

ASSIGNMENT-3

Assignment3.sql:

```
use AssignmentDB
```

```
select * from EMP;
```

```
select * from DEPT;
```

```
-- 1. Retrieve a list of MANAGERS.
```

```
select DISTINCT E1.MGR_ID, E2.ENAME AS MANAGER_NAME  
from EMP E1  
JOIN EMP E2 ON E1.MGR_ID = E2.EMPNO  
where E1.MGR_ID IS NOT NULL;
```

```
-- 2. Find out the names and salaries of all employees earning more than 1000 per month.
```

```
select ENAME, SAL from EMP where SAL > 1000;
```

```
-- 3. Display the names and salaries of all employees except JAMES.
```

```
select ENAME, SAL from EMP where ENAME NOT IN ('JAMES');
```

```
-- 4. Find out the details of employees whose names begin with 'S'.
```

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

```
-- 5. Find out the names of all employees that have 'A' anywhere in their name.
```

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

```
-- 6. Find out the names of all employees that have 'L' as their third character in their name.
```

```
select ENAME from EMP where ENAME LIKE '__L%';
```

```
-- 7. Compute daily salary of JONES.
```

```
select ENAME, SAL/30 AS DAILY_SALARY from EMP where ENAME = 'JONES';
```

-- 8. Calculate the total monthly salary of all employees.

```
select SUM(SAL) AS TOTAL_MONTHLY_SALARY from EMP;
```

-- 9. Print the average annual salary.

```
select AVG(SAL) * 12 AS AVG_ANNUAL_SALARY from EMP;
```

-- 10. Select the name, job, salary, department number of all employees except SALESMAN from department number 30.

```
select ENAME, JOB, SAL, DEPTNO
```

```
from EMP
```

```
where DEPTNO = 30 AND JOB NOT IN ('SALESMAN');
```

-- 11. List unique departments of the EMP table.

```
select DISTINCT E.DEPTNO, D.DNAME
```

```
from EMP E
```

```
JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
```

-- 12. List the name and salary of employees who earn more than 1500 and are in department 10 or 30. Label the columns Employee and Monthly Salary respectively.

```
select ENAME AS Employee, SAL AS "Monthly Salary"
```

```
from EMP
```

```
where SAL > 1500 AND DEPTNO IN (10, 30);
```

-- 13. Display the name, job, and salary of all the employees whose job is MANAGER or ANALYST and their salary is not equal to 1000, 3000, or 5000.

```
select ENAME, JOB, SAL
```

```
from EMP
```

```
where (JOB = 'MANAGER' OR JOB = 'ANALYST') AND SAL NOT IN (1000, 3000, 5000);
```

-- 14. Display the name, salary and commission for all employees whose commission amount is greater than their salary increased by 10%.

```
select ENAME, SAL, COMM
```

```
from EMP
```

```
where COMM > SAL * 1.0;
```

-- 15. Display the name of all employees who have two Ls in their name and are in department 30 or their manager is 7782.

```
select ENAME
from EMP
where (ENAME LIKE '%L%L%' AND DEPTNO = 30) OR MGR_ID = 7782;
```

-- 16. Display the names of employees with experience of over 30 years and under 40 yrs. Count the total number of employees.

```
WITH ExperiencedEmployees AS (
    select ENAME
    from EMP
    where DATEDIFF(YEAR, HIREDATE, GETDATE()) > 30
    AND DATEDIFF(YEAR, HIREDATE, GETDATE()) < 40
)
select ENAME, (select COUNT(*) from ExperiencedEmployees) AS
TOTAL_EMPLOYEES
from ExperiencedEmployees;
```

-- 17. Retrieve the names of departments in ascending order and their employees in descending order.

```
select D.DNAME, E.ENAME
from DEPT D
JOIN EMP E ON D.DEPTNO = E.DEPTNO
ORDER BY D.DNAME ASC, E.ENAME DESC;
```

-- 18. Find out experience of MILLER.

```
select ENAME, DATEDIFF(YEAR, HIREDATE, GETDATE()) AS EXPERIENCE_YEARS
from EMP
where ENAME = 'MILLER';
```

Queries Snapshots:

Employee Table

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

```
use AssignmentDB;

select * from EMP;

select * from DEPT;
```

The Results pane displays the data from the EMP table. The status bar at the bottom indicates "Query executed successfully." and "14 rows".

