

Name:- Atish Kumar

Roll No:- 120CS0173

Date:- 06 April, 2022

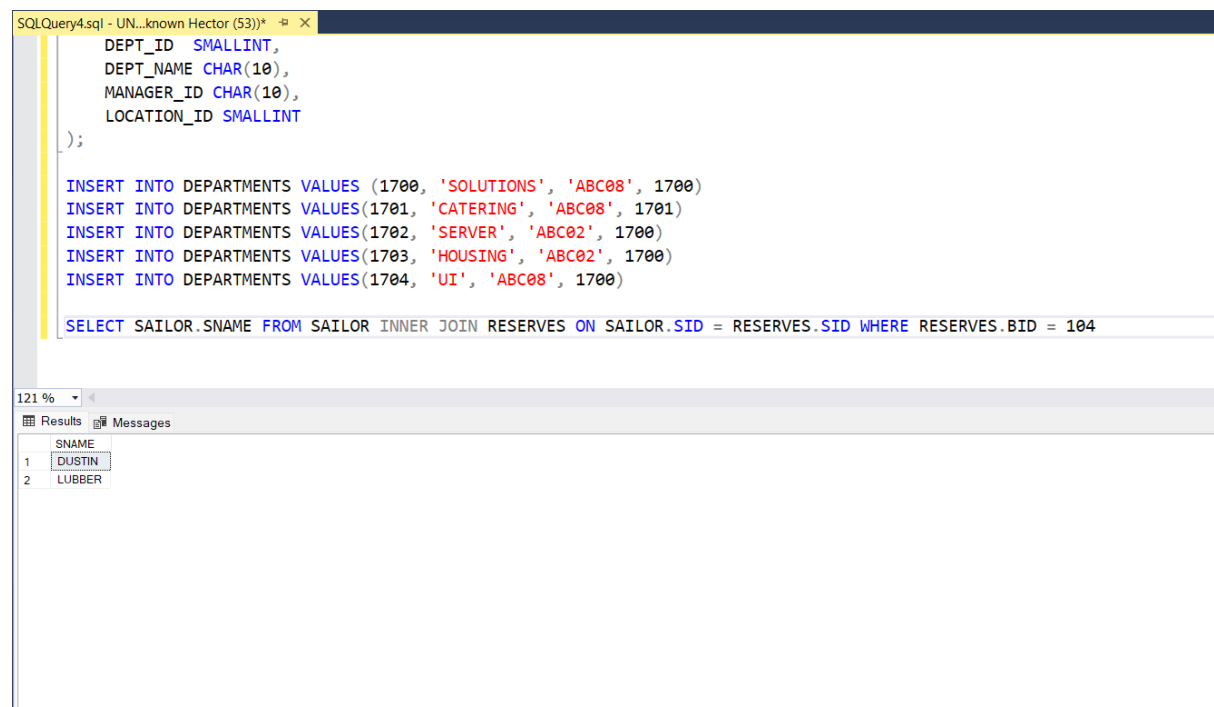
DE Lab Assignment 07

1. Find the names of the sailor who have reserved boat number 104.

Program:-

```
SELECT SAILOR.SNAME FROM SAILOR INNER JOIN RESERVES ON SAILOR.SID = RESERVES.SID WHERE  
RESERVES.BID = 104
```

Output:-



The screenshot shows the SQL Developer interface. The main window displays a SQL script with several INSERT statements for the DEPARTMENTS table and a SELECT statement for the SAILOR table. The SELECT statement is highlighted, and the results are shown in the Results pane below. The Results pane shows two rows: DUSTIN and LUBBER.

```
SQLQuery4.sql - UN...known Hector (53)*  
DEPT_ID SMALLINT,  
DEPT_NAME CHAR(10),  
MANAGER_ID CHAR(10),  
LOCATION_ID SMALLINT  
);  
  
INSERT INTO DEPARTMENTS VALUES (1700, 'SOLUTIONS', 'ABC08', 1700)  
INSERT INTO DEPARTMENTS VALUES(1701, 'CATERING', 'ABC08', 1701)  
INSERT INTO DEPARTMENTS VALUES(1702, 'SERVER', 'ABC02', 1700)  
INSERT INTO DEPARTMENTS VALUES(1703, 'HOUSING', 'ABC02', 1700)  
INSERT INTO DEPARTMENTS VALUES(1704, 'UI', 'ABC08', 1700)  
  
SELECT SAILOR.SNAME FROM SAILOR INNER JOIN RESERVES ON SAILOR.SID = RESERVES.SID WHERE RESERVES.BID = 104
```

