

## **ASSIGNMENT-2**

### **Assignment2.sql:**

use AssignmentDB

-- Create EMP Table

```
create table EMP (  
    EMPNO INT PRIMARY KEY,  
    ENAME VARCHAR(20),  
    JOB VARCHAR(20),  
    MGR_ID INT,  
    HIREDATE DATE,  
    SAL DECIMAL(7, 2),  
    COMM DECIMAL(7, 2),  
    DEPTNO INT,  
    FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO)  
);
```

-- Insert data into EMP

```
insert into EMP values  
(7369, 'SMITH', 'CLERK', 7902, '1980-12-17', 800, NULL, 20),  
(7499, 'ALLEN', 'SALESMAN', 7698, '1981-02-20', 1600, 300, 30),  
(7521, 'WARD', 'SALESMAN', 7698, '1981-02-22', 1250, 500, 30),  
(7566, 'JONES', 'MANAGER', 7839, '1981-04-02', 2975, NULL, 20),  
(7654, 'MARTIN', 'SALESMAN', 7698, '1981-09-28', 1250, 1400, 30),  
(7698, 'BLAKE', 'MANAGER', 7839, '1981-05-01', 2850, NULL, 30),  
(7782, 'CLARK', 'MANAGER', 7839, '1981-06-09', 2450, NULL, 10),  
(7788, 'SCOTT', 'ANALYST', 7566, '1987-04-19', 3000, NULL, 20),  
(7839, 'KING', 'PRESIDENT', NULL, '1981-11-17', 5000, NULL, 10),  
(7844, 'TURNER', 'SALESMAN', 7698, '1981-09-08', 1500, 0, 30),
```

```
(7876, 'ADAMS', 'CLERK', 7788, '1987-05-23', 1100, NULL, 20),  
(7900, 'JAMES', 'CLERK', 7698, '1981-12-03', 950, NULL, 30),  
(7902, 'FORD', 'ANALYST', 7566, '1981-12-03', 3000, NULL, 20),  
(7934, 'MILLER', 'CLERK', 7782, '1982-01-23', 1300, NULL, 10);
```

```
select * from EMP;
```

```
-- Create DEPT Table
```

```
create table DEPT (  
    DEPTNO INT PRIMARY KEY,  
    DNAME VARCHAR(20),  
    LOC VARCHAR(20)  
);
```

```
-- Insert data into DEPT
```

```
insert into DEPT values  
(10, 'ACCOUNTING', 'NEW YORK'),  
(20, 'RESEARCH', 'DALLAS'),  
(30, 'SALES', 'CHICAGO'),  
(40, 'OPERATIONS', 'BOSTON');
```

```
select * from DEPT;
```

```
-- 1. List all employees whose name begins with 'A'.
```

```
select * from EMP where ENAME LIKE 'A%';
```

```
-- 2. Select all those employees who don't have a manager.
```

```
select * from EMP where MGR_ID IS NULL;
```

-- 3. List employee name, number and salary for those employees who earn in the range 1200 to 1400.

```
select ENAME, EMPNO, SAL from EMP where SAL BETWEEN 1200 AND 1400;
```

-- 4. Give all the employees in the RESEARCH department a 10% pay rise. Verify that this has been done by listing all their details before and after the rise.

-- Before the rise

```
select * from EMP where DEPTNO = 20;
```

-- Apply the pay rise

```
update EMP set SAL = SAL * 1.10 where DEPTNO = 20;
```

-- After the rise

```
select * from EMP where DEPTNO = 20;
```

-- 5. Find the number of CLERKS employed. Give it a descriptive heading.

```
select COUNT(*) AS NumberOfClerks from EMP where JOB = 'CLERK';
```

-- 6. Find the average salary for each job type and the number of people employed in each job.

```
select JOB, AVG(SAL) AS AverageSalary, COUNT(*) AS NumberOfEmployees  
from EMP  
GROUP BY JOB;
```

-- 7. List the employees with the lowest and highest salary.

