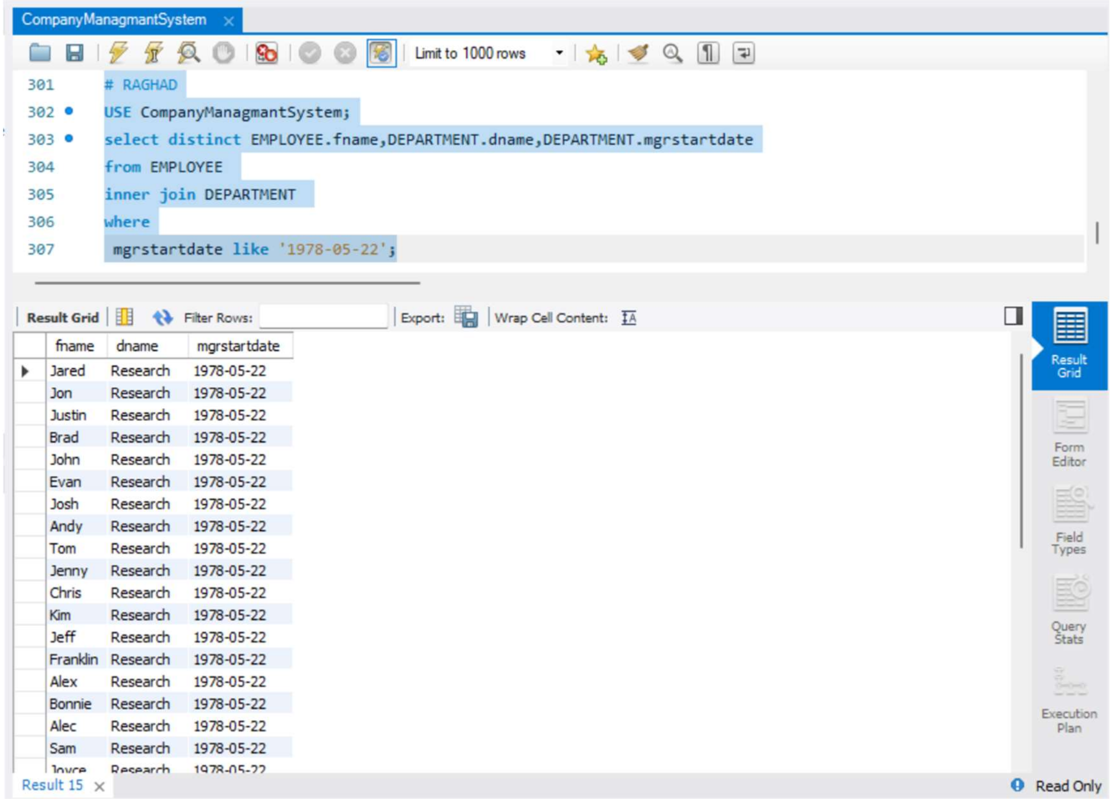
```sql
 # SARAH
# sub-query finds employees who work on more than one project
USE CompanyManagmantSystem;
SELECT fname, lname, ssn
FROM EMPLOYEE
WHERE ssn IN (
    SELECT essn
    FROM WORKS_ON
    GROUP BY essn
    HAVING COUNT(DISTINCT pno) > 1
);
```

| fname | lname | ssn |
|---|---|---|
| John | Smith | 123456789 |
| Franklin | Wong | 333445555 |
| Joyce | English | 453453453 |
| Red | Bacher | 666666613 |
| Jennifer | Wallace | 987654321 |
| Ahmad | Jabbar | 987987987 |
| Alicia | Zelaya | 999887777 |
| NULL | NULL | NULL |