Results

SNAME
1 DUSTIN
2 LUBBER

2. Find the names of the sailor who have reserved both green and red boat.

Program:-

```
SELECT SNAME FROM SAILOR WHERE SID IN  
(SELECT RESERVES.SID FROM RESERVES INNER JOIN BOATS ON BOATS.BID = RESERVES.BID WHERE  
BOATS.BCOLOR = 'RED' OR BOATS.BCOLOR = 'GREEN');
```

Output:-

```
SQLQuery4.sql - UN...known Hector (53)*
);

INSERT INTO DEPARTMENTS VALUES (1700, 'SOLUTIONS', 'ABC08', 1700)
INSERT INTO DEPARTMENTS VALUES(1701, 'CATERING', 'ABC08', 1701)
INSERT INTO DEPARTMENTS VALUES(1702, 'SERVER', 'ABC02', 1700)
INSERT INTO DEPARTMENTS VALUES(1703, 'HOUSING', 'ABC02', 1700)
INSERT INTO DEPARTMENTS VALUES(1704, 'UI', 'ABC08', 1700)

SELECT SAILOR.SNAME FROM SAILOR INNER JOIN RESERVES ON SAILOR.SID = RESERVES.SID WHERE RESERVES.BID = 104

SELECT SNAME FROM SAILOR WHERE SID IN
((SELECT RESERVES.SID FROM RESERVES INNER JOIN BOATS ON BOATS.BID = RESERVES.BID WHERE BOATS.BCOLOR = 'RED' OR BOATS.BCOLOR = 'GREEN'));
```

121 %

Results Messages

	SNAME
1	DUSTIN
2	LUBBER
3	HORATIO
4	HORATIO

3. Find the names of the sailors who have reserved all boats.

Program:-

```
SELECT SAILOR.SNAME FROM SAILOR INNER JOIN(SELECT RESERVES.SID FROM RESERVES GROUP BY
RESERVES.SID HAVING COUNT(DISTINCT
CONCAT(RESERVES.SID, RESERVES.BID)) = (SELECT COUNT(DISTINCT BOATS.BID) FROM BOATS))
TEMP ON SAILOR.SID = TEMP.SID;
```

Output:-

```
SQLQuery4.sql - UN...known Hector (53)*
INSERT INTO DEPARTMENTS VALUES(1703, 'HOUSING', 'ABC02', 1700)
INSERT INTO DEPARTMENTS VALUES(1704, 'UI', 'ABC08', 1700)

SELECT SAILOR.SNAME FROM SAILOR INNER JOIN RESERVES ON SAILOR.SID = RESERVES.SID WHERE RESERVES.BID = 104

SELECT SNAME FROM SAILOR WHERE SID IN
((SELECT RESERVES.SID FROM RESERVES INNER JOIN BOATS ON BOATS.BID = RESERVES.BID WHERE BOATS.BCOLOR = 'RED' OR BOATS.BCOLOR = 'GREEN'));
```

```
SELECT SAILOR.SNAME FROM SAILOR INNER JOIN(SELECT RESERVES.SID FROM RESERVES GROUP BY RESERVES.SID HAVING COUNT(DISTINCT
CONCAT(RESERVES.SID, RESERVES.BID)) = (SELECT COUNT(DISTINCT BOATS.BID) FROM BOATS)) TEMP ON SAILOR.SID = TEMP.SID;
```

