Exercise session: Into to SQL

Sergey Paramonov

KU Leuven

9 March 2015

SQL: A

Question: Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project

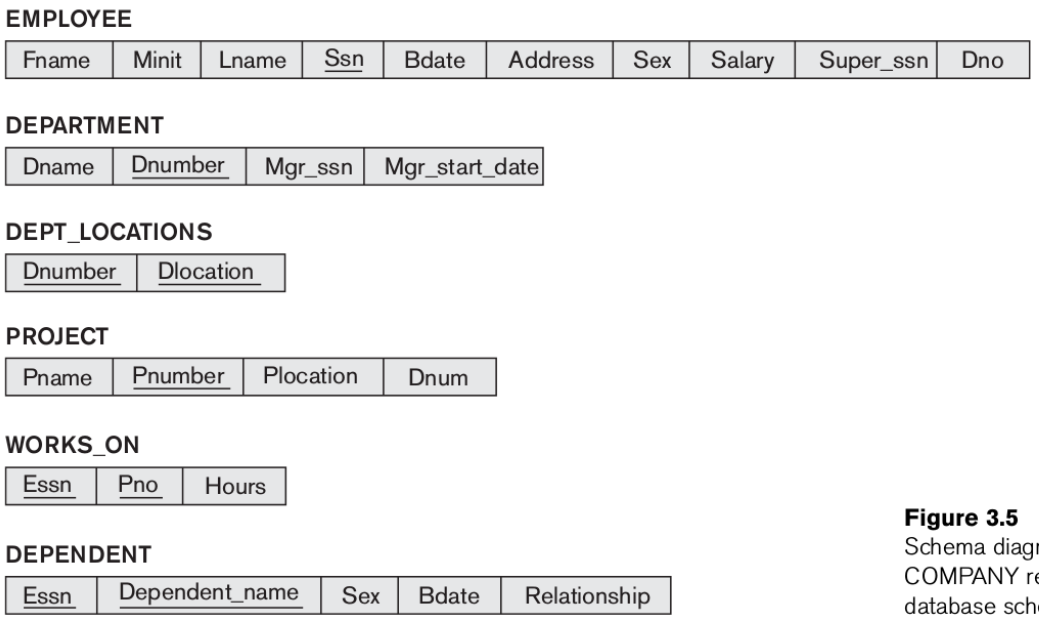

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: A

Question: Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project

```
SELECT Fname, Lname
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE Ssn=Essn
AND Pno=Pnumber
AND Dno=5
AND Hours>=10
AND Pname='ProductX'
```

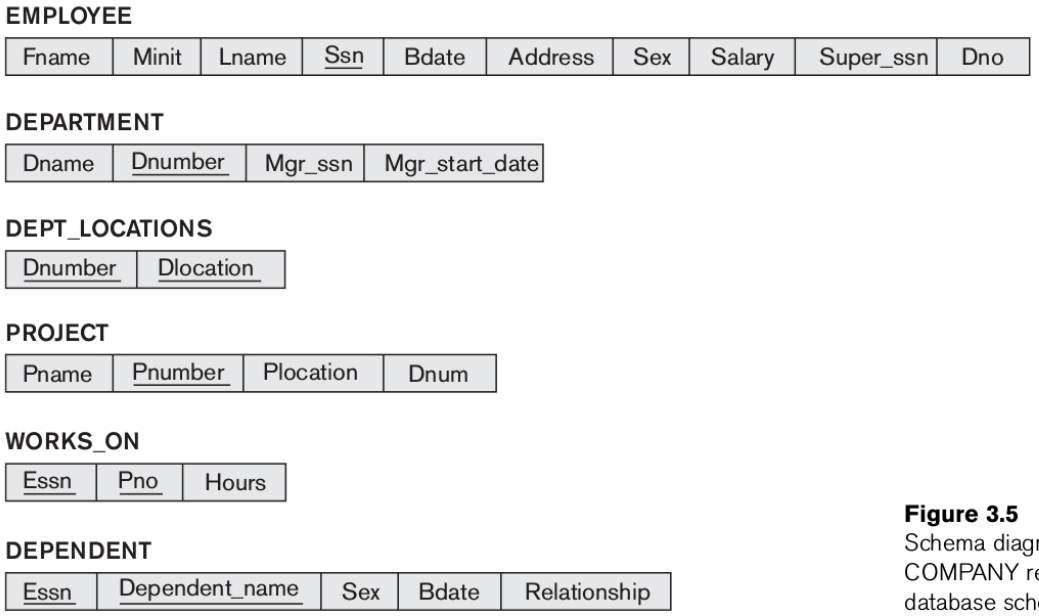

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: A. Alternative Solution

Question: Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project

```
SELECT Fname, Lname
FROM (EMPLOYEE JOIN WORKS_ON ON Ssn=Essn)
             JOIN PROJECT  ON Pno=Pnumber
WHERE Dno=5
AND Hours>=10
AND Pname='ProductX'
```

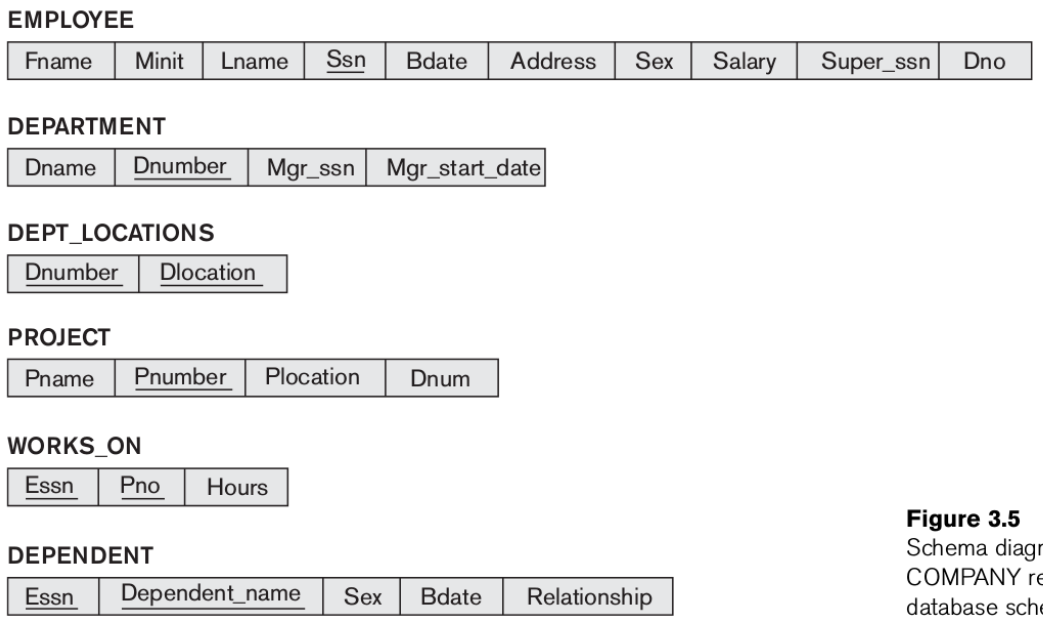

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: A. *Very* Alternative Solution

Question: Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project

```
SELECT Fname, Lname
FROM EMPLOYEE JOIN WORKS_ON ON Ssn=Essn
WHERE Hours>=10
AND Pno IN
(SELECT Pnumber
FROM PROJECT
WHERE Pname='ProductX')
```