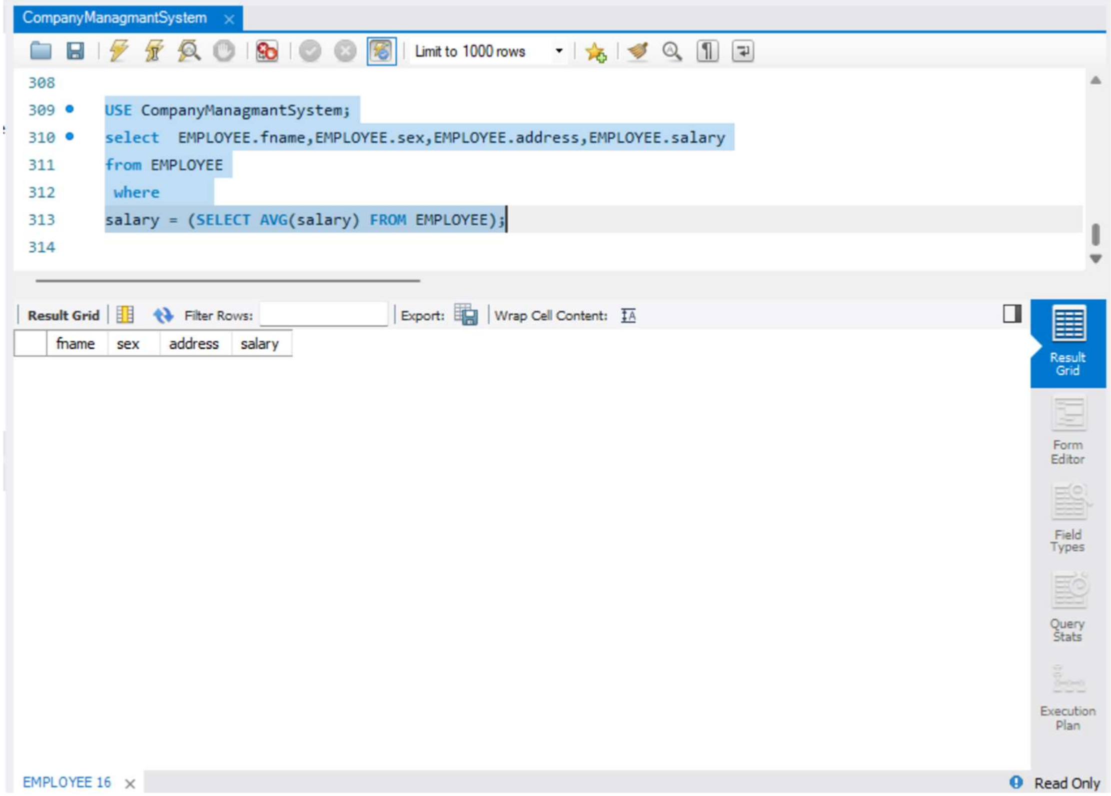
```sql
# query retrieves the department name and the employee names who work in each department
USE CompanyManagmantSystem;
SELECT D.dname, E.fname, E.lname
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.dno = D.dnumber;

# RAGHAD
```

| dname | fname | lname |
|---|---|---|
| Administration | Jennifer | Wallace |
| Administration | Ahmad | Jabbar |
| Administration | Alicia | Zelaya |
| Hardware | Evan | Wallis |
| Hardware | Josh | Zell |
| Hardware | Andy | Vile |
| Hardware | Tom | Brand |
| Hardware | Jenny | Vos |
| Hardware | Chris | Carter |
| Hardware | Alex | Freed |
| Hardware | Bonnie | Bays |
| Hardware | Alec | Best |
| Hardware | Sam | Snedden |
| Headquarters | James | Borg |
| Research | John | Smith |
| Research | Franklin | Wong |
| Research | Joyce | English |
| Research | Ramesh | Narayan |
| Sales | Bob | Bender |