EMPNO	ENAME	JOB	MGR_ID	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	968.00	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	3599.75	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
7789	SCOTT	ANALYST	7566	1987-04-19	3630.00	NULL	20
7839	KING	PRESIDE...	NULL	1981-11-17	5000.00	NULL	10
7844	TURIN...	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1987-05-23	1331.00	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950.00	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3630.00	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10

Department Table:

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

```
use AssignmentDB;

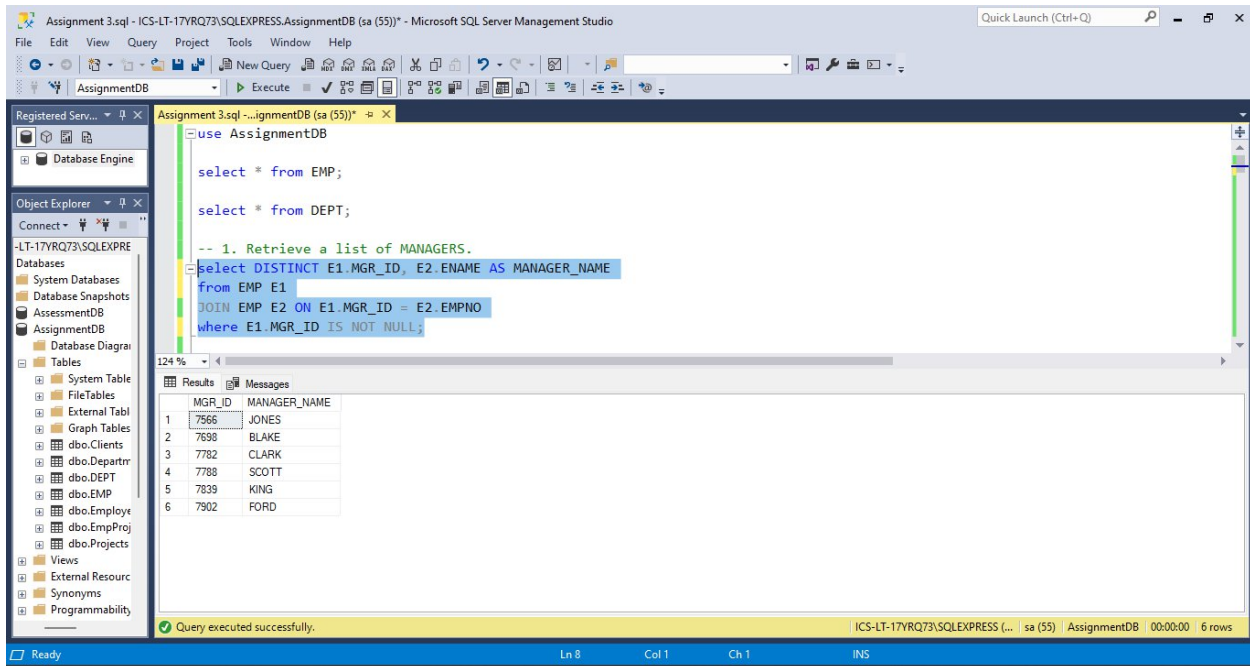
select * from EMP;

select * from DEPT;
```

The Results pane displays the data from the DEPT table. The status bar at the bottom indicates "Query executed successfully." and "4 rows".