121 %

Results Messages

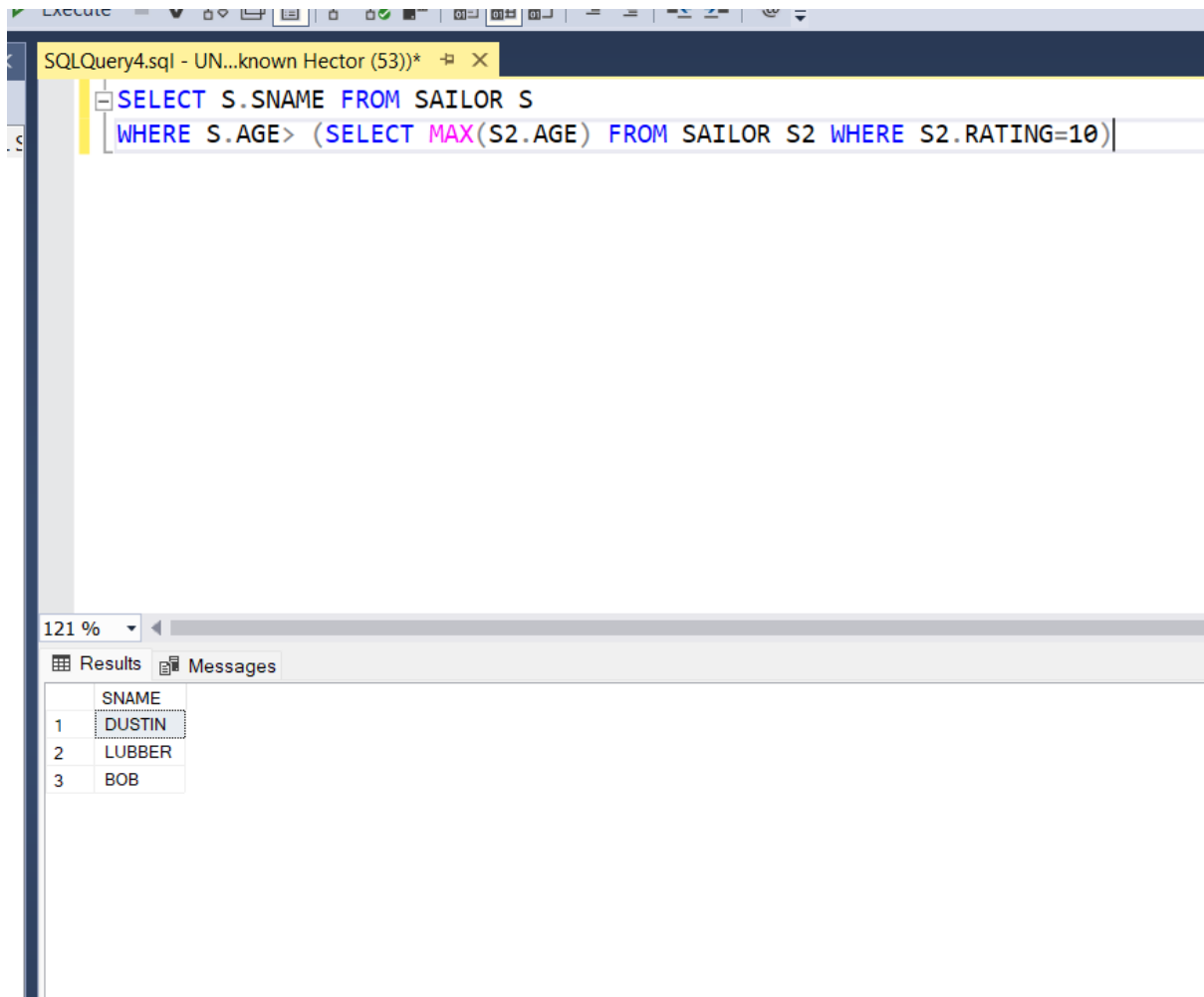
	SNAME
1	DUSTIN

4. Find the names of the sailors who are older than the oldest sailor with a rating of 10.

Program:-

```
SELECT S.SNAME FROM SAILOR S
WHERE S.AGE > (SELECT MAX(S2.AGE) FROM SAILOR S2 WHERE S2.RATING=10)
```

Output:-



The screenshot shows a SQL query execution window. The query is: `SELECT S.SNAME FROM SAILOR S WHERE S.AGE > (SELECT MAX(S2.AGE) FROM SAILOR S2 WHERE S2.RATING=10)`. The results are displayed in a table with the following data:

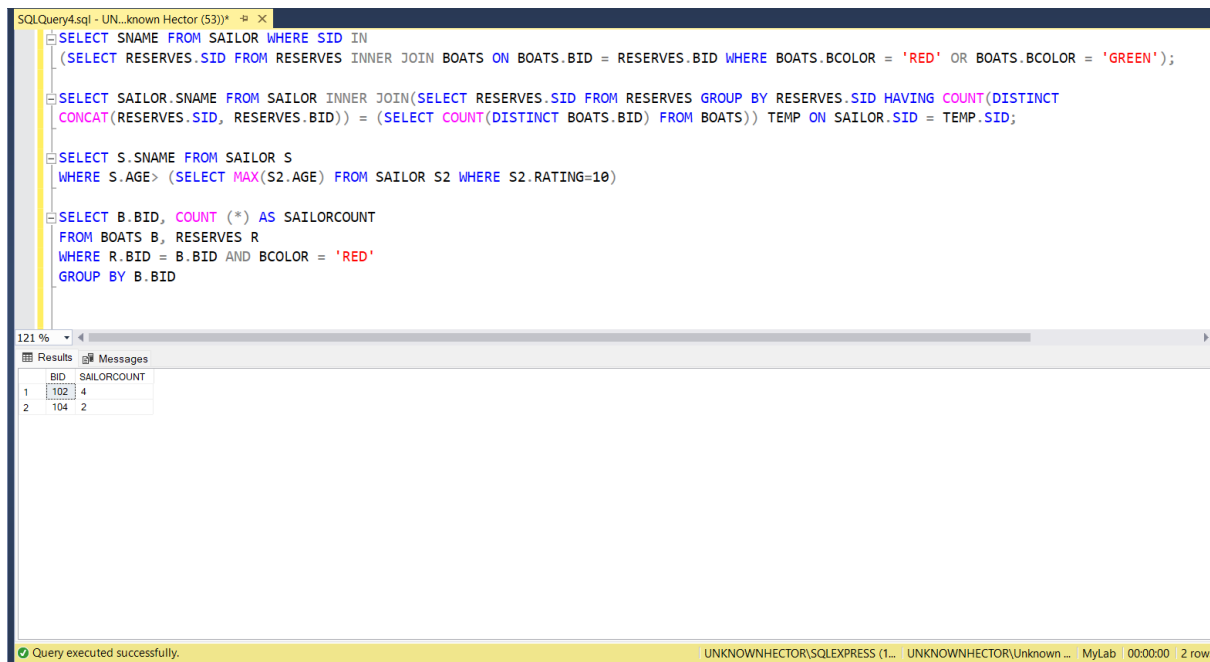
	SNAME
1	DUSTIN
2	LUBBER
3	BOB

5. For each red boat find the number of reservation for this boat.

Program:-

```
SELECT B.BID, COUNT (*) AS SAILORCOUNT
FROM BOATS B, RESERVES R
WHERE R.BID = B.BID AND BCOLOR = 'RED'
GROUP BY B.BID
```

Output:-



The screenshot shows a SQL Query Editor window with the following query:

```
SELECT SNAME FROM SAILOR WHERE SID IN
(SELECT RESERVES.SID FROM RESERVES INNER JOIN BOATS ON BOATS.BID = RESERVES.BID WHERE BOATS.BCOLOR = 'RED' OR BOATS.BCOLOR = 'GREEN');

SELECT SAILOR.SNAME FROM SAILOR INNER JOIN (SELECT RESERVES.SID FROM RESERVES GROUP BY RESERVES.SID HAVING COUNT(DISTINCT
CONCAT(RESERVES.SID, RESERVES.BID)) = (SELECT COUNT(DISTINCT BOATS.BID) FROM BOATS)) TEMP ON SAILOR.SID = TEMP.SID;

SELECT S.SNAME FROM SAILOR S
WHERE S.AGE > (SELECT MAX(S2.AGE) FROM SAILOR S2 WHERE S2.RATING=10)

SELECT B.BID, COUNT (*) AS SAILORCOUNT
FROM BOATS B, RESERVES R
WHERE R.BID = B.BID AND BCOLOR = 'RED'
GROUP BY B.BID
```

The Results tab shows the following output:

BID	SAILORCOUNT
102	4
104	2

The status bar at the bottom indicates: Query executed successfully. UNKNOWNHECTOR\SQLEXPRESS (1... UNKNOWNHECTOR\Unknown... MyLab | 00:00:00 | 2 rows

6. Find the average age of the sailors who are of voting age for each rating level that has at least two such sailors.

Program:-

```
SELECT S.RATING, AVG(S.AGE) AS AVGAGE
FROM SAILOR S
WHERE S.AGE > 18
GROUP BY S.RATING
HAVING COUNT (*) > 1
```

Output:-

SQLQuery4.sql - UN...known Hector (53))*

```

SELECT S.SNAME FROM SAILOR S
WHERE S.AGE > (SELECT MAX(S2.AGE) FROM SAILOR S2 WHERE S2.RATING=10)

SELECT S.RATING, AVG(S.AGE) AS AVGAGE
FROM SAILOR S
WHERE S.AGE > 18
GROUP BY S.RATING
HAVING COUNT(*) > 1

```

121 %

Results Messages

	RATING	AVGAGE
1	3	44
2	7	40
3	8	40

7. Show the employees having same first name .

Program:-

```

SELECT E1.FIRST_NAME FROM EMPLOYES E1, EMPLOYES E2 WHERE E1.FIRST_NAME = E2.FIRST_NAME
AND E1.EMP_ID <> E2.EMP_ID

```

Output:-

SQLQuery4.sql - UN...known Hector (53))*

```

SELECT S.RATING, AVG(S.AGE) AS AVGAGE
FROM SAILOR S
WHERE S.AGE > 18
GROUP BY S.RATING
HAVING COUNT(*) > 1

SELECT E1.FIRST_NAME FROM EMPLOYES E1, EMPLOYES E2 WHERE E1.FIRST_NAME = E2.FIRST_NAME AND E1.EMP_ID <> E2.EMP_ID