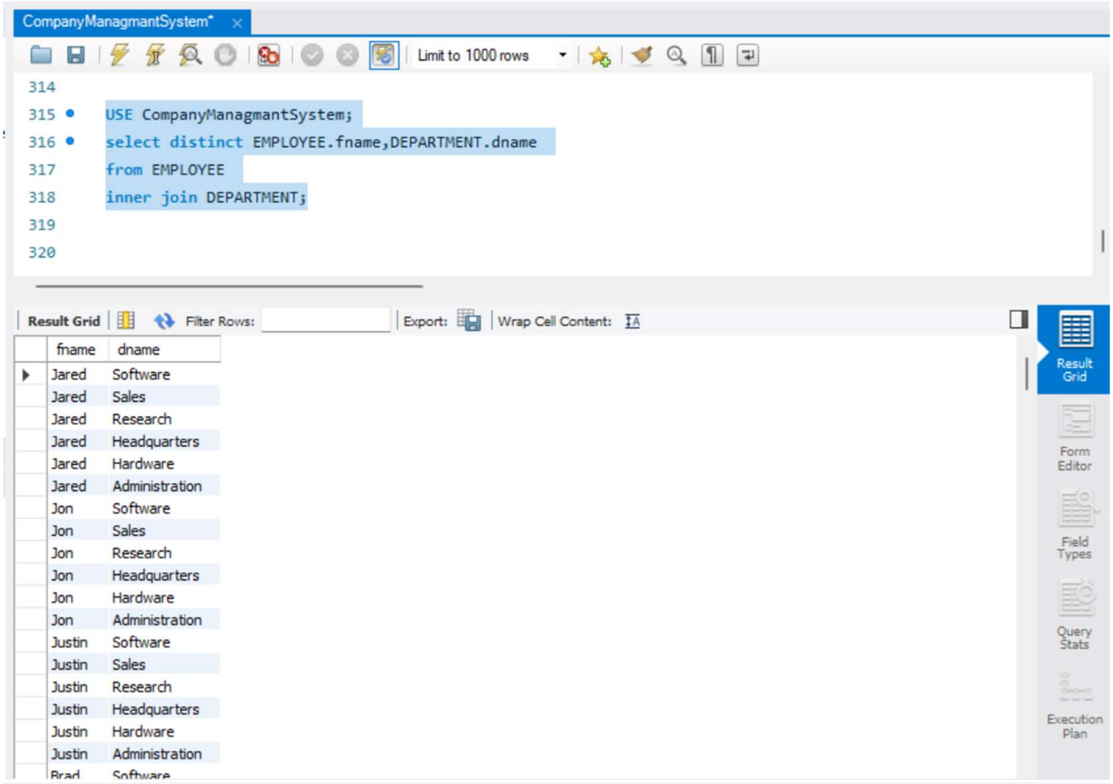
```
# RAGHAD
USE CompanyManagmantSystem;
select distinct EMPLOYEE.fname,DEPARTMENT.dname,DEPARTMENT.mgrstartdate
from EMPLOYEE
inner join DEPARTMENT
where
 mgrstartdate like '1978-05-22';
```

| fname | dname | mgrstartdate |
|---|---|---|
| Jared | Research | 1978-05-22 |
| Jon | Research | 1978-05-22 |
| Justin | Research | 1978-05-22 |
| Brad | Research | 1978-05-22 |
| John | Research | 1978-05-22 |
| Evan | Research | 1978-05-22 |
| Josh | Research | 1978-05-22 |
| Andy | Research | 1978-05-22 |
| Tom | Research | 1978-05-22 |
| Jenny | Research | 1978-05-22 |
| Chris | Research | 1978-05-22 |
| Kim | Research | 1978-05-22 |
| Jeff | Research | 1978-05-22 |
| Franklin | Research | 1978-05-22 |
| Alex | Research | 1978-05-22 |
| Bonnie | Research | 1978-05-22 |
| Alec | Research | 1978-05-22 |
| Sam | Research | 1978-05-22 |
| Joyce | Research | 1978-05-22 |