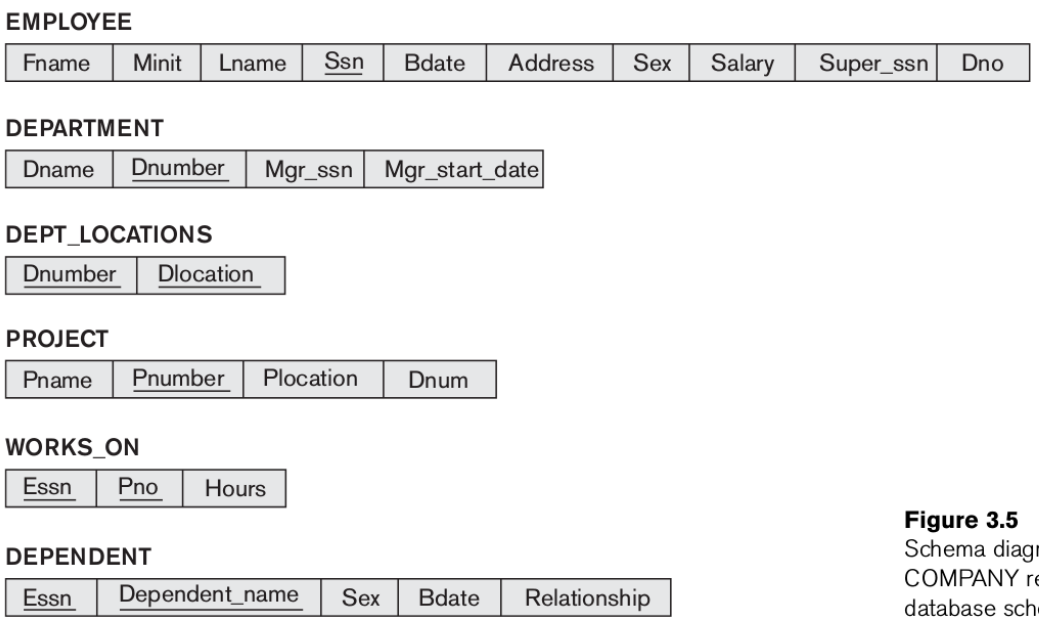

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: B

Question: List the names of all employees who have a dependent with the same first name as themselves

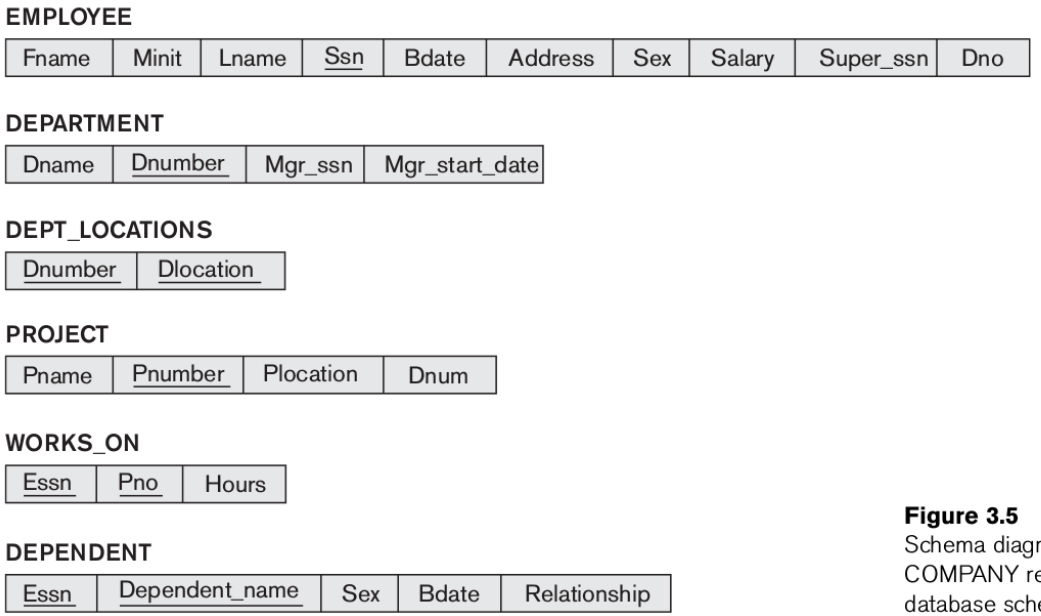

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: B

Question: List the names of all employees who have a dependent with the same first name as themselves

Answer:

```
SELECT Fname, Lname
FROM EMPLOYEE, DEPENDENT
WHERE Ssn=Essn
AND Dependent_name=Fname
```

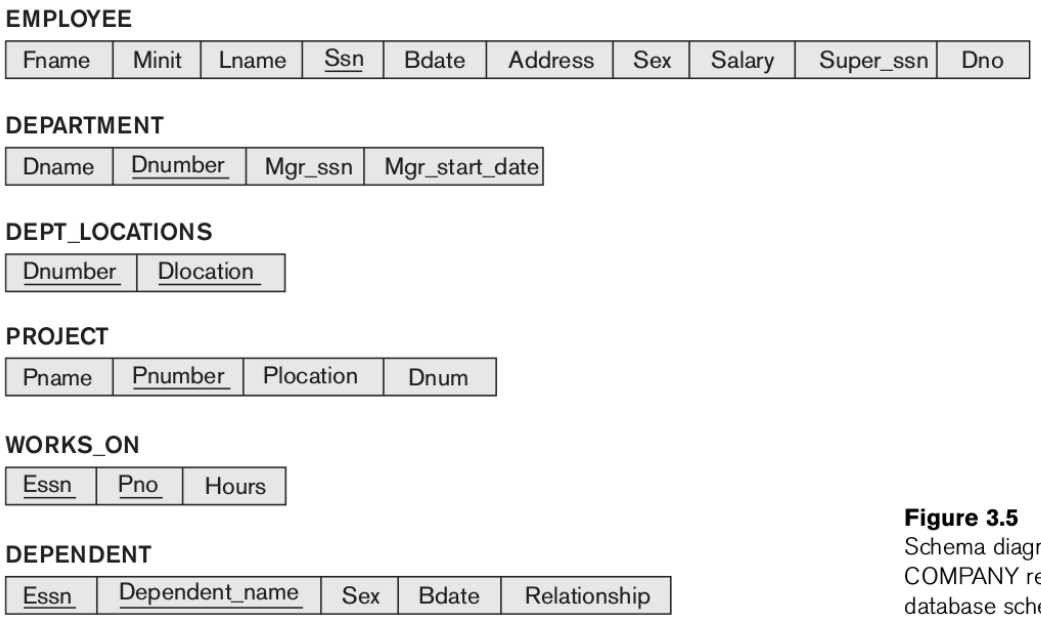

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: C

Question: Find the names of all employees who are directly supervised by “Franklin Wong”

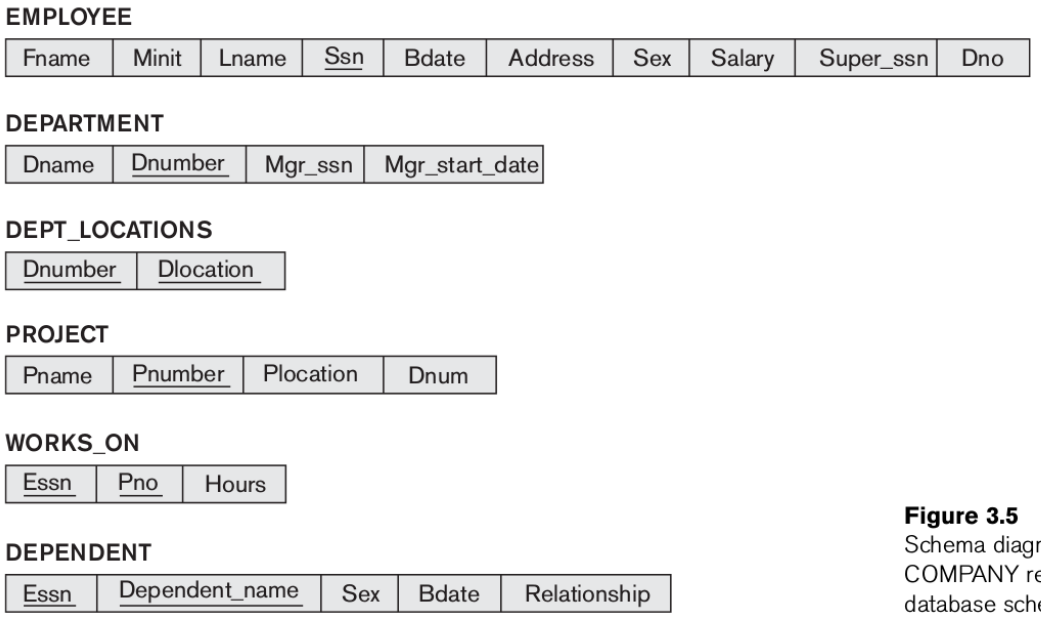

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: C

Question: Find the names of all employees who are directly supervised by "Franklin Wong"

```
ELECT E.Fname, E.Lname
FROM EMPLOYEE E, EMPLOYEE S
WHERE E.Super_ssn=S.Ssn
AND S.Fname='Franklin'
AND S.Lname='Wong'
```

