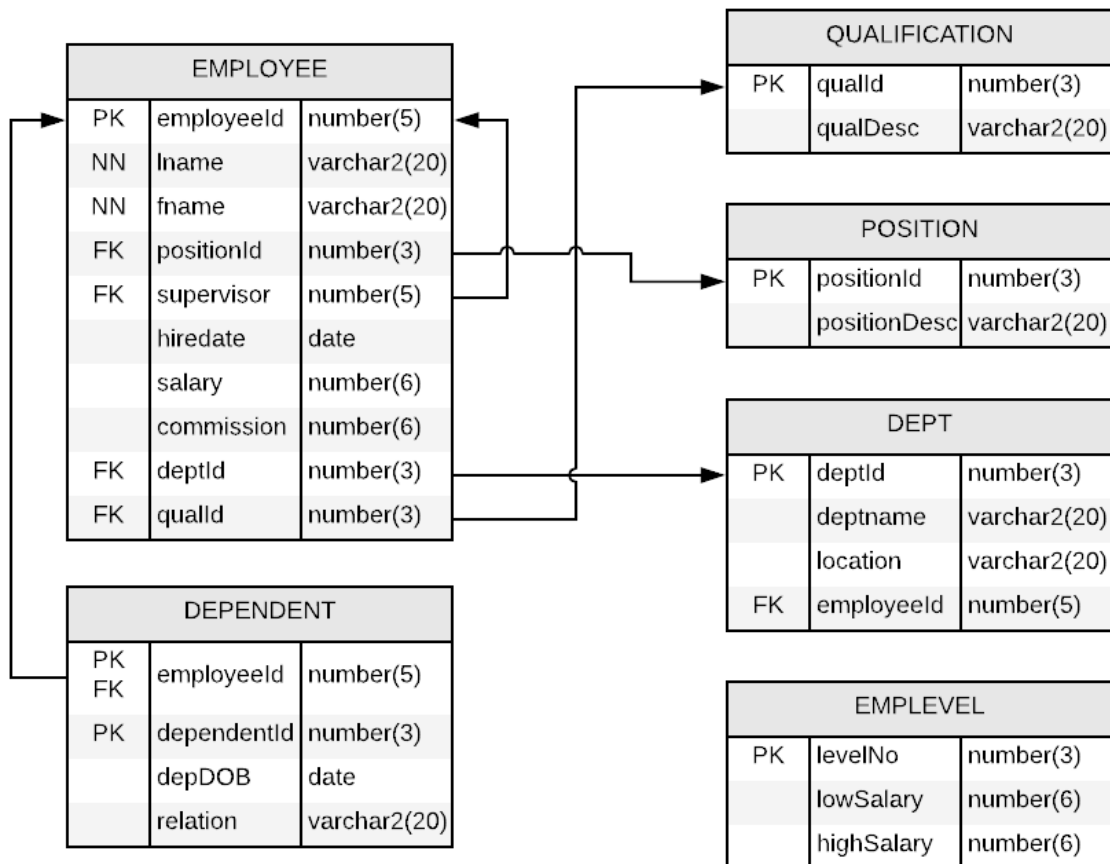


Muhammed Emin Ay

20120205038

1-)Hafta7 LAB5.pptx dosyasının 15. sayfasında verilen veri modeline göre veri tabanındaki tabloları oluşturunuz.



```
CREATE TABLE QUALIFICATION (  
    qualid NUMBER(3) PRIMARY KEY NOT NULL,  
    qualDesc VARCHAR2(20) NOT NULL  
);
```

```
CREATE TABLE POSITION (  
    positionid NUMBER(3) PRIMARY KEY NOT NULL,  
    positionDesc VARCHAR2(20),  
);
```