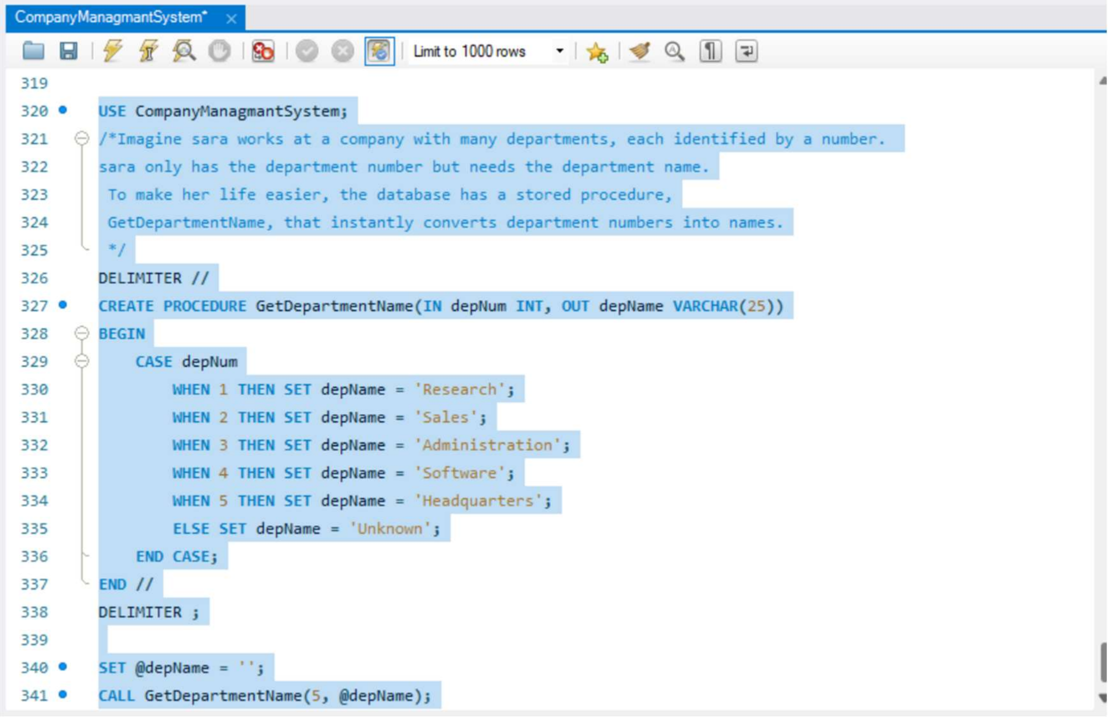
```
USE CompanyManagmantSystem;
select  EMPLOYEE.fname,EMPLOYEE.sex,EMPLOYEE.address,EMPLOYEE.salary
from EMPLOYEE
 where
salary = (SELECT AVG(salary) FROM EMPLOYEE);
```

| fname | sex | address | salary |
|---|---|---|---|

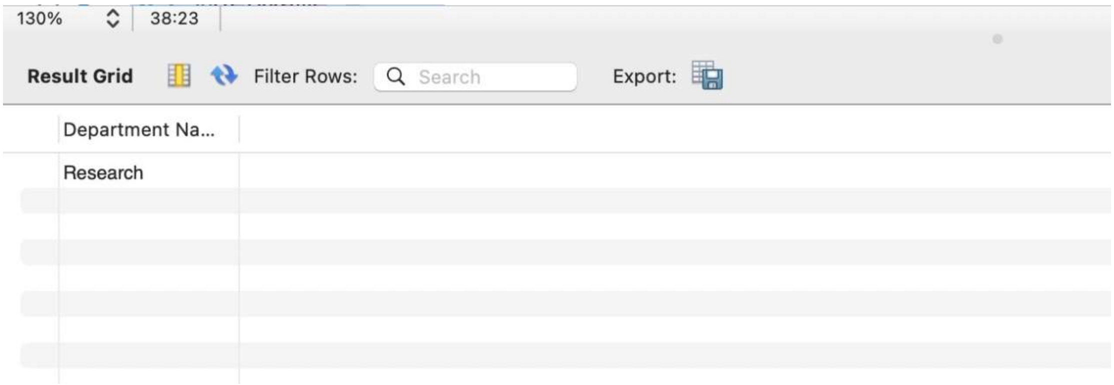


```sql
USE CompanyManagmantSystem;
select distinct EMPLOYEE.fname,DEPARTMENT.dname
from EMPLOYEE
inner join DEPARTMENT;
```