```

121 %

Results Messages

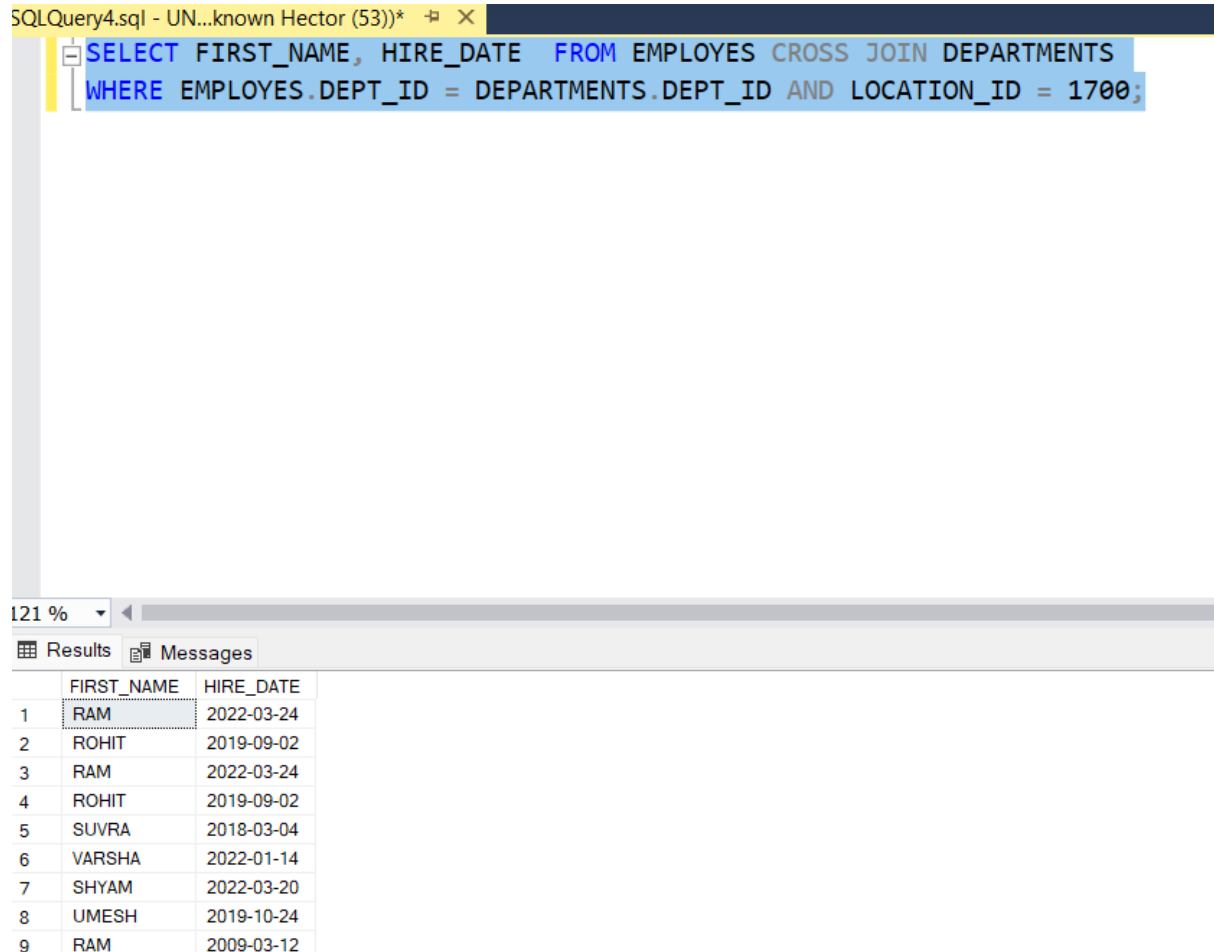
	FIRST_NAME
1	RAM
2	RAM

8. Write a query to output the names and hire dates of all latest hires in their respective departments in the location ID 1700.

Program:-

```
SELECT FIRST_NAME, HIRE_DATE FROM EMPLOYES CROSS JOIN DEPARTMENTS
WHERE EMPLOYES.DEPT_ID = DEPARTMENTS.DEPT_ID AND LOCATION_ID = 1700;
```

Output:-



The screenshot shows the SQL Developer interface. The top pane displays the query: `SELECT FIRST_NAME, HIRE_DATE FROM EMPLOYES CROSS JOIN DEPARTMENTS WHERE EMPLOYES.DEPT_ID = DEPARTMENTS.DEPT_ID AND LOCATION_ID = 1700;`. The bottom pane shows the results of the query in a table with 9 rows and 2 columns: FIRST_NAME and HIRE_DATE. The results are sorted by hire date in descending order.