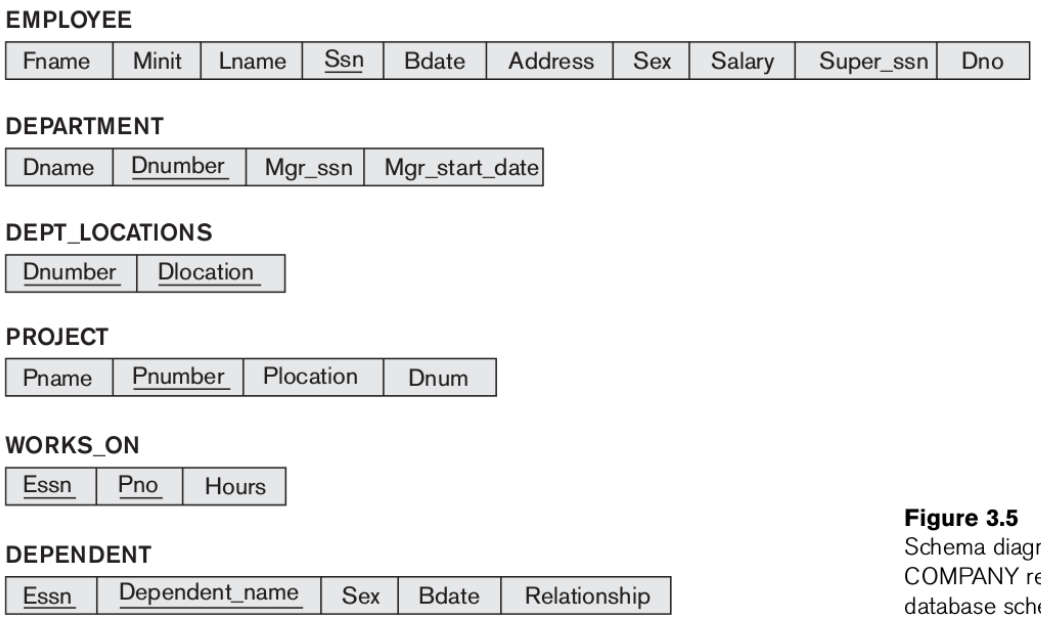

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: D

Question: For each project, list the project name and the total hours per week (by all employees) spent on that project

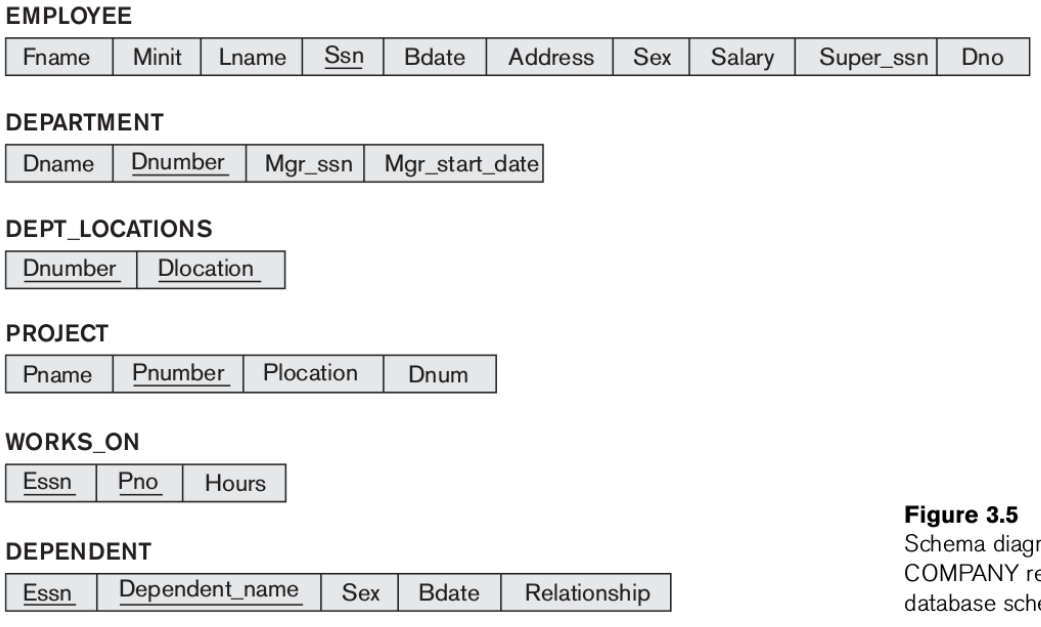

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: D

Question: For each project, list the project name and the total hours per week (by all employees) spent on that project

```
SELECT Pno, Pname, SUM(Hours)
FROM PROJECT JOIN WORKS_ON ON Pno=Pnumber
GROUP BY Pno, Pname
```