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

1. Retrieve a list of MANAGERS.



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

```
use AssignmentDB

select * from EMP;

select * from DEPT;

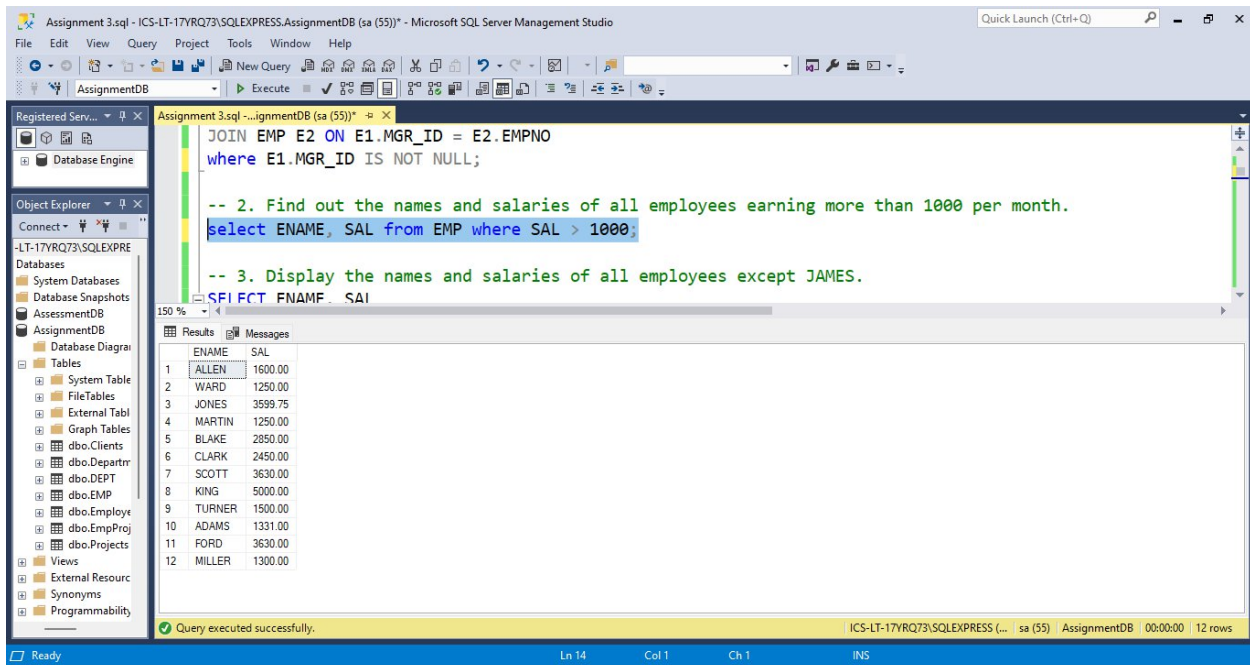
-- 1. Retrieve a list of MANAGERS.
select DISTINCT E1.MGR_ID, E2.ENAME AS MANAGER_NAME
from EMP E1
JOIN EMP E2 ON E1.MGR_ID = E2.EMPNO
where E1.MGR_ID IS NOT NULL;
```

The query results are displayed in the Results pane, showing a list of managers with their MGR_ID and MANAGER_NAME:

MGR_ID	MANAGER_NAME
7566	JONES
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7902	FORD

The status bar at the bottom indicates the query was executed successfully, returning 6 rows.

2. Find out the names and salaries of all employees earning more than 1000 per month.



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

```
JOIN EMP E2 ON E1.MGR_ID = E2.EMPNO
where E1.MGR_ID IS NOT NULL;

-- 2. Find out the names and salaries of all employees earning more than 1000 per month.
select ENAME, SAL from EMP where SAL > 1000;

-- 3. Display the names and salaries of all employees except JAMES.
SELECT ENAME, SAL
```

The query results are displayed in the Results pane, showing a list of employees with their ENAME and SAL:

ENAME	SAL
ALLEN	1600.00
WARD	1250.00
JONES	3599.75
MARTIN	1250.00
BLAKE	2850.00
CLARK	2450.00
SCOTT	3630.00
KING	5000.00
TURNER	1500.00
ADAMS	1331.00
FORD	3630.00
MILLER	1300.00

The status bar at the bottom indicates the query was executed successfully, returning 12 rows.

3. Display the names and salaries of all employees except JAMES.

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

```
-- 2. Find out the names and salaries of all employees earning more than 1000 per month.
select ENAME, SAL from EMP where SAL > 1000;

-- 3. Display the names and salaries of all employees except JAMES.
select ENAME, SAL from EMP where ENAME NOT IN ('JAMES');

-- 4. Find out the details of employees whose names begin with 'S'.
```

The Results pane displays the output of the first query, showing a list of employees with their names (ENAME) and salaries (SAL):

ENAME	SAL
SMITH	968.00
ALLEN	1600.00
WARD	1250.00
JONES	3599.75
MARTIN	1250.00
BLAKE	2850.00
CLARK	2450.00
SCOTT	3630.00
KING	5000.00
TURNER	1500.00
ADAMS	1331.00
FORD	3630.00
MILLER	1300.00

The status bar at the bottom indicates "Query executed successfully." and "13 rows".