	FIRST_NAME	HIRE_DATE
1	RAM	2022-03-24
2	ROHIT	2019-09-02
3	RAM	2022-03-24
4	ROHIT	2019-09-02
5	SUVRA	2018-03-04
6	VARSHA	2022-01-14
7	SHYAM	2022-03-20
8	UMESH	2019-10-24
9	RAM	2009-03-12

9. Create a column create_date in department and set it to earliest hire_date for each department.

Program:-

```
ALTER TABLE DEPARTMENTS
ADD CREATE_DATE DATE;

UPDATE DEPARTMENTS
SET DEPARTMENTS.CREATE_DATE = EMPLOYEE.CREATE_DATE
FROM DEPARTMENTS
INNER JOIN
```

```

(SELECT MIN(HIRE_DATE) AS CREATE_DATE,DEPT_ID FROM EMPLOYEES
GROUP BY DEPT_ID) AS EMPLOYEE
ON DEPARTMENTS.DEPT_ID = EMPLOYEE.DEPT_ID;

SELECT *FROM DEPARTMENTS;

```

Output:-

SQLQuery4.sql - UN...known Hector (53)

```

ALTER TABLE DEPARTMENTS
ADD CREATE_DATE DATE;

UPDATE DEPARTMENTS
SET DEPARTMENTS.CREATE_DATE = EMPLOYEE.CREATE_DATE
FROM DEPARTMENTS
INNER JOIN
(SELECT MIN(HIRE_DATE) AS CREATE_DATE,DEPT_ID FROM EMPLOYEES
GROUP BY DEPT_ID) AS EMPLOYEE
ON DEPARTMENTS.DEPT_ID = EMPLOYEE.DEPT_ID;

SELECT *FROM DEPARTMENTS;

```

121 %

Results Messages

	DEPT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID	CREATE_DATE
1	1700	SOLUTIONS	ABC08	1700	NULL
2	1700	SOLUTIONS	ABC08	1700	NULL
3	1701	CATERING	ABC08	1701	NULL
4	1702	SERVER	ABC02	1700	NULL
5	1703	HOUSING	ABC02	1700	NULL
6	1704	UI	ABC08	1700	NULL

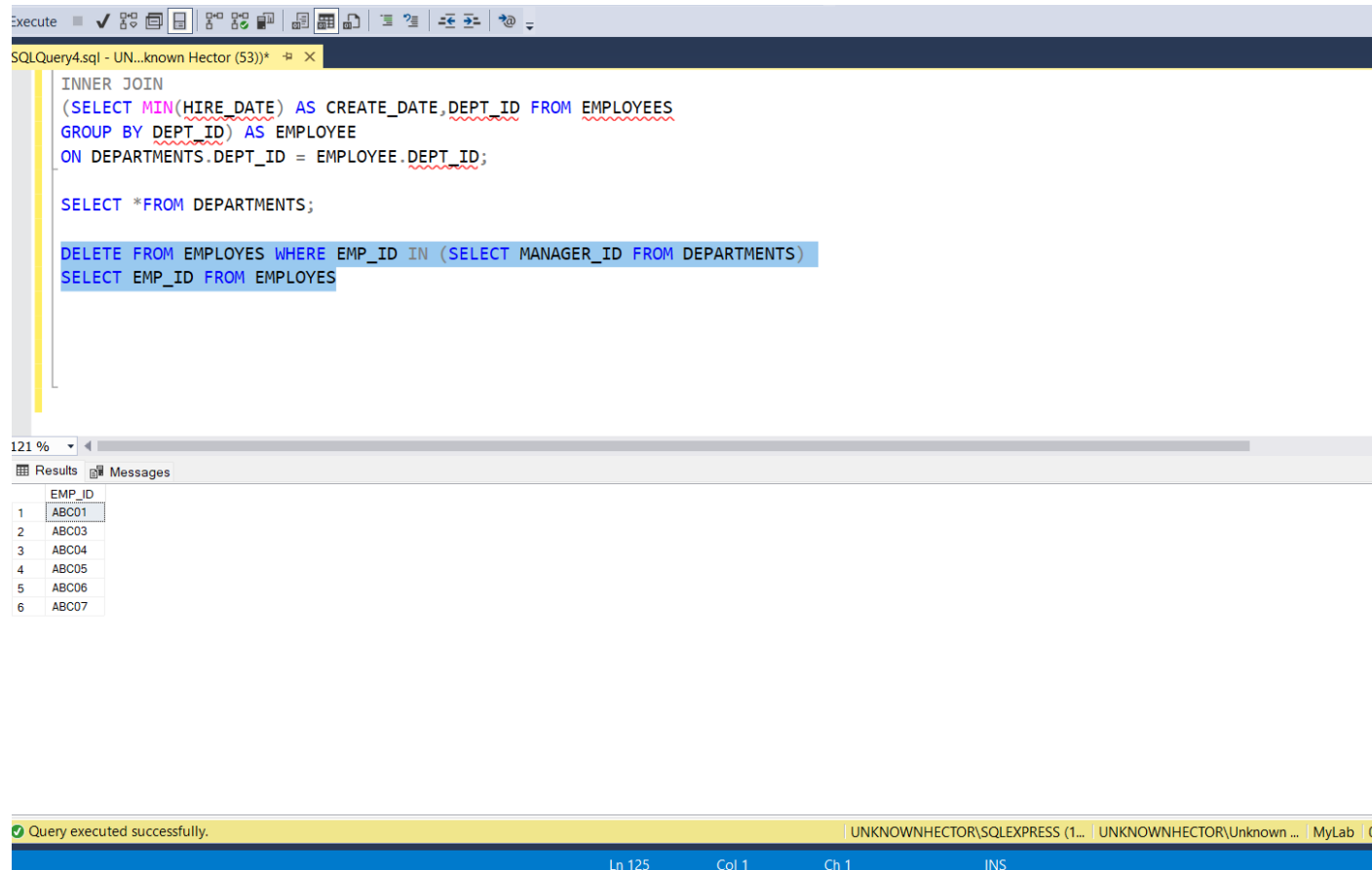
Query executed successfully. UNKNOWNHECTOR\SQLEXPRESS (1... UNKNOWNHECTOR\Unknown ... MyLab 00:00:00 6 rows