-- Employee with the lowest salary

```
select * from EMP where SAL = (select MIN(SAL) from EMP);
```

-- Employee with the highest salary

```
select * from EMP where SAL = (select MAX(SAL) from EMP);
```

-- 8. List full details of departments that don't have any employees.

```
select * from DEPT
```

```
where DEPTNO NOT IN (select DISTINCT DEPTNO from EMP);
```

-- 9. Get the names and salaries of all the analysts earning more than 1200 who are based in department 20. Sort the answer by ascending order of name.

```
select ENAME, SAL from EMP
```

```
where JOB = 'ANALYST' AND SAL > 1200 AND DEPTNO = 20
```

```
ORDER BY ENAME ASC;
```

-- 10. For each department, list its name and number together with the total salary paid to employees in that department.

```
select D.DEPTNO, D.DNAME, SUM(E.SAL) AS TotalSalary
```

```
from DEPT D
```

```
LEFT JOIN EMP E ON D.DEPTNO = E.DEPTNO
```

```
GROUP BY D.DEPTNO, D.DNAME;
```

-- 11. Find out salary of both MILLER and SMITH.

```
select ENAME, SAL from EMP where ENAME IN ('MILLER', 'SMITH');
```

-- 12. Find out the names of the employees whose name begin with 'A' or 'M'.

```
select ENAME from EMP where ENAME LIKE 'A%' OR ENAME LIKE 'M%';
```

-- 13. Compute yearly salary of SMITH.

```
select ENAME, SAL * 12 AS YearlySalary from EMP where ENAME = 'SMITH';
```

-- 14. List the name and salary for all employees whose salary is not in the range of 1500 and 2850.

```
SELECT ENAME, SAL FROM EMP WHERE SAL NOT BETWEEN 1500 AND 2850;
```

-- 15. Find all managers who have more than 2 employees reporting to them

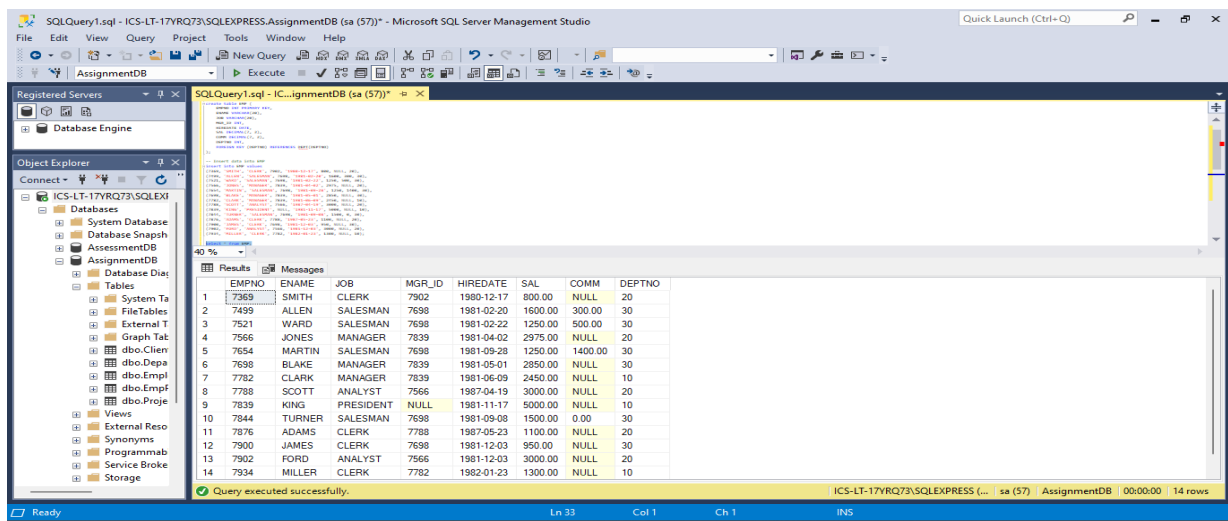
```

select MGR_ID, COUNT(*) AS NumberOfEmployees
from EMP where MGR_ID IS NOT NULL
GROUP BY MGR_ID
HAVING COUNT(*) > 2;