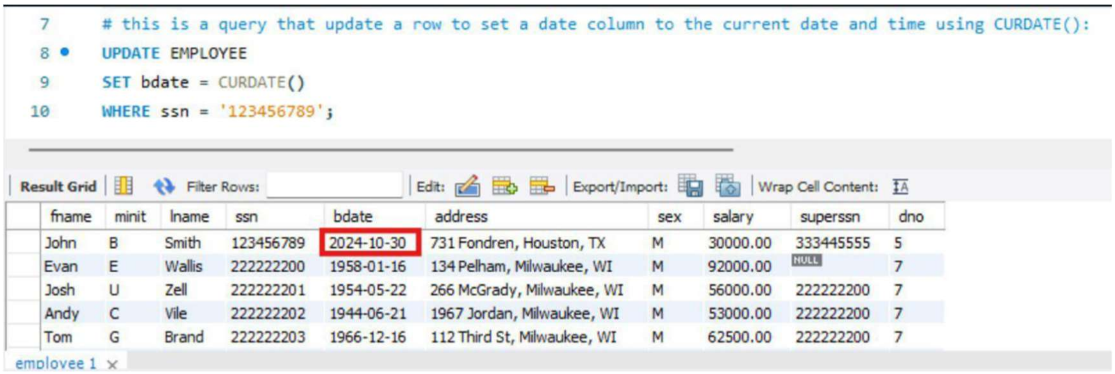
| fname | dname |
|---|---|
| Jared | Software |
| Jared | Sales |
| Jared | Research |
| Jared | Headquarters |
| Jared | Hardware |
| Jared | Administration |
| Jon | Software |
| Jon | Sales |
| Jon | Research |
| Jon | Headquarters |
| Jon | Hardware |
| Jon | Administration |
| Justin | Software |
| Justin | Sales |
| Justin | Research |
| Justin | Headquarters |
| Justin | Hardware |
| Justin | Administration |
| Brad | Software |



```sql
USE CompanyManagmantSystem;
/*Imagine sara works at a company with many departments, each identified by a number.
sara only has the department number but needs the department name.
 To make her life easier, the database has a stored procedure,
 GetDepartmentName, that instantly converts department numbers into names.
 */
DELIMITER //
CREATE PROCEDURE GetDepartmentName(IN depNum INT, OUT depName VARCHAR(25))
BEGIN
    CASE depNum
        WHEN 1 THEN SET depName = 'Research';
        WHEN 2 THEN SET depName = 'Sales';
        WHEN 3 THEN SET depName = 'Administration';
        WHEN 4 THEN SET depName = 'Software';
        WHEN 5 THEN SET depName = 'Headquarters';
        ELSE SET depName = 'Unknown';
    END CASE;
END //
DELIMITER ;

SET @depName = '';
CALL GetDepartmentName(5, @depName);
```