4. Find out the details of employees whose names begin with 'S'.

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

```
-- 3. Display the names and salaries of all employees except JAMES.
select ENAME, SAL from EMP where ENAME NOT IN ('JAMES');

-- 4. Find out the details of employees whose names begin with 'S'.
select * from EMP where ENAME LIKE 'S%';

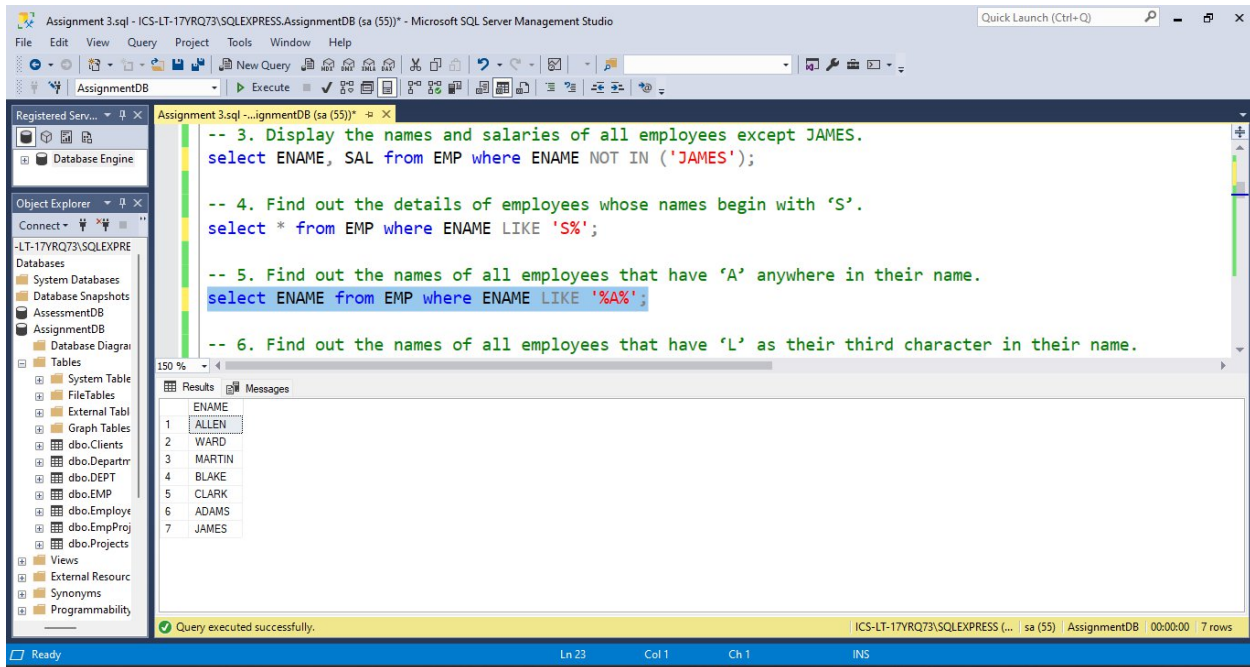
-- 5. Find out the names of all employees that have 'A' anywhere in their name.
select FNAME from EMP where FNAME LIKE '%A%';
```

The Results pane displays the output of the second query, showing a list of employees with their names (ENAME) and salaries (SAL):

ENAME	SAL
SMITH	968.00
SCOTT	3630.00

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

5. Find out the names of all employees that have 'A' anywhere in their name.



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'AssignmentDB' database selected. The right pane shows a query window with the following SQL code:

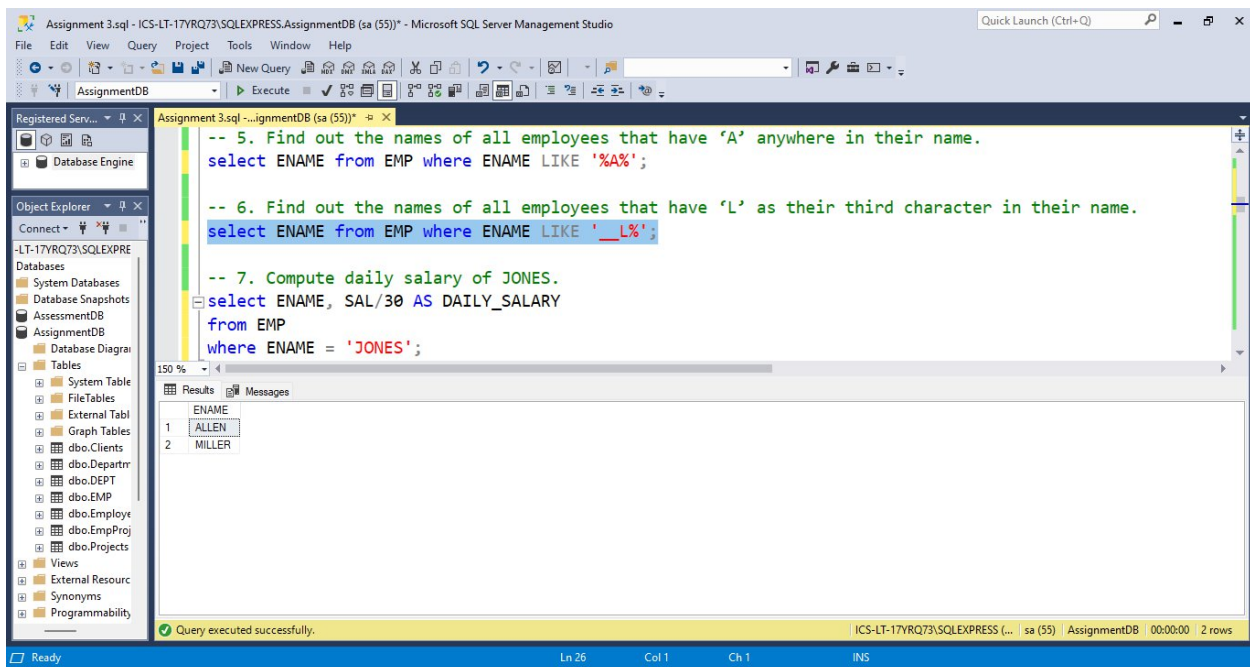
```
-- 3. Display the names and salaries of all employees except JAMES.  
select ENAME, SAL from EMP where ENAME NOT IN ('JAMES');  
  
-- 4. Find out the details of employees whose names begin with 'S'.  
select * from EMP where ENAME LIKE 'S%';  
  
-- 5. Find out the names of all employees that have 'A' anywhere in their name.  
select ENAME from EMP where ENAME LIKE '%A%';  
  
-- 6. Find out the names of all employees that have 'L' as their third character in their name.
```

The 'Results' pane shows the output of the third query, displaying a list of employee names:

ENAME
1 ALLEN
2 WARD
3 MARTIN
4 BLAKE
5 CLARK
6 ADAMS
7 JAMES

The status bar at the bottom indicates 'Query executed successfully.' and '7 rows'.

6. Find out the names of all employees that have 'L' as their third character in their name.



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'AssignmentDB' database selected. The right pane shows a query window with the following SQL code:

```
-- 5. Find out the names of all employees that have 'A' anywhere in their name.  
select ENAME from EMP where ENAME LIKE '%A%';  
  
-- 6. Find out the names of all employees that have 'L' as their third character in their name.  
select ENAME from EMP where ENAME LIKE '__L%';  
  
-- 7. Compute daily salary of JONES.  
select ENAME, SAL/30 AS DAILY_SALARY  
from EMP  
where ENAME = 'JONES';
```

The 'Results' pane shows the output of the sixth query, displaying a list of employee names:

ENAME
1 ALLEN
2 MILLER

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

7. Compute daily salary of JONES.

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

```
-- 5. Find out the names of all employees that have 'A' anywhere in their name.
select ENAME from EMP where ENAME LIKE '%A%';

-- 6. Find out the names of all employees that have 'L' as their third character in their name.
select ENAME from EMP where ENAME LIKE '__L%';

-- 7. Compute daily salary of JONES.
select ENAME, SAL/30 AS DAILY_SALARY from EMP where ENAME = 'JONES';

-- 8. Calculate the total monthly salary of all employees.
```

The query results are displayed in the Results pane, showing a single row for JONES with a daily salary of 119.991666.

ENAME	DAILY_SALARY
JONES	119.991666

The status bar at the bottom indicates the query was executed successfully.

8. Calculate the total monthly salary of all employees.

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

```
-- 6. Find out the names of all employees that have 'L' as their third character in their name.
select ENAME from EMP where ENAME LIKE '__L%';

-- 7. Compute daily salary of JONES.
select ENAME, SAL/30 AS DAILY_SALARY from EMP where ENAME = 'JONES';

-- 8. Calculate the total monthly salary of all employees.
select SUM(SAL) AS TOTAL_MONTHLY_SALARY from EMP;

-- 9. Print the average annual salary.
```

The query results are displayed in the Results pane, showing a single row for the total monthly salary of 31308.75.

TOTAL_MONTHLY_SALARY
31308.75

The status bar at the bottom indicates the query was executed successfully.