```

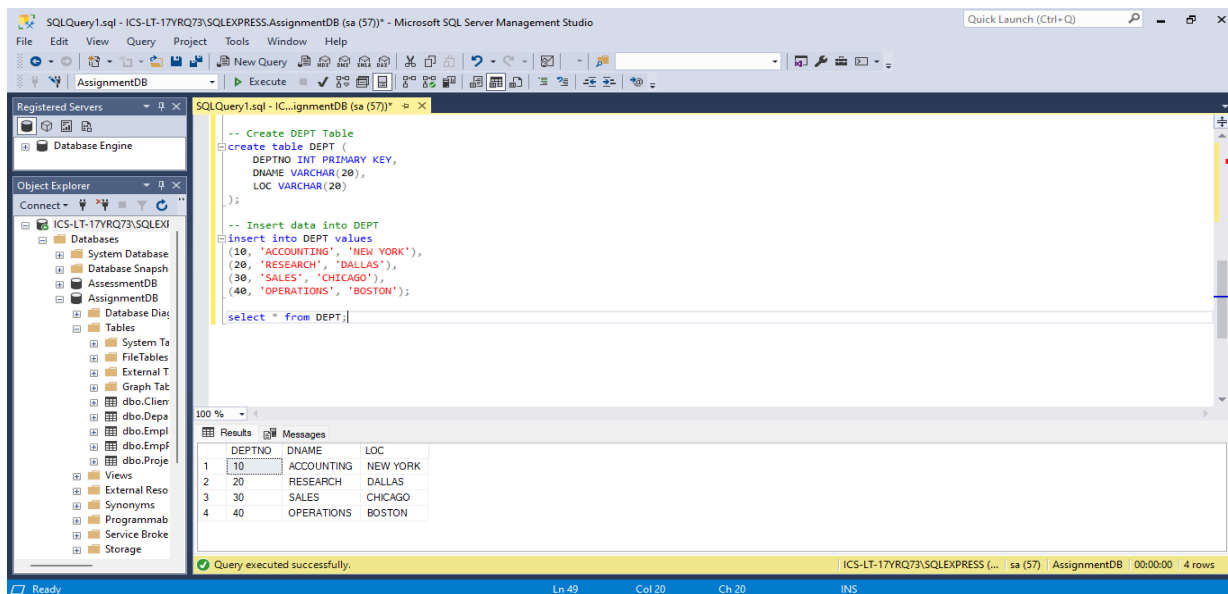
## Queries Snapshots:

### Employee Table



EMPNO	ENAME	JOB	MGR_ID	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
7788	SCOTT	ANALYST	7566	1987-04-19	3000.00	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1987-05-23	1100.00	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950.00	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3000.00	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10

### Department Table



```

-- Create DEPT Table
create table DEPT (
    DEPTNO INT PRIMARY KEY,
    DNAME VARCHAR(20),
    LOC VARCHAR(20)
);

-- Insert data into DEPT
insert into DEPT values
(10, 'ACCOUNTING', 'NEW YORK'),
(20, 'RESEARCH', 'DALLAS'),
(30, 'SALES', 'CHICAGO'),
(40, 'OPERATIONS', 'BOSTON');

select * from DEPT;

```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

1. List all employees whose name begins with 'A'.

SQLQuery1.sql - ICS-LT-17YRQ73\SQLEXPRESS.AssignmentDB (sa (57)) - Microsoft SQL Server Management Studio

Registered Servers: Database Engine

Object Explorer: ICS-LT-17YRQ73\SQLEXPRESS

Query: `select * from DEPT;`

-- 1. List all employees whose name begins with 'A'.