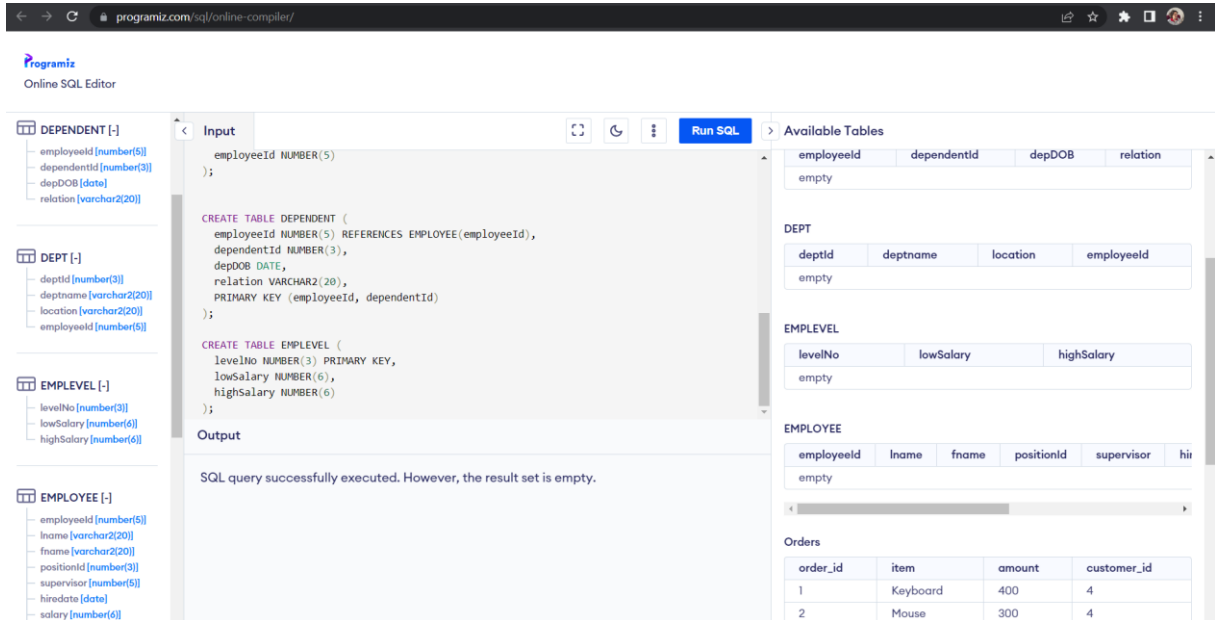
```
CREATE TABLE DEPT (  
    deptid NUMBER(3) PRIMARY KEY NOT NULL,  
    deptname VARCHAR2(20),  
    location VARCHAR2(20)  
);
```

```
CREATE TABLE EMPLOYEE (  
    employeeid NUMBER(5) PRIMARY KEY NOT NULL,  
    lname VARCHAR2(20) NOT NULL,  
    fname VARCHAR2(20) NOT NULL,  
    positionid NUMBER(3) REFERENCES POSITION(positionid),  
    supervisor NUMBER(5) REFERENCES EMPLOYEE(employeeid),  
    hiredate DATE,  
    salary NUMBER(6),  
    commission NUMBER(6),  
    deptid NUMBER(3) REFERENCES DEPT(deptid),  
    qualid NUMBER(3) REFERENCES QUALIFICATION(qualid)  
);
```

```
CREATE TABLE EMPLEVEL (  
    levelNo NUMBER(3) PRIMARY KEY NOT NULL,  
    employeeid NUMBER(5) REFERENCES EMPLOYEE(employeeid)  
);
```

```
CREATE TABLE DEPENDENT (  
    dependentid NUMBER(3) PRIMARY KEY NOT NULL,  
    employeeid NUMBER(5) REFERENCES EMPLOYEE(employeeid),  
    levelNo NUMBER(3) REFERENCES EMPLEVEL(levelNo),
```

depDOB DATE,
lowSalary NUMBER(6),
relation VARCHAR2(20),
highSalary NUMBER(6)
);



2-)16. sayfadaki örnek kayıtları bu tablolara giriniz.