10. Delete entry from employee who is his own manager

Program:-

```
DELETE FROM EMPLOYEES WHERE EMP_ID IN (SELECT MANAGER_ID FROM DEPARTMENTS)
SELECT EMP_ID FROM EMPLOYEES
```

Output:-



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL code:

```
INNER JOIN
(SELECT MIN(HIRE_DATE) AS CREATE_DATE,DEPT_ID FROM EMPLOYEES
GROUP BY DEPT_ID) AS EMPLOYEE
ON DEPARTMENTS.DEPT_ID = EMPLOYEE.DEPT_ID;

SELECT *FROM DEPARTMENTS;

DELETE FROM EMPLOYEES WHERE EMP_ID IN (SELECT MANAGER_ID FROM DEPARTMENTS)
SELECT EMP_ID FROM EMPLOYEES
```

The bottom pane shows the results of the query, displaying a table with the following data:

EMP_ID
1 ABC01
2 ABC03
3 ABC04
4 ABC05
5 ABC06
6 ABC07

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

All Program

```
CREATE TABLE SAILOR
(
    SID TINYINT,
    SNAME CHAR(10),
    RATING TINYINT,
    AGE TINYINT
);

INSERT INTO SAILOR VALUES(22, 'DUSTIN', 7, 45);
INSERT INTO SAILOR VALUES(29, 'BRUTUS', 1, 33);
INSERT INTO SAILOR VALUES(31, 'LUBBER', 8, 55);
INSERT INTO SAILOR VALUES(32, 'ANDY', 8, 25);
INSERT INTO SAILOR VALUES(58, 'RUSTY', 10, 35);
INSERT INTO SAILOR VALUES(64, 'HORATIO', 7, 35);
INSERT INTO SAILOR VALUES(71, 'ZORBA', 10, 16);
INSERT INTO SAILOR VALUES(74, 'HORATIO', 9, 35);
INSERT INTO SAILOR VALUES(85, 'ART', 3, 25);
INSERT INTO SAILOR VALUES(95, 'BOB', 3, 63);

CREATE TABLE BOATS
(
    BID TINYINT,
    BNAME CHAR(10),
    BCOLOR CHAR(10)
);

INSERT INTO BOATS VALUES(101, 'INTERLAKE', 'BLUE');
INSERT INTO BOATS VALUES(102, 'INTERLAKE', 'RED');
INSERT INTO BOATS VALUES(103, 'CLIPPER', 'GREEN');
INSERT INTO BOATS VALUES(104, 'MARINE', 'RED');

CREATE TABLE RESERVES
(
    SID TINYINT,
    BID TINYINT,
    DATE DATE
);

INSERT INTO RESERVES VALUES(22, 101, '1998-10-10')
INSERT INTO RESERVES VALUES(22, 102, '1998-10-10')
INSERT INTO RESERVES VALUES(22, 103, '1998-08-10')
INSERT INTO RESERVES VALUES(22, 104, '1998-07-10')
INSERT INTO RESERVES VALUES(31, 102, '1998-10-11')
INSERT INTO RESERVES VALUES(31, 103, '1998-06-11')
INSERT INTO RESERVES VALUES(31, 104, '1998-12-11')
INSERT INTO RESERVES VALUES(64, 101, '1998-05-09')
INSERT INTO RESERVES VALUES(64, 102, '1998-08-09')
INSERT INTO RESERVES VALUES(74, 103, '1998-08-09')

CREATE TABLE EMPLOYES
(
    EMP_ID CHAR(10),
    FIRST_NAME CHAR(10),
    LAST_NAME CHAR(10),
    HIRE_DATE DATE,
    DEPT_ID SMALLINT,
    SALARY INT
);
```

```

INSERT INTO EMPLOYES VALUES('ABC01', 'RAM', 'KUMAR', '2022-03-24', 1700, 60000)
INSERT INTO EMPLOYES VALUES('ABC02', 'RAMAN', 'SHARMA', '2021-08-08', 1701, 70000)
INSERT INTO EMPLOYES VALUES('ABC03', 'SUVRU', 'SARKAR', '2018-03-04', 1702, 65000)
INSERT INTO EMPLOYES VALUES('ABC04', 'SHYAM', 'KUMAR', '2022-03-20', 1703, 60000)
INSERT INTO EMPLOYES VALUES('ABC05', 'ROHIT', 'VERMA', '2019-09-02', 1700, 80000)
INSERT INTO EMPLOYES VALUES('ABC06', 'VARSHA', 'KUMARI', '2022-01-14', 1702, 60000)
INSERT INTO EMPLOYES VALUES('ABC07', 'UMESH', 'SINHA', '2019-10-24', 1703, 70000)
INSERT INTO EMPLOYES VALUES('ABC08', 'RAM', 'SAHOO', '2009-03-12', 1704, 50000)