9. Print the average annual salary.

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

```
-- 7. Compute daily salary of JONES.
select ENAME, SAL/30 AS DAILY_SALARY from EMP where ENAME = 'JONES';

-- 8. Calculate the total monthly salary of all employees.
select SUM(SAL) AS TOTAL_MONTHLY_SALARY from EMP;

-- 9. Print the average annual salary.
select AVG(SAL) * 12 AS AVG_ANNUAL_SALARY from EMP;

-- 10. Select the name, job, salary, department number of all employees except SALESMAN from department 1
```

The query results pane shows the output for the third query:

AVG_ANNUAL_SALARY
26836.071420

The status bar at the bottom indicates "Query executed successfully." and "1 rows".

10. Select the name, job, salary, department number of all employees except SALESMAN from department number 30.

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

```
-- 9. Print the average annual salary.
select AVG(SAL) * 12 AS AVG_ANNUAL_SALARY from EMP;

-- 10. Select the name, job, salary, department number of all employees except SALESMAN from department 1
select ENAME, JOB, SAL, DEPTNO
from EMP
where DEPTNO = 30 AND JOB NOT IN ('SALESMAN');

-- 11. List unique departments of the EMP table.
select DISTINCT E.DEPTNO, D.DNAME
```

The query results pane shows the output for the second query:

ENAME	JOB	SAL	DEPTNO
BLAKE	MANAGER	2850.00	30
JAMES	CLERK	950.00	30

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

11. List unique departments of the EMP table.

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

```
-- 11. List unique departments of the EMP table.
select DISTINCT E.DEPTNO, D.DNAME
from EMP E
JOIN DEPT D ON E.DEPTNO = D.DEPTNO;

-- 12. List the name and salary of employees who earn more than 1500 and are in department 10 or 30. Label
select ENAME AS Employee, SAL AS "Monthly Salary"
from EMP
```

The Object Explorer on the left shows the database structure for 'AssignmentDB', including tables like EMP, DEPT, and various system tables. The Results pane at the bottom displays the output of the first query:

DEPTNO	DNAME
10	ACCOUNTING
20	RESEARCH
30	SALES

The status bar at the bottom indicates the query was executed successfully, returning 3 rows.

12. List the name and salary of employees who earn more than 1500 and are in department 10 or 30. Label the columns Employee and Monthly Salary respectively.

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

```
-- 12. List the name and salary of employees who earn more than 1500 and are in department 10 or 30. Label
select ENAME AS Employee, SAL AS "Monthly Salary"
from EMP
where SAL > 1500 AND DEPTNO IN (10, 30);

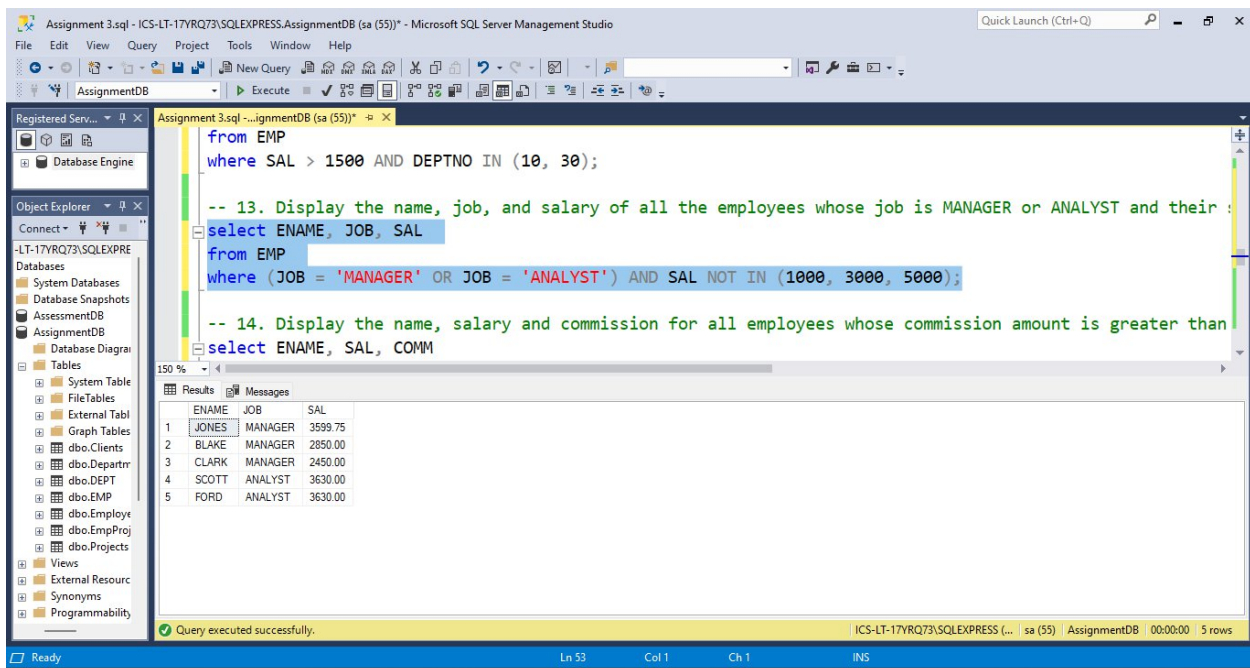
-- 13. Display the name, job, and salary of all the employees whose job is MANAGER or ANALYST and their
select ENAME, JOB, SAL
from EMP
where (JOB = 'MANAGER' OR JOB = 'ANALYST') AND SAL NOT IN (1000, 3000, 5000);
```