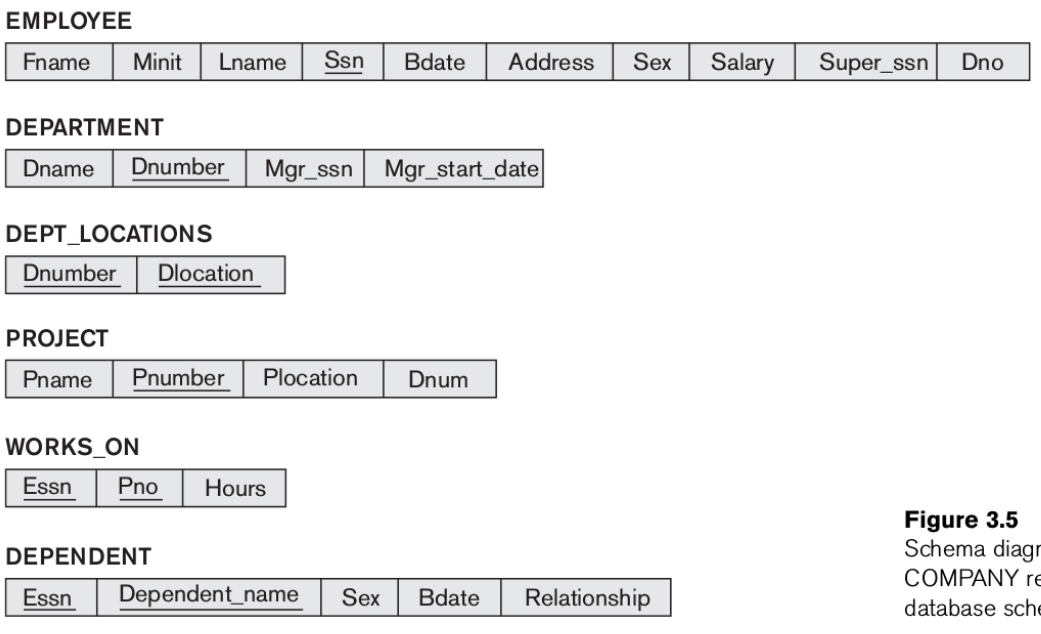

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: E

Question: Retrieve the names of all employees who work on every project

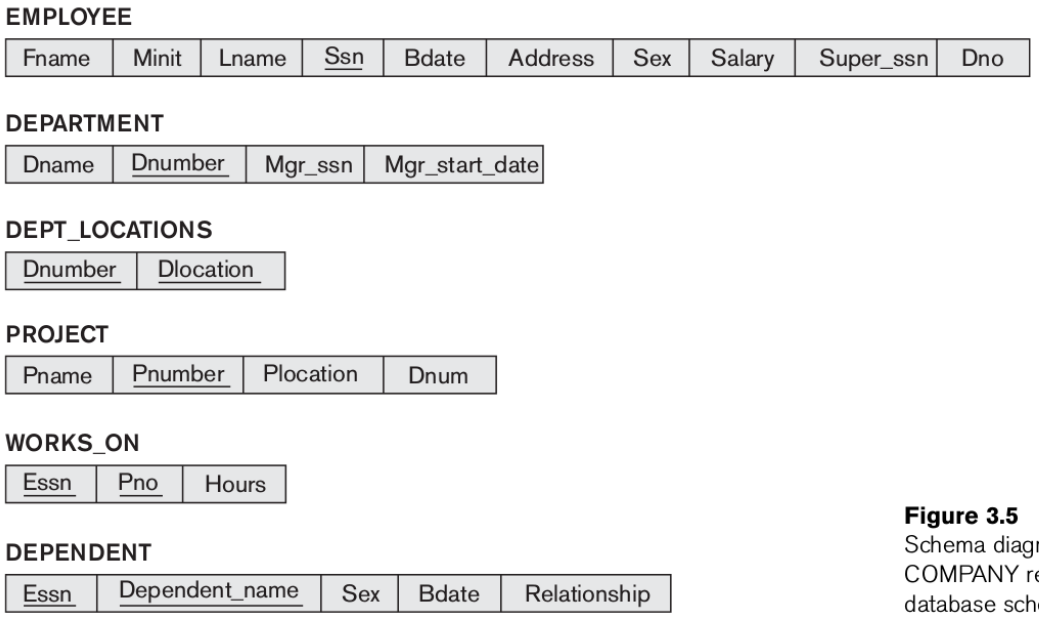

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: E

Question: Retrieve the names of all employees who work on every project

```
SELECT Fname, Lname
FROM EMPLOYEE
WHERE NOT EXISTS
 (SELECT *
  FROM PROJECT
   WHERE NOT EXISTS
    (SELECT *
     FROM WORKS_ON
      WHERE Pno = Pnumber
       AND Essn = Ssn))
```

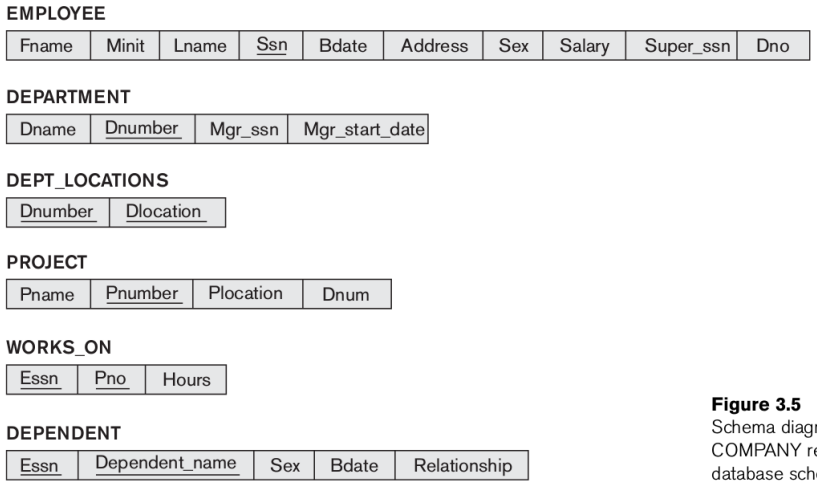

Figure 3.5
Schema diagram for t
COMPANY relational
database schema.

SQL: F

Question: Retrieve the names of all employees who do not work on any project

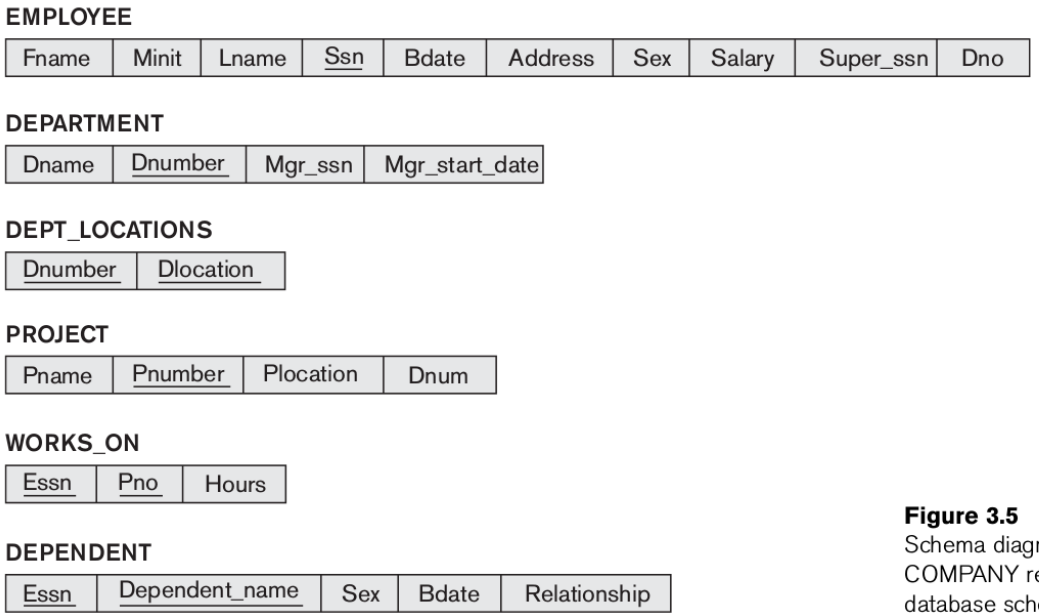

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: F

Question: Retrieve the names of all employees who do not work on any project

```
SELECT Fname, Lname
FROM EMPLOYEE
WHERE NOT EXISTS
 (SELECT *
  FROM WORKS_ON
   WHERE Ssn=Essn)
```

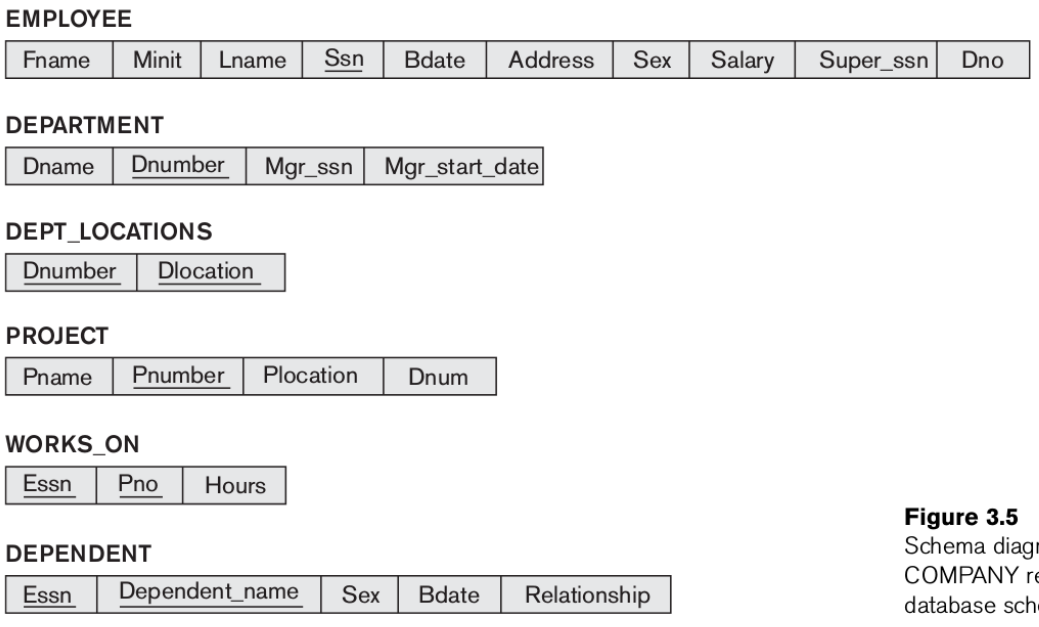

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: G

Question: For each department, retrieve the department name and the average salary