EMPLOYEE									
EMPLOYEEID	LNAM	FNAME	POSITIONID	SUPERVISOR	HIREDATE	SALARY	COMMISSION	DEPTID	QUALID
111	Smith	John	1	(null)	17-04-1995	265000	35000	10	1
543	Dev	Derek	2	111	15-03-2010	80000	20000	20	1
246	Houston	Larry	2	111	19-05-1997	150000	10000	40	2
123	Roberts	Sandi	2	111	01-12-2006	75000	(null)	10	2
433	McCall	Alex	3	543	10-05-2012	66500	(null)	20	4
135	Garner	Stanley	2	111	29-06-2011	45000	5000	30	5
200	Shaw	Jinku	5	135	02-01-2015	24500	3000	30	(null)
222	Chen	Sunny	4	123	15-08-2014	35000	(null)	10	3

DEPT			
DEPTID	DEPTNAME	LOCATION	EMPLOYEEID
10	Finance	Charlotte	123
20	InfoSys	New York	543
30	Sales	Woodbridge	135
40	Marketing	Los Angeles	246

POSITION	
POSITIONID	POSITIONDESC
1	President
2	Manager
3	Programmer
4	Accountant
5	Salesman

QUALIFICATION	
QUALID	QUALDESC
1	Doctorate
2	Masters
3	Bachelors
4	Associates
5	High School

DEPENDENT			
EMPLOYEEID	DEPENDENTID	DEPDOB	RELATION
543		128-09-1958	Spouse
543		214-10-1988	Son
200		110-06-1976	Spouse
222		104-02-1975	Spouse

EMPLEVEL		
LEVELNO	LOWSALARY	HIGHSALARY
1	1	25000
2	25001	50000
3	50001	100000
4	100001	500000

```
INSERT INTO QUALIFICATION VALUES (1, 'Doctorate');
INSERT INTO QUALIFICATION VALUES (2, 'Masters');
INSERT INTO QUALIFICATION VALUES (3, 'Bachelors');
INSERT INTO QUALIFICATION VALUES (4, 'Associates');
INSERT INTO QUALIFICATION VALUES (5, 'High School');
```

```
INSERT INTO POSITION VALUES (1, 'President');
INSERT INTO POSITION VALUES (2, 'Manager');
INSERT INTO POSITION VALUES (3, 'Programmer');
INSERT INTO POSITION VALUES (4, 'Accountant');
INSERT INTO POSITION VALUES (5, 'Salesman');
```

```
INSERT INTO EMPLEVEL VALUES (1, 1, 25000);
INSERT INTO EMPLEVEL VALUES (2, 25001, 50000);
INSERT INTO EMPLEVEL VALUES (3, 50001, 100000);
INSERT INTO EMPLEVEL VALUES (4, 100001, 500000);
```

```
INSERT INTO DEPT (deptId, deptname, location) VALUES (10, 'Finance', 'Charlotte');
INSERT INTO DEPT (deptId, deptname, location) VALUES (20, 'InfoSys', 'New York');
INSERT INTO DEPT (deptId, deptname, location) VALUES (30, 'Sales', 'Woodbridge');
INSERT INTO DEPT (deptId, deptname, location) VALUES (40, 'Marketing', 'Los Angeles');
```

```
INSERT INTO EMPLOYEE VALUES (111, 'Smith', 'John', 1, NULL, '04/17/1995', 265000, 35000, 10, 1);
INSERT INTO EMPLOYEE VALUES (543, 'Dev', 'Derek', 2, 111, '03/15/2010', 80000, 20000, 20, 1);
INSERT INTO EMPLOYEE VALUES (246, 'Houston', 'Larry', 2, 111, '05/19/1997', 150000, 10000, 40, 2);
INSERT INTO EMPLOYEE VALUES (123, 'Roberts', 'Sandi', 2, 111, '12/01/2006', 75000, NULL, 10, 2);
```