Query: `select * from EMP where ENAME LIKE 'A%';`

Results:

	EMPNO	ENAME	JOB	MGR_ID	HIREDATE	SAL	COMM	DEPTNO
1	7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
2	7876	ADAMS	CLERK	7788	1987-05-23	1100.00	NULL	20

Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (57)) AssignmentDB 00:00:00 2 rows

2. Select all those employees who don't have a manager.

SQLQuery1.sql - ICS-LT-17YRQ73\SQLEXPRESS.AssignmentDB (sa (57)) - Microsoft SQL Server Management Studio

Registered Servers: Database Engine

Object Explorer: ICS-LT-17YRQ73\SQLEXPRESS

Query: `-- 2. Select all those employees who don't have a manager.`

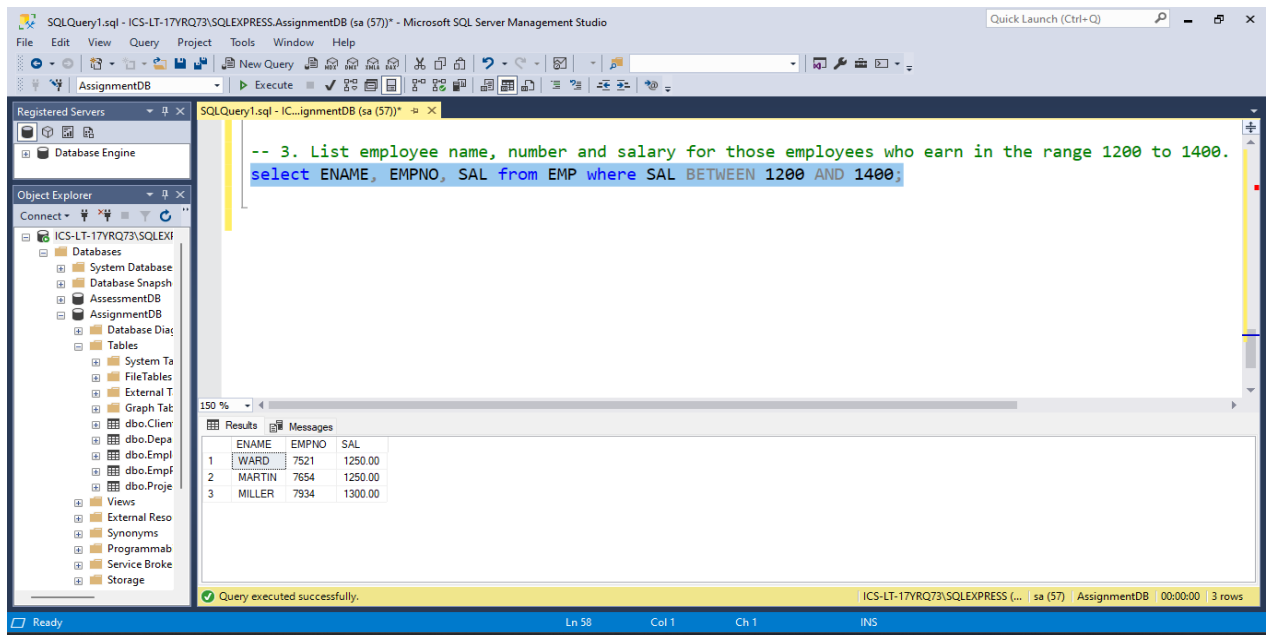
Query: `select * from EMP where MGR_ID IS NULL;`

Results:

	EMPNO	ENAME	JOB	MGR_ID	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10

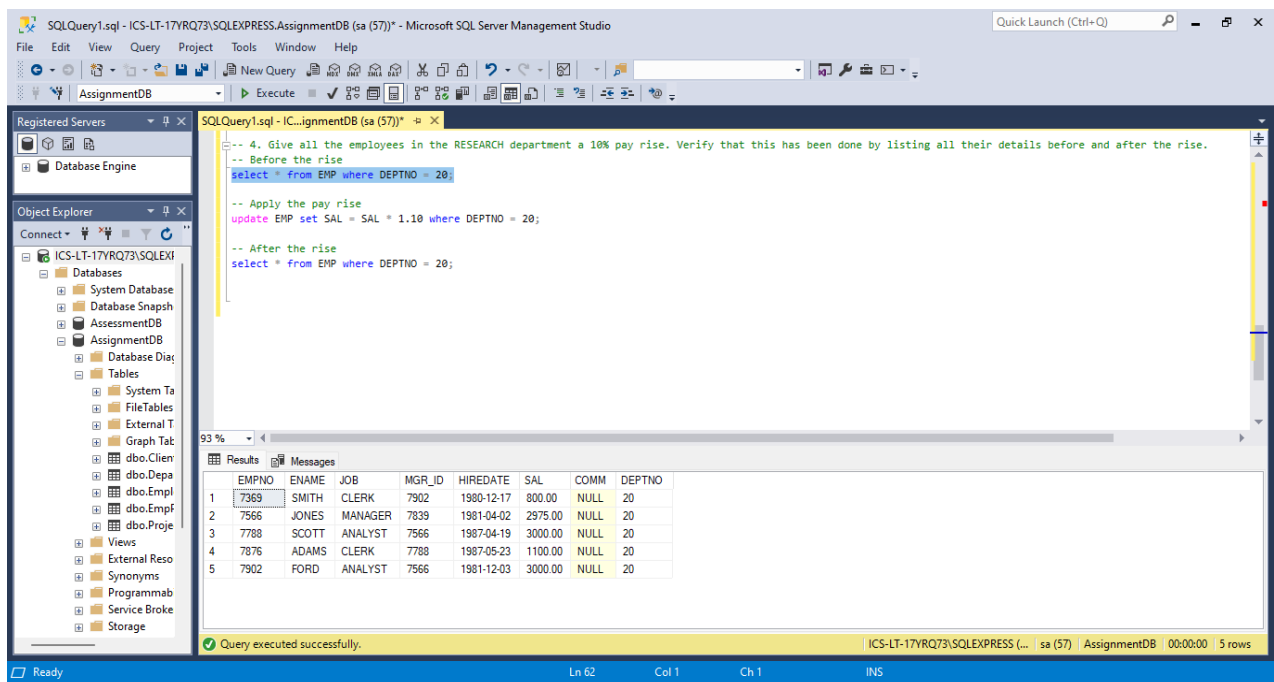
Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (57)) AssignmentDB 00:00:00 1 rows

3. List employee name, number and salary for those employees who earn in the range 1200 to 1400.



4. Give all the employees in the RESEARCH department a 10% pay rise. Verify that this has been done by listing all their details before and after the rise.

i) Before the rise



ii) After the rise

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following SQL script:

```
-- 4. Give all the employees in the RESEARCH department a 10% pay rise. Verify that this has been done by listing all their details before and after the rise.
-- Before the rise
select * from EMP where DEPTNO = 20;

-- Apply the pay rise
update EMP set SAL = SAL * 1.10 where DEPTNO = 20;

-- After the rise
select * from EMP where DEPTNO = 20;
```

The results pane shows the data after the update. The status bar indicates "Query executed successfully." and "5 rows".

EMPNO	ENAME	JOB	MGR_ID	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	880.00	NULL	20
7566	JONES	MANAGER	7839	1981-04-02	3272.50	NULL	20
7788	SCOTT	ANALYST	7566	1987-04-19	3300.00	NULL	20
7876	ADAMS	CLERK	7788	1987-05-23	1210.00	NULL	20
7902	FORD	ANALYST	7566	1981-12-03	3300.00	NULL	20

5. Find the number of CLERKS employed. Give it a descriptive heading.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following SQL script:

```
-- 5. Find the number of CLERKS employed. Give it a descriptive heading.
select COUNT(*) AS NumberOfClerks from EMP where JOB = 'CLERK';
```

The results pane shows the result of the query. The status bar indicates "Query executed successfully." and "1 rows".