The Object Explorer on the left shows the database structure for 'AssignmentDB'. The Results pane at the bottom displays the output of the second query:

Employee	Monthly Salary
ALLEN	1600.00
BLAKE	2850.00
CLARK	2450.00
KING	5000.00

The status bar at the bottom indicates the query was executed successfully, returning 4 rows.

13. Display the name, job, and salary of all the employees whose job is MANAGER or ANALYST and their salary is not equal to 1000, 3000, or 5000.



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

```

from EMP
where SAL > 1500 AND DEPTNO IN (10, 30);

-- 13. Display the name, job, and salary of all the employees whose job is MANAGER or ANALYST and their :
select ENAME, JOB, SAL
from EMP
where (JOB = 'MANAGER' OR JOB = 'ANALYST') AND SAL NOT IN (1000, 3000, 5000);

-- 14. Display the name, salary and commission for all employees whose commission amount is greater than
select ENAME, SAL, COMM

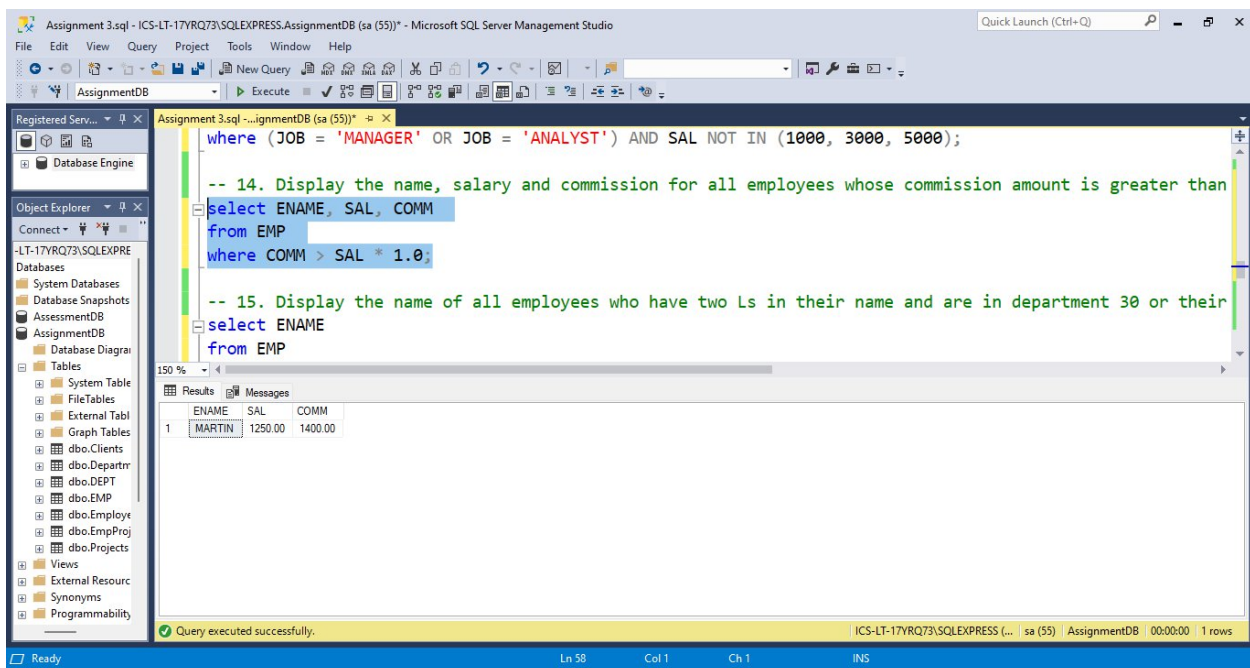
```

The Results pane displays the following data:

ENAME	JOB	SAL
JONES	MANAGER	3599.75
BLAKE	MANAGER	2850.00
CLARK	MANAGER	2450.00
SCOTT	ANALYST	3630.00
FORD	ANALYST	3630.00

Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (55)) | AssignmentDB 00:00:00 5 rows

14. Display the name, salary and commission for all employees whose commission amount is greater than their salary increased by 10%.