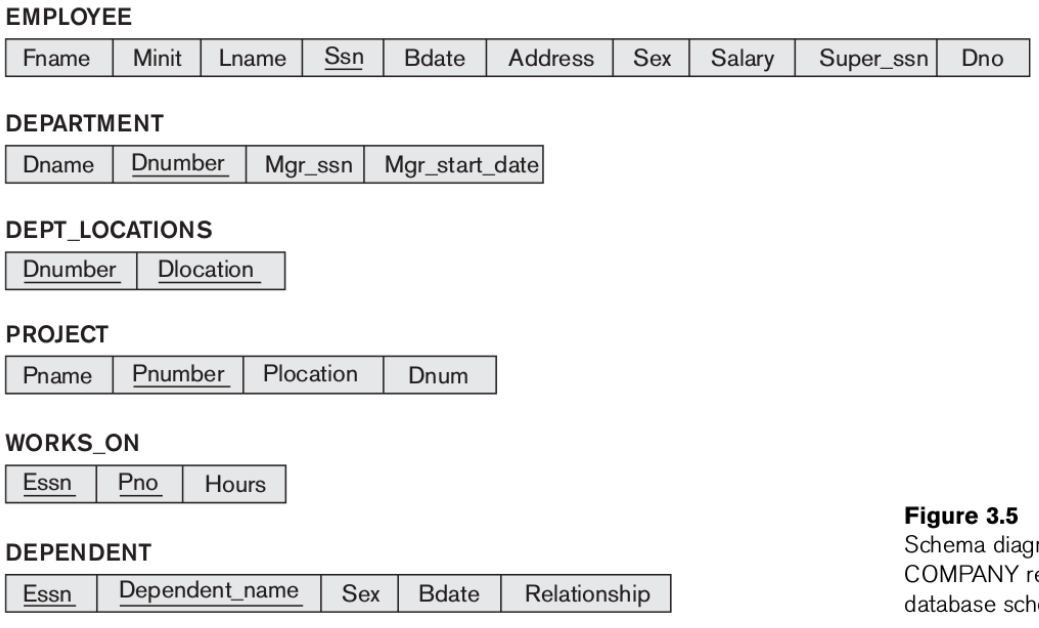

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: G

Question: For each department, retrieve the department name and the average salary

```
SELECT Dno, Dname, AVG(Salary)
FROM DEPARTMENT JOIN EMPLOYEE ON Dno=Dnumber
GROUP BY Dno, Dname
```

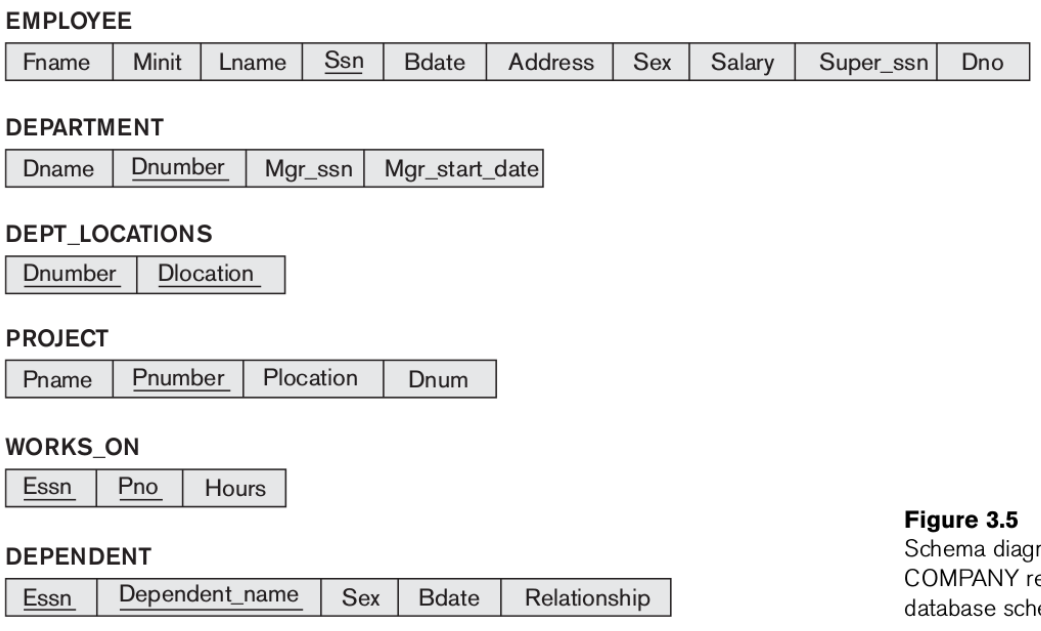

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: H

Question: Retrieve the average salary of all female employees

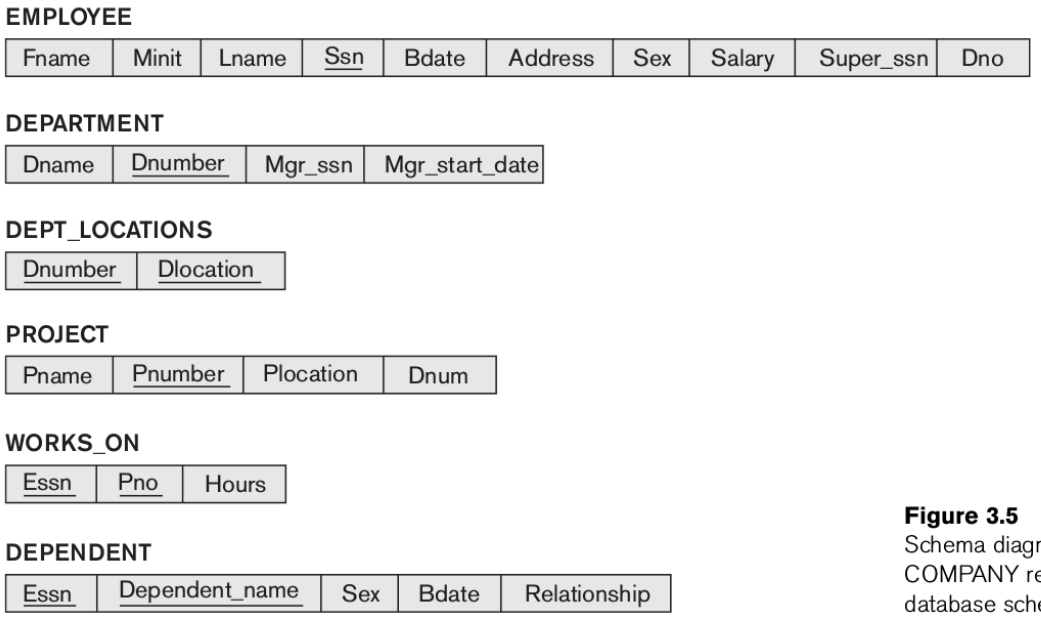

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: H

Question: Retrieve the average salary of all female employees

```
SELECT AVG(Salary)
FROM EMPLOYEE
WHERE Sex='F'
```