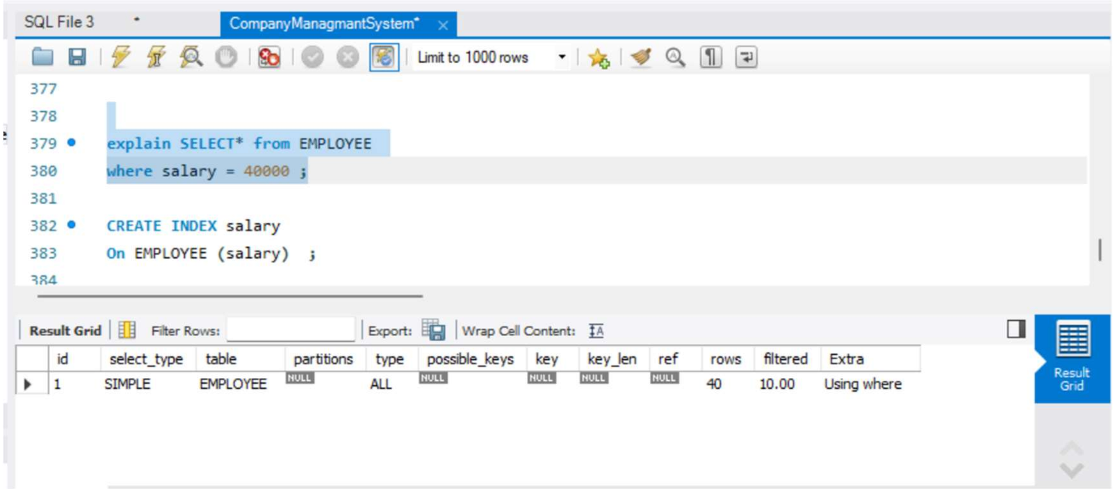


| Department Na... |
|---|
| Research |



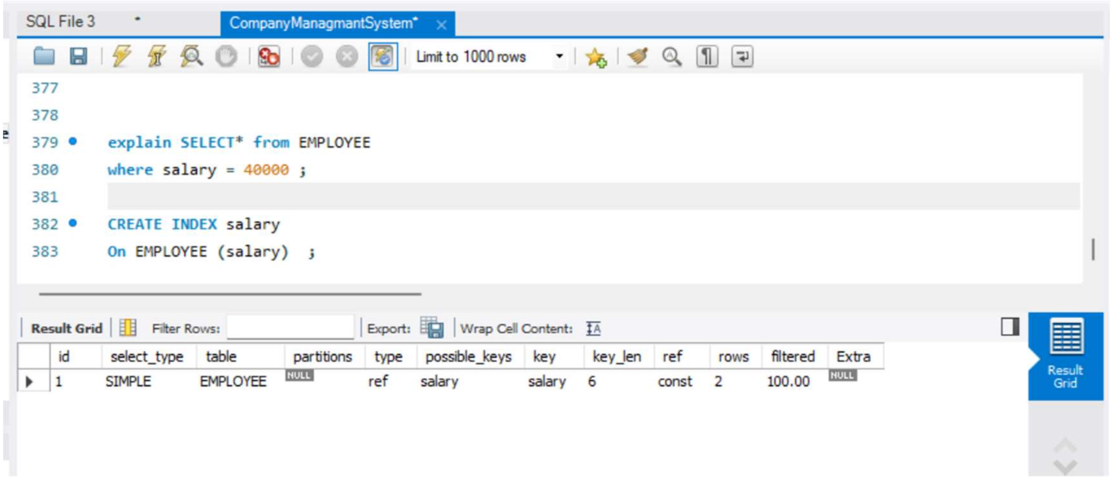
```sql
# this is a query that update a row to set a date column to the current date and time using CURDATE():
UPDATE EMPLOYEE
SET bdate = CURDATE()
WHERE ssn = '123456789';
```

| fname | minit | lname | ssn | bdate | address | sex | salary | superssn | dno |
|---|---|---|---|---|---|---|---|---|---|
| John | B | Smith | 123456789 | 2024-10-30 | 731 Fondren, Houston, TX | M | 30000.00 | 333445555 | 5 |
| Evan | E | Wallis | 222222200 | 1958-01-16 | 134 Pelham, Milwaukee, WI | M | 92000.00 | NULL | 7 |
| Josh | U | Zell | 222222201 | 1954-05-22 | 266 McGrady, Milwaukee, WI | M | 56000.00 | 222222200 | 7 |
| Andy | C | Vile | 222222202 | 1944-06-21 | 1967 Jordan, Milwaukee, WI | M | 53000.00 | 222222200 | 7 |
| Tom | G | Brand | 222222203 | 1966-12-16 | 112 Third St, Milwaukee, WI | M | 62500.00 | 222222200 | 7 |



```sql
explain SELECT* from EMPLOYEE
where salary = 40000 ;

CREATE INDEX salary
On EMPLOYEE (salary)  ;
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | SIMPLE | EMPLOYEE | NULL | ALL | NULL | NULL | NULL | NULL | 40 | 10.00 | Using where |



```sql
explain SELECT* from EMPLOYEE
where salary = 40000 ;

CREATE INDEX salary
On EMPLOYEE (salary)  ;
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | SIMPLE | EMPLOYEE | NULL | ref | salary | salary | 6 | const | 2 | 100.00 | NULL |

# Conclusion: -

The Company Management System offers an organized method for overseeing major elements of a company, such as personnel, divisions, assignments, and sites. The system makes it possible to efficiently track personnel roles, project progress, and departmental duties by clearly defining the links between these entities. This integrated system facilitates simple access to crucial business data, which not only helps to expedite operations but also enables informed decision-making. In the end, it is an essential instrument for enhancing organizational effectiveness and guaranteeing efficient resource management.