INSERT INTO EMPLOYEE VALUES (433, 'McCall', 'Alex', 3, 543, '05/10/2012', 66500, NULL, 20, 4);

INSERT INTO EMPLOYEE VALUES (135, 'Garner', 'Stanley', 2, 111, '06/29/2011', 45000, 5000, 30, 5);

INSERT INTO EMPLOYEE VALUES (200, 'Shaw', 'Jinku', 5, 135, '01/02/2015', 24500, 3000, 30, NULL);

INSERT INTO EMPLOYEE VALUES (222, 'Chen', 'Sunny', 4, 123, '08/15/2014', 35000, NULL, 10, 3);

INSERT INTO DEPENDENT VALUES (543, 1, '09/28/1958', 'Spouse');

INSERT INTO DEPENDENT VALUES (543, 2, '10/14/1988', 'Son');

INSERT INTO DEPENDENT VALUES (200, 1, '06/10/1976', 'Spouse');

INSERT INTO DEPENDENT VALUES (222, 1, '02/04/1975', 'Spouse');

programiz.com/sql/online-compiler/

Programiz
Online SQL Editor

DEPENDENT [-]
- employeeid [number(5)]
- dependantid [number(3)]
- depDOB [date]
- relation [varchar(20)]

DEPT [-]
- deptid [number(3)]
- deptname [varchar(20)]
- location [varchar(20)]
- employeeid [number(5)]

EMPLEVEL [-]
- levelNo [number(3)]
- lowSalary [number(6)]
- highSalary [number(6)]

EMPLOYEE [-]
- employeeid [number(5)]
- lname [varchar(20)]
- fname [varchar(20)]
- positionid [number(3)]
- supervisor [number(5)]
- hiredate [date]
- salary [number(6)]

Input

```
INSERT INTO EMPLOYEE VALUES (123, 'Roberts', 'Sandi', 2, 111, '12/01/2006', 75000, NULL, 10, 2);  
INSERT INTO EMPLOYEE VALUES (433, 'McCall', 'Alex', 3, 543, '05/10/2012', 66500, NULL, 20, 4);  
INSERT INTO EMPLOYEE VALUES (135, 'Garner', 'Stanley', 2, 111, '06/29/2011', 45000, 5000, 30, 5);  
INSERT INTO EMPLOYEE VALUES (200, 'Shaw', 'Jinku', 5, 135, '01/02/2015', 24500, 3000, 30, NULL);  
INSERT INTO EMPLOYEE VALUES (222, 'Chen', 'Sunny', 4, 123, '08/15/2014', 35000, NULL, 10, 3);  
  
INSERT INTO DEPENDENT VALUES (543, 1, '09/28/1958', 'Spouse');  
INSERT INTO DEPENDENT VALUES (543, 2, '10/14/1988', 'Son');  
INSERT INTO DEPENDENT VALUES (200, 1, '06/10/1976', 'Spouse');  
INSERT INTO DEPENDENT VALUES (222, 1, '02/04/1975', 'Spouse');
```

Run SQL

Available Tables

DEPT

deptid	deptname	location	employeeid
10	Finance	Charlotte	
20	InfoSys	New York	
30	Sales	Woodbridge	
40	Marketing	Los Angeles	

EMPLEVEL

levelNo	lowSalary	highSalary
1	1	25000
2	25001	50000
3	50001	100000
4	100001	500000

EMPLOYEE

employeeid	lname	fname	positionid	supervisor
111	Smith	John	1	
543	Dev	Derek	2	111
246	Houston	Larry	2	111
123	Roberts	Sandi	2	111

Output

SQL query successfully executed. However, the result set is empty.