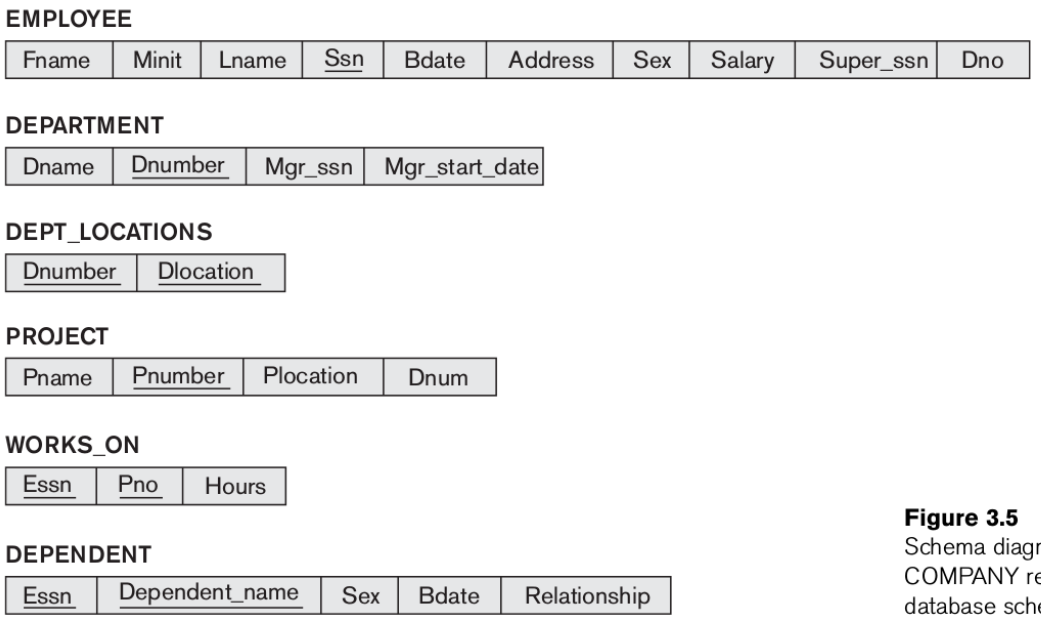

Figure 3.5
Schema diagram for the COMPANY relational database schema.

Question: Find the names and addresses of all employees who work on at least one project located in Houston but whose department has no location in Houston

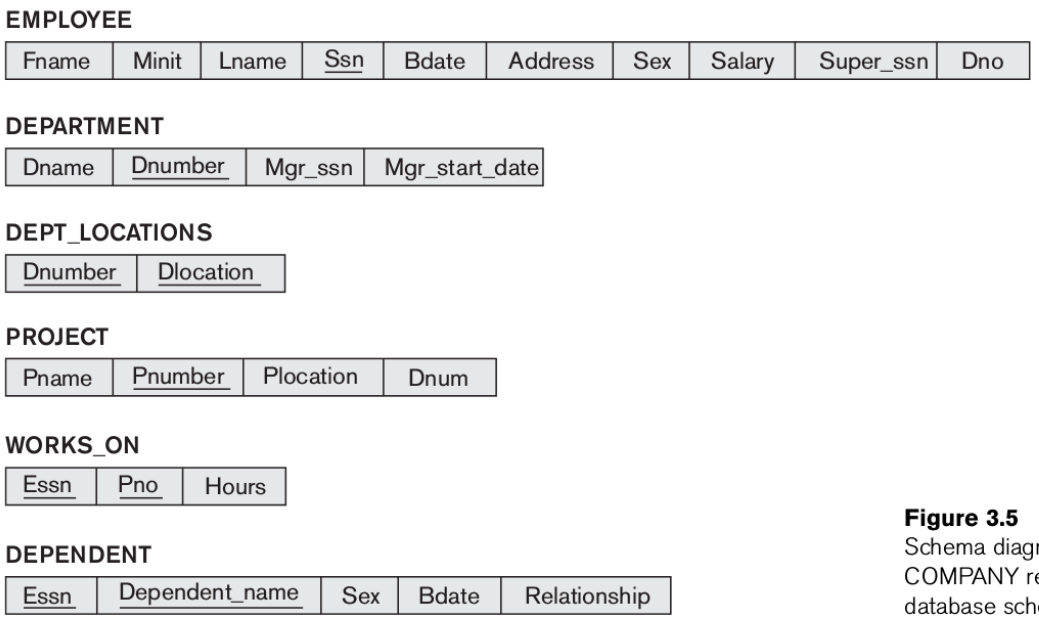

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: I

Question: Find the names and addresses of all employees who work on at least one project located in Houston but whose department has no location in Houston

```
SELECT Fname, Lname, Address
FROM EMPLOYEE
WHERE EXISTS
 (SELECT *
  FROM WORKS_ON, PROJECT
  WHERE Ssn = Essn
  AND Pno = Pnumber
  AND PLocation = 'Houston')
  AND NOT EXISTS
        (SELECT *
         FROM DEPT_LOCATIONS
         WHERE Dnumber = Dno
         AND DLocation = 'Houston')
```

SQL: J

Question: List the last names of all department managers who have no dependents

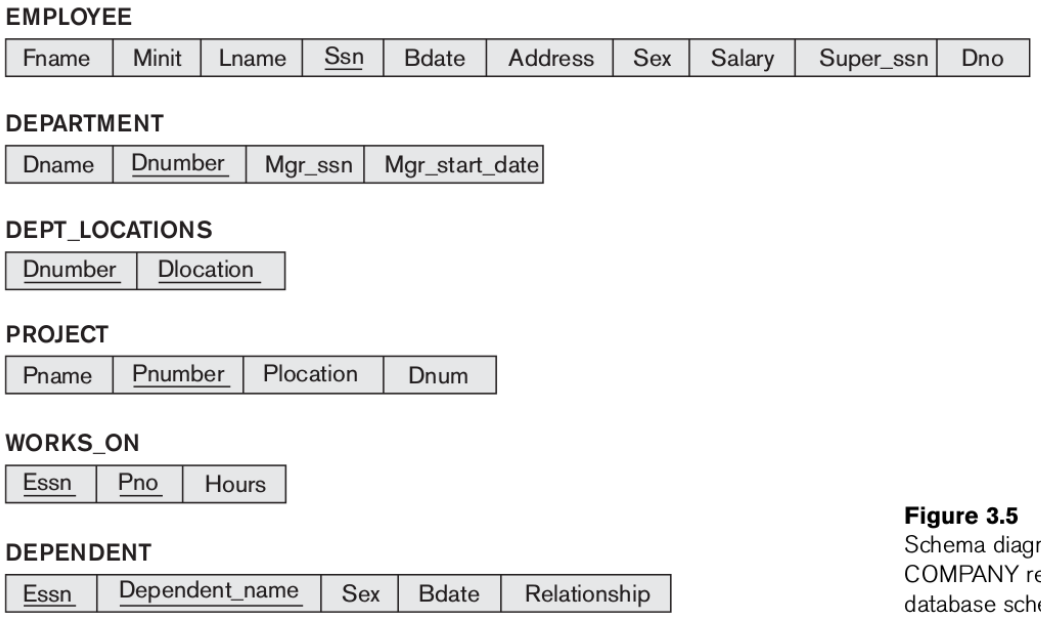

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL: J

Question: List the last names of all department managers who have no dependents

```
SELECT Lname
FROM EMPLOYEE, DEPARTMENT
WHERE Ssn=Mgr_ssn
 AND NOT EXISTS
  (SELECT *
   FROM DEPENDENT
    WHERE Essn = Ssn)
```

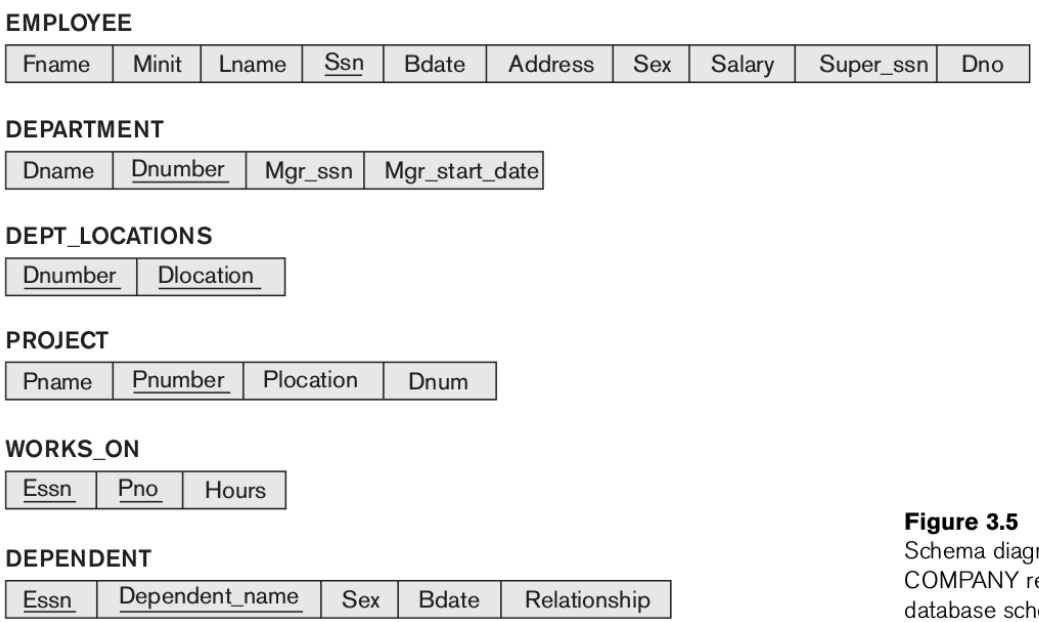

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Exercise 2: A

Question: For each department whose average employee salary is more than $30,000, retrieve the department name and the number of employees working for that department.