NumberOfClerks
4

6. Find the average salary for each job type and the number of people employed in each job.



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
-- 6. Find the average salary for each job type and the number of people employed in each job.
select JOB, AVG(SAL) AS AverageSalary, COUNT(*) AS NumberOfEmployees
from EMP
GROUP BY JOB;
```

The query has been executed successfully, and the results are displayed in the Results pane:

JOB	AverageSalary	NumberOfEmployees
ANALYST	3300.000000	2
CLERK	1085.000000	4
MANAGER	2857.500000	3
PRESIDENT	5000.000000	1
SALESMAN	1400.000000	4

The status bar at the bottom indicates: "Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (57)) AssignmentDB 00:00:00 5 rows".

7. List the employees with the lowest and highest salary.

i) Employee with lowest salary

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
-- 7. List the employees with the lowest and highest salary.
-- Employee with the lowest salary
SELECT * FROM EMP WHERE SAL = (SELECT MIN(SAL) FROM EMP);

-- Employee with the highest salary
SELECT * FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP);
```

The query has been executed successfully, and the results are displayed in the Results pane:

EMPNO	ENAME	JOB	MGR_ID	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	880.00	NULL	20

The status bar at the bottom indicates: "Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (57)) AssignmentDB 00:00:00 1 rows".

ii) Employee with highest salary

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```

SELECT JOB, AVG(SAL) AS AverageSalary, COUNT(*) AS Number of Employees
FROM EMP
GROUP BY JOB;

-- 7. List the employees with the lowest and highest salary.
-- Employee with the lowest salary
SELECT * FROM EMP WHERE SAL = (SELECT MIN(SAL) FROM EMP);

-- Employee with the highest salary
SELECT * FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP);

```

The Results pane shows the output of the first query, displaying a single row for the 'PRESIDENT' job:

EMPNO	ENAME	JOB	MGR_ID	HIREDATE	SAL	COMM	DEPTNO	
1	7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10

The status bar at the bottom indicates the query was executed successfully, returning 1 row.

8. List full details of departments that don't have any employees.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```

-- 8. List full details of departments that don't have any employees.
select * from DEPT
where DEPTNO NOT IN (select DISTINCT DEPTNO from EMP);

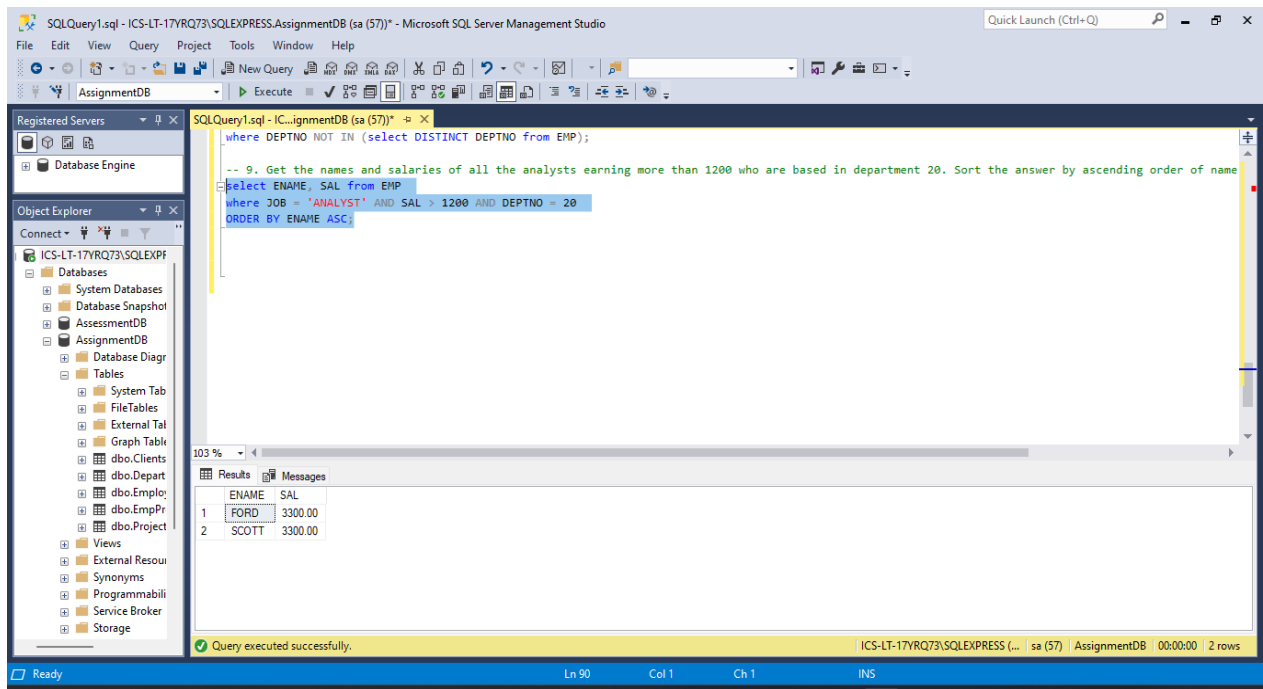
```