SORGULAR

1-) Display employee Jinku Shaw's department name.

```
SELECT d.deptname
```

```
FROM employee e
```

```
JOIN dept d ON e.deptId = d.deptId
```

```
WHERE e.fname = 'Jinku' AND e.lname = 'Shaw';
```

The screenshot shows the Programiz Online SQL Editor interface. On the left, there is a sidebar with a list of available tables: DEPENDENT, DEPT, EMPLEVEL, and EMPLOYEE. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains the SQL query:

```
SELECT d.deptname
FROM employee e
JOIN dept d ON e.deptId = d.deptId
WHERE e.fname = 'Jinku' AND e.lname = 'Shaw';
```

 The 'Output' section displays the result:

deptname
Sales

. On the right side, there are preview tables for DEPT, EMPLEVEL, and EMPLOYEE. The DEPT table has columns: deptid, deptname, location, employeeid. The EMPLEVEL table has columns: levelNo, lowSalary, highSalary. The EMPLOYEE table has columns: employeeid, lname, fname, positionid, supervisor.

2-) Find name of the supervisor for employee number 433.

```
SELECT e1.fname, e1.lname FROM employee e1
```

```
INNER JOIN employee e2 ON e1.employeeId = e2.supervisor WHERE e2.employeeId=433;
```

The screenshot shows the Programiz Online SQL Editor interface. On the left, there is a sidebar with a list of available tables: DEPENDENT, DEPT, EMPLEVEL, and EMPLOYEE. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains the SQL query:

```
SELECT e1.fname, e1.lname FROM employee e1
INNER JOIN employee e2 ON e1.employeeId = e2.supervisor WHERE e2.employeeId=433;
```

 The 'Output' section displays the result:

fname	lname
Derek	Dev

.

3-)Who has same qualification as Stanley Garner?

SELECT fname,lname from employee where qualId in(select qualId from employee where fname='Stanley');

The screenshot shows a SQL IDE interface. On the left, there is a schema explorer with tables: DEPENDENT, DEPT, EMPLEVEL, and EMPLOYEE. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains the SQL query: `SELECT fname,lname from employee where qualId in(select qualId from employee where fname='Stanley');`. The 'Output' section displays the results in a table with two columns: 'fname' and 'lname'. The results are: Stanley and Garner.

fname	lname
Stanley	Garner

4-)Which department has more employees than department 20?

SELECT deptId, COUNT(*) AS Employees FROM EMPLOYEE GROUP BY deptId
HAVING COUNT(*) > (SELECT COUNT(*) FROM EMPLOYEE GROUP BY deptId HAVING deptId = 20);

The screenshot shows a SQL IDE interface. On the left, there is a schema explorer with tables: DEPENDENT, DEPT, EMPLEVEL, and EMPLOYEE. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains the SQL query: `SELECT deptId, COUNT(*) AS Employees FROM EMPLOYEE GROUP BY deptId HAVING COUNT(*) > (SELECT COUNT(*) FROM EMPLOYEE GROUP BY deptId HAVING deptId = 20);`. The 'Output' section displays the results in a table with two columns: 'deptId' and 'Employees'. The results are: 10 and 3.