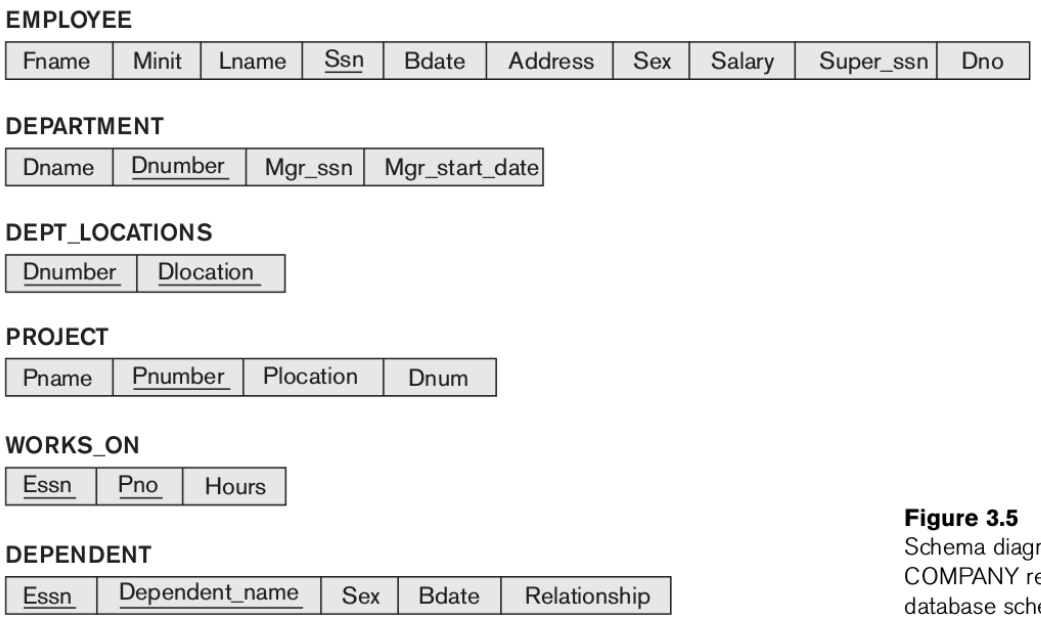

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Exercise 2: A

Question: For each department whose average employee salary is more than $30,000, retrieve the department name and the number of employees working for that department.

```
SELECT Dnumber, Dname, COUNT(*)
FROM DEPARTMENT, EMPLOYEE
WHERE Dno=Dnumber
GROUP BY Dnumber, Dname
HAVING AVG(Salary) > 30000
```

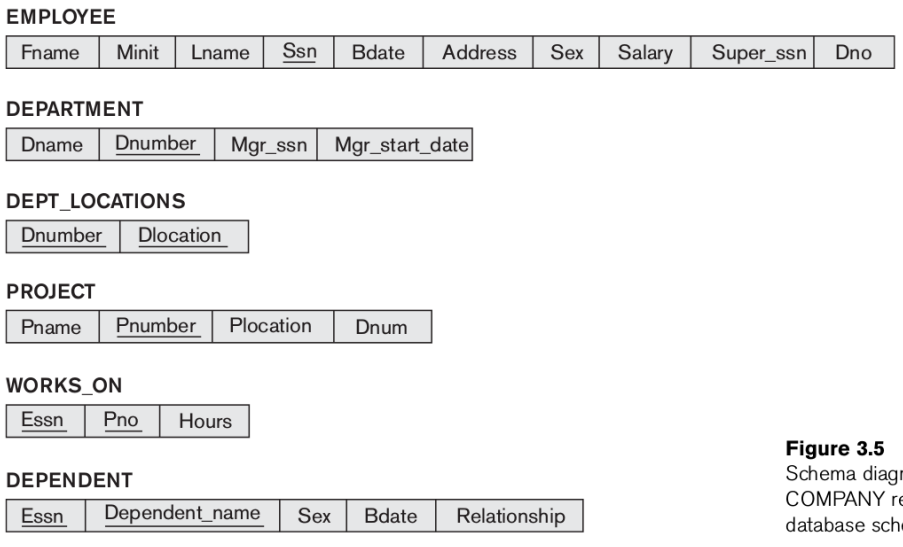

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Exercise 2: B

Previous: For each department whose average employee salary is more than $30,000, retrieve the department name and the number of employees working for that department

Question: Suppose that we want the number of male employees in each department rather than all employees. Can we specify this query in SQL? Why or why not?

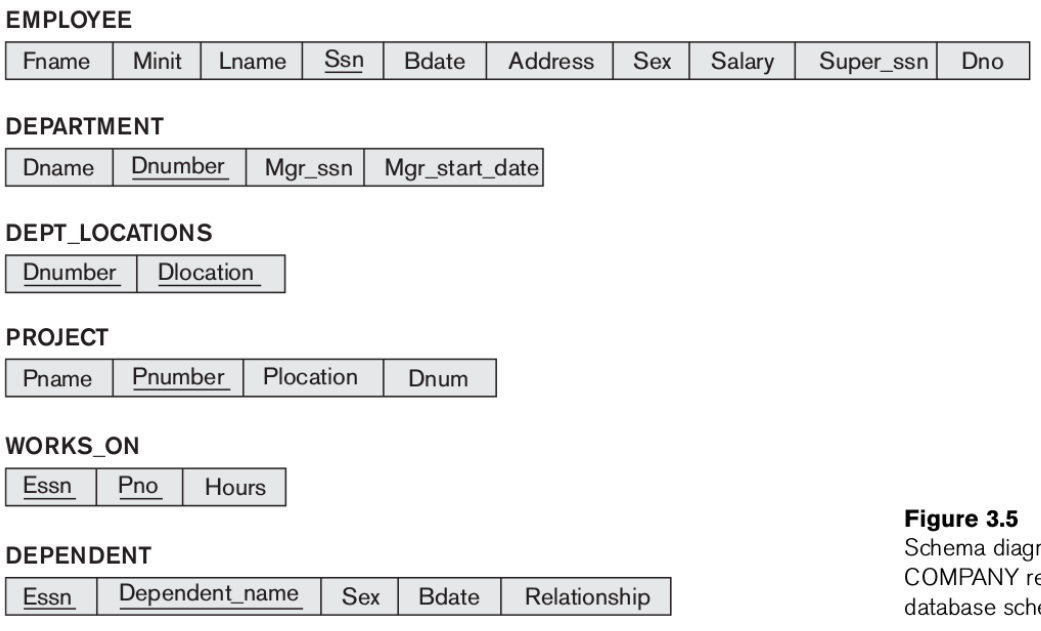

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Exercise 2: B

Previous: For each department whose average employee salary is more than $30,000, retrieve the department name and the number of employees working for that department
Question: Suppose that we want the number of male employees in each department rather than all employees

```
SELECT Dnumber, Dname, COUNT(*)
FROM DEPARTMENT, EMPLOYEE
WHERE Sex = 'M'
AND Dno = Dnumber
AND Dno IN
   (SELECT Dno
    FROM EMPLOYEE
    GROUP BY Dno
    HAVING AVG(Salary) > 30000)
    GROUP BY Dnumber, Dname
```