The Results pane shows the output of the query, displaying a single row for department 40:

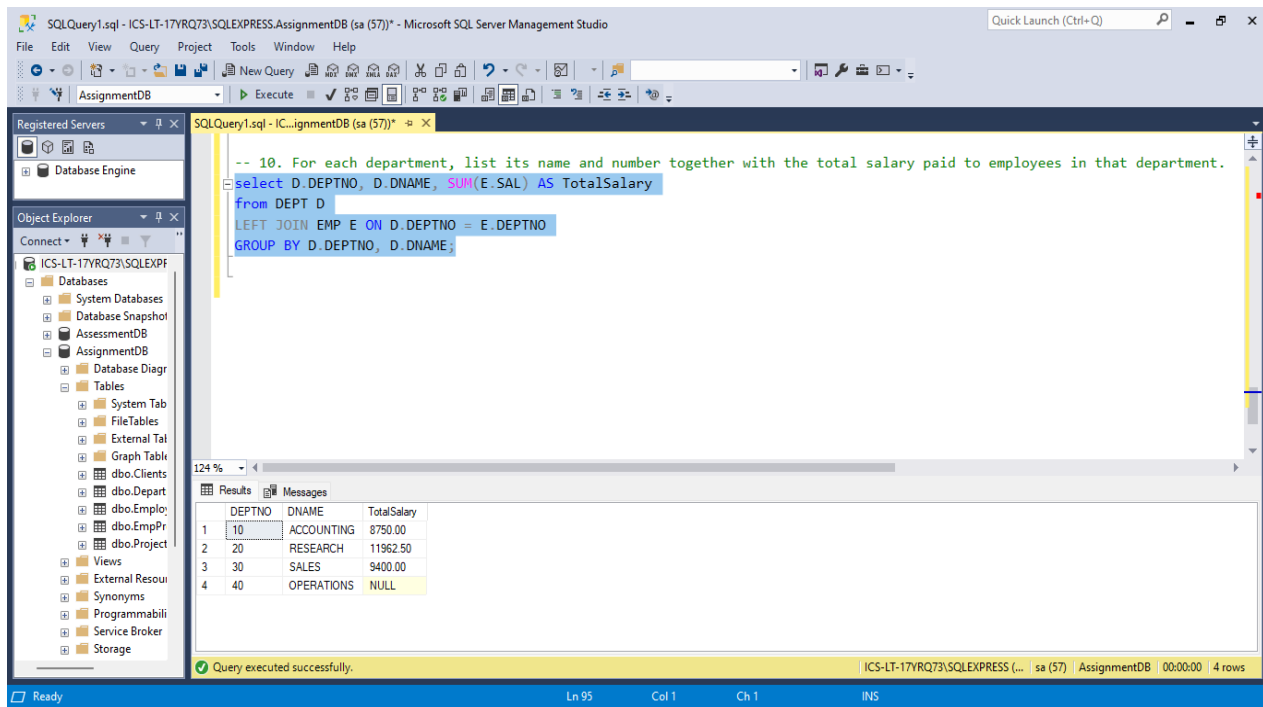
DEPTNO	DNAME	LOC
1	40	OPERATIONS BOSTON

The status bar at the bottom indicates the query was executed successfully, returning 1 row.