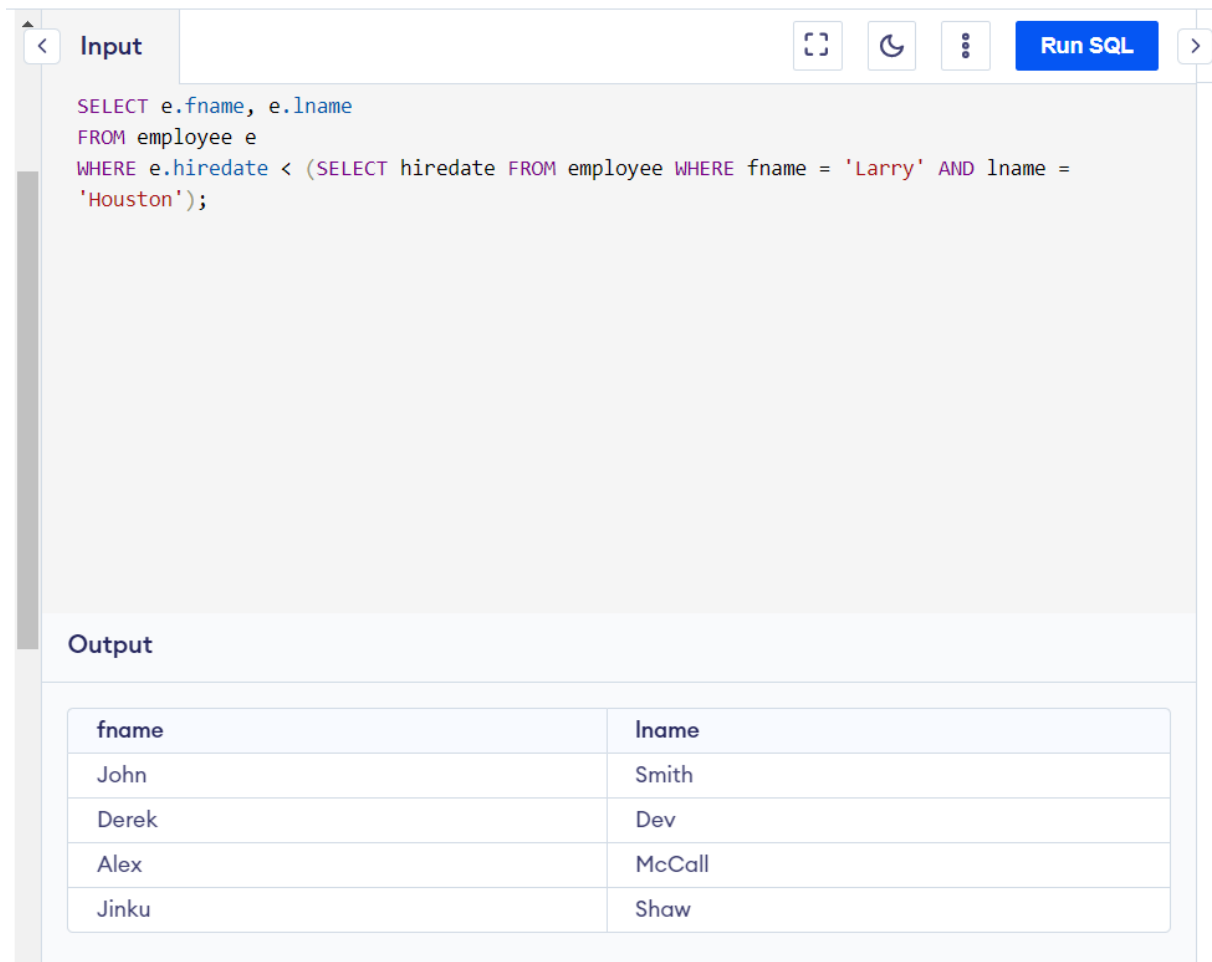
deptId	Employees
10	3

5-)Which employees are working in the company longer than Larry Houston?

```
SELECT e.fname, e.lname
```

```
FROM employee e
```

```
WHERE e.hiredate < (SELECT hiredate FROM employee WHERE fname = 'Larry' AND lname = 'Houston');
```



The screenshot shows a SQL query editor interface. At the top, there is a tab labeled 'Input' and a 'Run SQL' button. The query text is as follows:

```
SELECT e.fname, e.lname
FROM employee e
WHERE e.hiredate < (SELECT hiredate FROM employee WHERE fname = 'Larry' AND lname = 'Houston');
```