```

```

CREATE TABLE DEPARTMENTS
(
    DEPT_ID SMALLINT,
    DEPT_NAME CHAR(10),
    MANAGER_ID CHAR(10),
    LOCATION_ID SMALLINT
);

```

```

INSERT INTO DEPARTMENTS VALUES (1700, 'SOLUTIONS', 'ABC08', 1700)
INSERT INTO DEPARTMENTS VALUES(1701, 'CATERING', 'ABC08', 1701)
INSERT INTO DEPARTMENTS VALUES(1702, 'SERVER', 'ABC02', 1700)
INSERT INTO DEPARTMENTS VALUES(1703, 'HOUSING', 'ABC02', 1700)
INSERT INTO DEPARTMENTS VALUES(1704, 'UI', 'ABC08', 1700)

```

```

SELECT SAILOR.SNAME FROM SAILOR INNER JOIN RESERVES ON SAILOR.SID = RESERVES.SID WHERE
RESERVES.BID = 104

```

```

SELECT SNAME FROM SAILOR WHERE SID IN
(SELECT RESERVES.SID FROM RESERVES INNER JOIN BOATS ON BOATS.BID = RESERVES.BID WHERE
BOATS.BCOLOR = 'RED' OR BOATS.BCOLOR = 'GREEN');

```

```

SELECT SAILOR.SNAME FROM SAILOR INNER JOIN (SELECT RESERVES.SID FROM RESERVES GROUP BY
RESERVES.SID HAVING COUNT(DISTINCT
CONCAT(RESERVES.SID, RESERVES.BID)) = (SELECT COUNT(DISTINCT BOATS.BID) FROM BOATS))
TEMP ON SAILOR.SID = TEMP.SID;

```

```

SELECT S.SNAME FROM SAILOR S
WHERE S.AGE > (SELECT MAX(S2.AGE) FROM SAILOR S2 WHERE S2.RATING=10)

```

```

SELECT B.BID, COUNT (*) AS SAILORCOUNT
FROM BOATS B, RESERVES R
WHERE R.BID = B.BID AND BCOLOR = 'RED'
GROUP BY B.BID

```

```

SELECT S.RATING, AVG(S.AGE) AS AVGAGE
FROM SAILOR S
WHERE S.AGE > 18
GROUP BY S.RATING
HAVING COUNT(*) > 1

```

```

SELECT E1.FIRST_NAME FROM EMPLOYES E1, EMPLOYES E2 WHERE E1.FIRST_NAME = E2.FIRST_NAME
AND E1.EMP_ID <> E2.EMP_ID

```

```

SELECT FIRST_NAME, HIRE_DATE FROM EMPLOYES CROSS JOIN DEPARTMENTS
WHERE EMPLOYES.DEPT_ID = DEPARTMENTS.DEPT_ID AND LOCATION_ID = 1700;

```

```

ALTER TABLE DEPARTMENTS
ADD CREATE_DATE DATE;

```

```

UPDATE DEPARTMENTS
SET DEPARTMENTS.CREATE_DATE = EMPLOYEE.CREATE_DATE
FROM DEPARTMENTS

```

INNER JOIN

```
(SELECT MIN(HIRE_DATE) AS CREATE_DATE,DEPT_ID FROM EMPLOYEES  
GROUP BY DEPT_ID) AS EMPLOYEE  
ON DEPARTMENTS.DEPT_ID = EMPLOYEE.DEPT_ID;
```

```
SELECT *FROM DEPARTMENTS;
```

```
DELETE FROM EMPLOYEES WHERE EMP_ID IN (SELECT MANAGER_ID FROM DEPARTMENTS)  
SELECT EMP_ID FROM EMPLOYEES
```