9. Get the names and salaries of all the analysts earning more than 1200 who are based in department 20. Sort the answer by ascending order of name.



10. For each department, list its name and number together with the total salary paid to employees in that department.



11. Find out salary of both MILLER and SMITH.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'SQLQuery1.sql - ICS-LT-17YRQ73\SQLEXPRESS.AssessmentDB (sa (57))'. The Object Explorer on the left shows the database structure, including 'AssessmentDB' and its tables. The main query editor contains the following SQL code:

```
-- 11. Find out salary of both MILLER and SMITH.  
select ENAME, SAL from EMP where ENAME IN ('MILLER', 'SMITH');
```

The query has been executed successfully, and the results are displayed in the 'Results' pane. The results show two rows of data:

	ENAME	SAL
1	SMITH	880.00
2	MILLER	1300.00

The status bar at the bottom indicates 'Query executed successfully.' and '2 rows'.

12. Find out the names of the employees whose name begin with 'A' or 'M'.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'SQLQuery1.sql - ICS-LT-17YRQ73\SQLEXPRESS.AssessmentDB (sa (57))'. The Object Explorer on the left shows the database structure. The main query editor contains the following SQL code:

```
-- 12. Find out the names of the employees whose name begin with 'A' or 'M'.  
select ENAME from EMP where ENAME LIKE 'A%' OR ENAME LIKE 'M%';
```

The query has been executed successfully, and the results are displayed in the 'Results' pane. The results show four rows of data:

	ENAME
1	ALLEN
2	MARTIN
3	ADAMS
4	MILLER

The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

13. Compute yearly salary of SMITH.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
-- 13. Compute yearly salary of SMITH.  
select ENAME, SAL * 12 AS YearlySalary from EMP where ENAME = 'SMITH';
```

The query has been executed successfully. The Results pane shows the following data:

ENAME	YearlySalary
SMITH	10560.00

The status bar at the bottom indicates: "Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (57)) AssignmentDB 00:00:00 1 rows".

14. List the name and salary for all employees whose salary is not in the range of 1500 and 2850.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
-- 14. List the name and salary for all employees whose salary is not in the range of 1500 and 2850.  
SELECT ENAME, SAL FROM EMP WHERE SAL NOT BETWEEN 1500 AND 2850;
```

The query has been executed successfully. The Results pane shows the following data:

ENAME	SAL
SMITH	800.00
WARD	1250.00
JONES	3272.50
MARTIN	1250.00
SCOTT	3300.00
KING	5000.00
ADAMS	1210.00
JAMES	950.00
FORD	3300.00
MILLER	1300.00

The status bar at the bottom indicates: "Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (57)) AssignmentDB 00:00:00 10 rows".

15. Find all managers who have more than 2 employees reporting to them

SQLQuery1.sql - ICS-LT-17YRQ73\SQLEXPRESS.AssignmentDB (sa (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

AssignmentDB

Registered Servers

Database Engine

Object Explorer

Connect

ICS-LT-17YRQ73\SQLEXPRESS

Databases

- System Databases
- Database Snapshot
- AssessmentDB
- AssignmentDB
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - External Tables
    - Graph Tables
    - dbo.Clients
    - dbo.Departments
    - dbo.Employees
    - dbo.EmployeeProjects
    - dbo.Projects
  - Views
  - External Resources
  - Synonyms
  - Programmability
  - Service Broker
  - Storage

SQLQuery1.sql - ICS-LT-17YRQ73\SQLEXPRESS.AssignmentDB (sa (57))

```
-- 15. Find all managers who have more than 2 employees reporting to them
select MGR_ID, COUNT(*) AS NumberOfEmployees
from EMP where MGR_ID IS NOT NULL
GROUP BY MGR_ID
HAVING COUNT(*) > 2;
```

150 %

Results Messages

	MGR_ID	NumberOfEmployees
1	7698	5
2	7839	3

Query executed successfully.

ICS-LT-17YRQ73\SQLEXPRESS (sa (57) AssignmentDB 00:00:00 2 rows

Ready Ln 113 Col 1 Ch 1 INS