Below the query editor, there is an 'Output' section displaying the results in a table format:

fname	lname
John	Smith
Derek	Dev
Alex	McCall
Jinku	Shaw

6-)Find all employees in the sales department by using a nested query.

```
SELECT e.fname, e.lname
```

```
FROM employee e
```

```
WHERE e.deptId = (SELECT deptId FROM dept WHERE deptname = 'Sales');
```


<

Input

⌵

🌙

⋮

Run SQL

>

```

SELECT e.fname, e.lname
FROM employee e
WHERE e.deptId = (SELECT deptId FROM dept WHERE deptname = 'Sales');

```

Output

fname	lname
Stanley	Garner
Jinku	Shaw

7-)Create a new table, EMP30, and populate it with employees in department 30, using an existing table and a subquery. Use EmployeeId, Lname, Fname, HireDate and Salary columns.

```
CREATE TABLE EMP30 AS
```

```
SELECT EmployeeId, Lname, Fname, HireDate, Salary
```

```
FROM employee
```

```
WHERE deptId = 30;
```

Online SQL Editor

EMP30 [-]

- employeeId [num]
- lname [text]
- fname [text]
- hiredate [num]
- salary [num]

<

Input

⌵

🌙

⋮

Run SQL

>

```

CREATE TABLE EMP30 AS
SELECT EmployeeId, Lname, Fname, HireDate, Salary
FROM employee
WHERE deptId = 30;

```

8-)Add more rows to EMP30 table with employee in department 40. Do not transfer employee's salary.

INSERT INTO EMP30 (employeeId, fname, lname, hiredate)

SELECT employeeId, fname, lname, hiredate FROM EMPLOYEE WHERE deptId = 40;

The screenshot shows a SQL IDE interface. On the left, the 'Input' tab contains the following SQL code:

```
INSERT INTO EMP30 (employeeId, fname, lname, hiredate)
SELECT employeeId, fname, lname, hiredate FROM EMPLOYEE WHERE deptId = 40;
```

On the right, the 'Available Tables' tab shows the 'EMP30' table with the following data:

employeeId	lname	fname	hiredate	salary
135	Garner	Stanley	06/29/2011	45000
200	Shaw	Jinku	01/02/2015	24500
246	Houston	Larry	05/19/1997	

9-)Use multiple level subquery to display dependent information for employees, who belong to FINANCE department.

SELECT * FROM DEPENDENT

WHERE employeeId IN

(SELECT employeeId FROM EMPLOYEE WHERE deptId

IN (SELECT deptId FROM DEPT WHERE deptname like 'Finance'));

The screenshot shows a SQL IDE interface. On the left, the 'Input' tab contains the following SQL code:

```
SELECT * FROM DEPENDENT
WHERE employeeId IN
(SELECT employeeId FROM EMPLOYEE WHERE deptId
IN (SELECT deptId FROM DEPT WHERE deptname like 'Finance'));
```

On the right, the 'Output' tab shows the result of the query as a table:

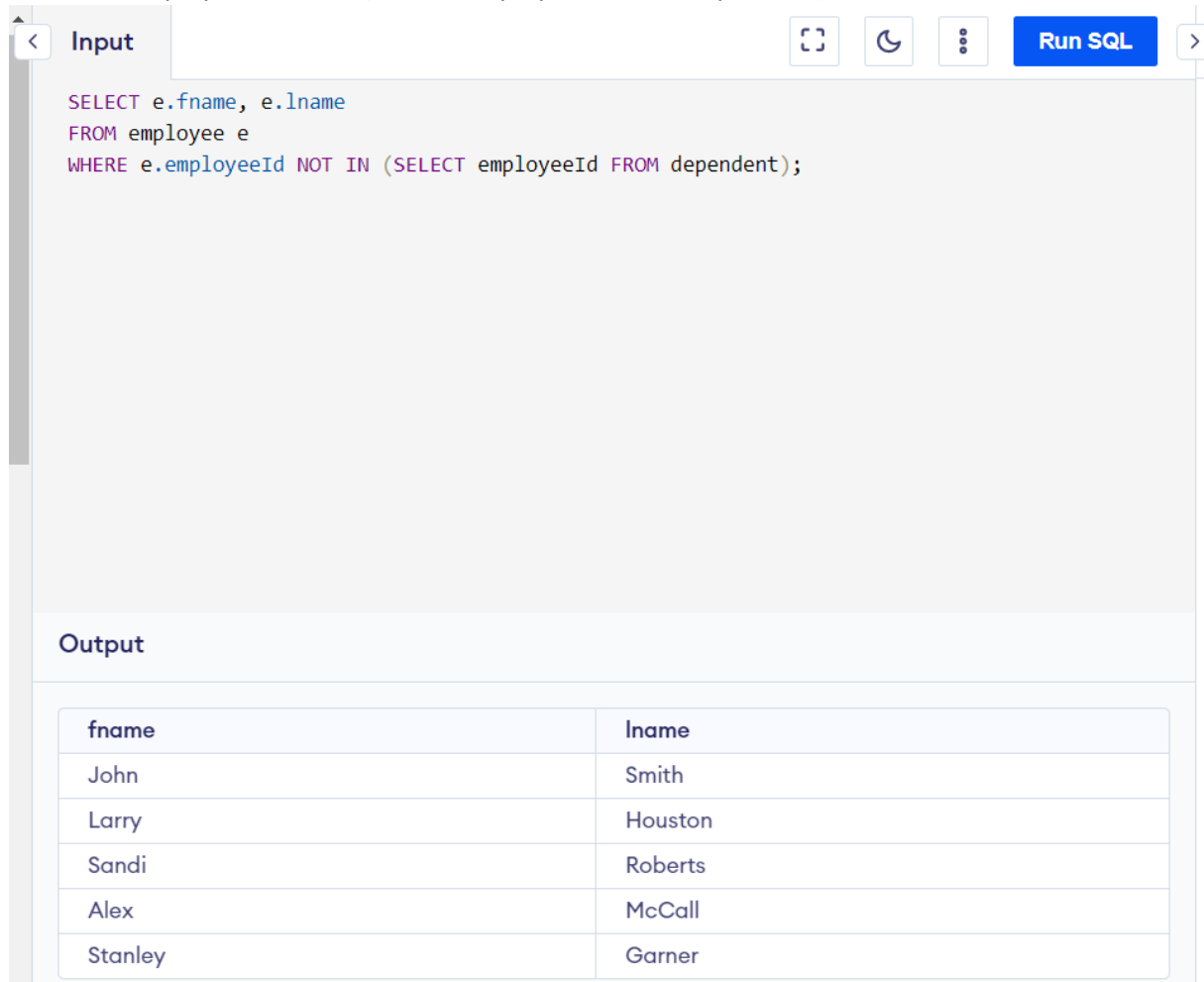
employeeId	dependentId	depDOB	relation
222	1	02/04/1975	Spouse

10-)Use set operator and subquery to find employees, who do not have any dependents.

```
SELECT e.fname, e.lname
```

```
FROM employee e
```

```
WHERE e.employeeId NOT IN (SELECT employeeId FROM dependent);
```



The screenshot shows a SQL query editor interface. At the top, there is a toolbar with icons for undo, redo, and a 'Run SQL' button. Below the toolbar, the query is entered in a text area. The query is: `SELECT e.fname, e.lname FROM employee e WHERE e.employeeId NOT IN (SELECT employeeId FROM dependent);`. Below the query area, there is an 'Output' section which displays a table with two columns: 'fname' and 'lname'. The table contains six rows of data.

fname	lname
John	Smith
Larry	Houston
Sandi	Roberts
Alex	McCall
Stanley	Garner

11-)Write a subquery that finds average salary by each department. Check to find if employee 543's salary satisfies =ANY, <ANY, >ANY, <ALL, or >ALL condition against those departmental average salaries.