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

```

where (JOB = 'MANAGER' OR JOB = 'ANALYST') AND SAL NOT IN (1000, 3000, 5000);

-- 14. Display the name, salary and commission for all employees whose commission amount is greater than
select ENAME, SAL, COMM
from EMP
where COMM > SAL * 1.10;

-- 15. Display the name of all employees who have two Ls in their name and are in department 30 or their
select ENAME
from EMP

```

The Results pane displays the following data:

ENAME	SAL	COMM
MARTIN	1250.00	1400.00

Query executed successfully. ICS-LT-17YRQ73\SQLEXPRESS (sa (55)) | AssignmentDB 00:00:00 1 rows

15. Display the name of all employees who have two Ls in their name and are in department 30 or their manager is 7782.

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

```
select ENAME, SAL, COMM
from EMP
where COMM > SAL * 1.0;

-- 15. Display the name of all employees who have two Ls in their name and are in department 30 or their
select ENAME
from EMP
where (ENAME LIKE '%L%L%' AND DEPTNO = 30) OR MGR_ID = 7782;

-- 16. Display the names of employees with experience of over 30 years and under 40 yrs. Count the total
```

The Results pane shows the output of the second query:

ENAME
1 ALLEN
2 MILLER

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

16. Display the names of employees with experience of over 30 years and under 40 yrs. Count the total number of employees.

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

```
where (ENAME LIKE '%L%L%' AND DEPTNO = 30) OR MGR_ID = 7782;

-- 16. Display the names of employees with experience of over 30 years and under 40 yrs. Count the total number of employees.
WITH ExperiencedEmployees AS (
select ENAME
from EMP
where DATEDIFF(YEAR, HIREDATE, GETDATE()) > 30
AND DATEDIFF(YEAR, HIREDATE, GETDATE()) < 40
)
select ENAME, (select COUNT(*) from ExperiencedEmployees) AS TOTAL_EMPLOYEES
from ExperiencedEmployees;
```

The Results pane shows the output of the query:

ENAME	TOTAL_EMPLOYEES
1 SCOTT	2
2 ADAMS	2

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

17. Retrieve the names of departments in ascending order and their employees in descending order.

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The query editor displays the following SQL code:

```
select ENAME, (select COUNT(*) from ExperiencedEmployees) AS TOTAL_EMPLOYEES
from ExperiencedEmployees;

-- 17. Retrieve the names of departments in ascending order and their employees in descending order.
select D.DNAME, E.ENAME
from DEPT D
JOIN EMP E ON D.DEPTNO = E.DEPTNO
ORDER BY D.DNAME ASC, E.ENAME DESC;
```

The Results pane shows the output of the query, displaying 14 rows of data with columns DNAME and ENAME:

DNAME	ENAME
ACCOUNTING	MILLER
ACCOUNTING	KING
ACCOUNTING	CLARK
RESEARCH	SMITH
RESEARCH	SCOTT
RESEARCH	JONES
RESEARCH	FORD
RESEARCH	ADAMS
SALES	WARD
SALES	TURNER
SALES	MARTIN
SALES	JAMES
SALES	BLAKE
SALES	ALLEN

The status bar at the bottom indicates "Query executed successfully." and "14 rows".

18. Find out experience of MILLER.

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The query editor displays the following SQL code:

```
JOIN EMP E ON D.DEPTNO = E.DEPTNO
ORDER BY D.DNAME ASC, E.ENAME DESC;

-- 18. Find out experience of MILLER.
select ENAME, DATEDIFF(YEAR, HIREDATE, GETDATE()) AS EXPERIENCE_YEARS
from EMP
where ENAME = 'MILLER';
```

The Results pane shows the output of the query, displaying 1 row of data with columns ENAME and EXPERIENCE_YEARS:

ENAME	EXPERIENCE_YEARS
MILLER	42

The status bar at the bottom indicates "Query executed successfully." and "1 rows".