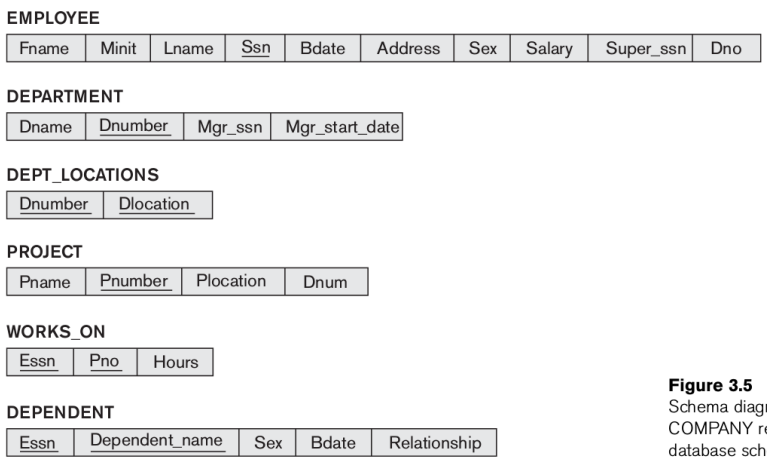

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: A

Question: A view that has the department name, manager name, and manager salary for every department

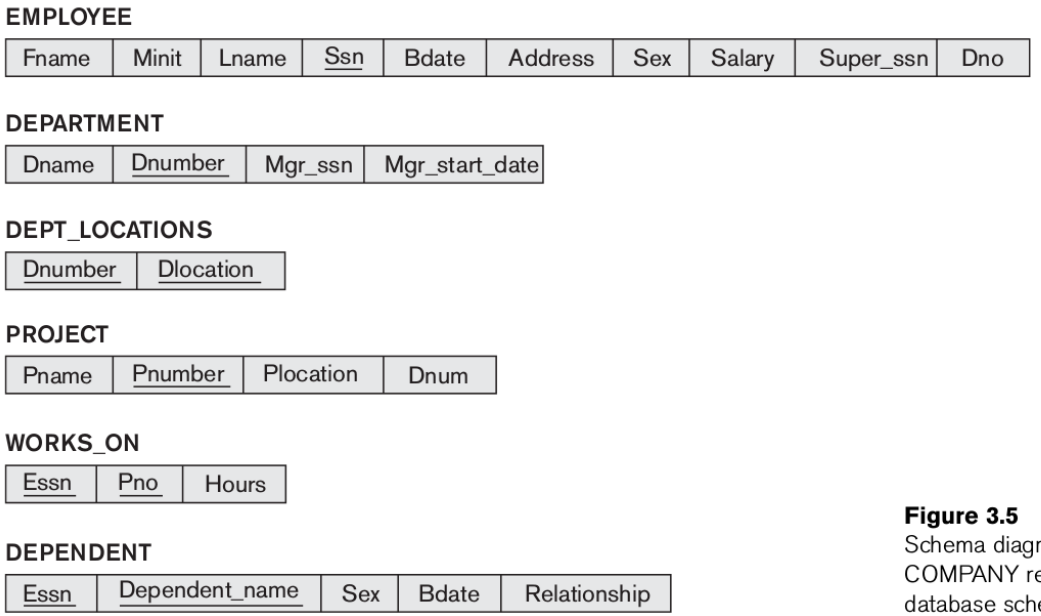

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: A

Question: A view that has the department name, manager name, and manager salary for every department

```
REATE VIEW VIEW_A (Dname, Mgr_fname, Mgr_lname, Mgr_sal)
AS
SELECT Dname, Fname, Lname, Salary
FROM DEPARTMENT, EMPLOYEE
WHERE Mgr_ssn = Ssn
```

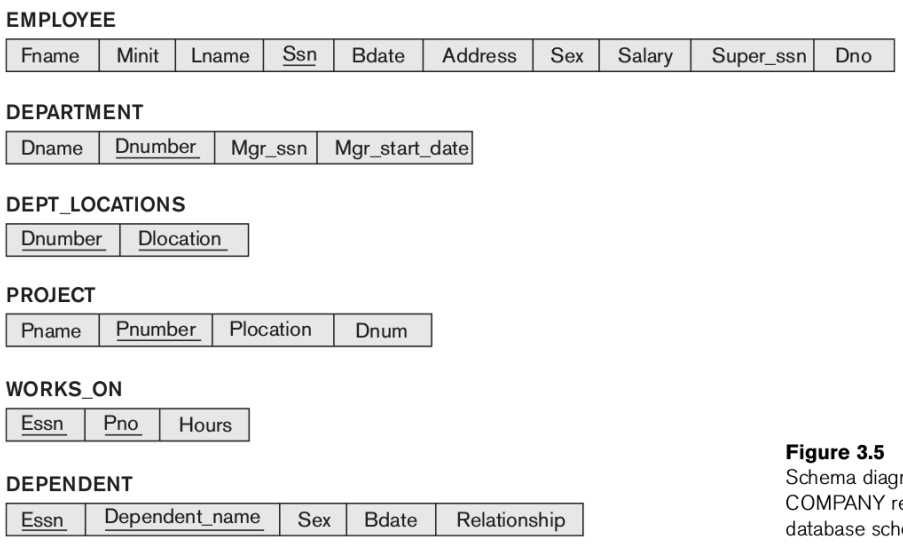

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: B

Question: A view that has the employee name, supervisor name, and employee salary for each employee who works in the "Research" department.

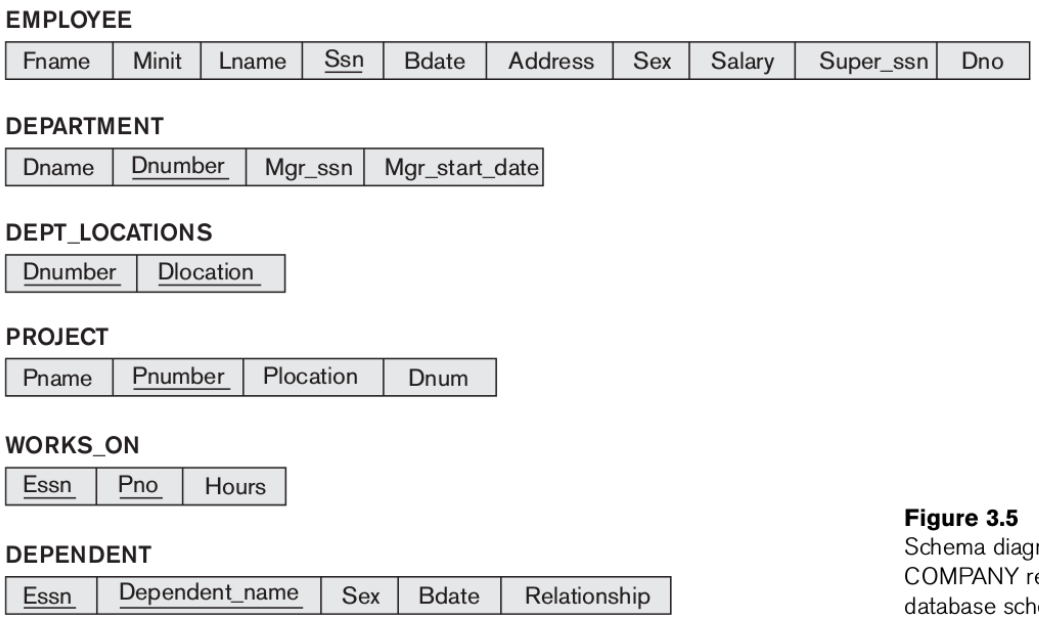

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: B

Question: A view that has the employee name, supervisor name, and employee salary for each employee who works in the "Research" department.

```
CREATE VIEW VIEW_B (EFname, ELname, SFname, SLname, ESalary
AS
SELECT E.Fname, E.Lname, S.Fname, S.Lname, E.Salary
FROM EMPLOYEE E S, DEPARTMENT
WHERE E.Super_ssn = S.Ssn
AND E.Dno = Dnumber
AND Dname = 'Research'
```

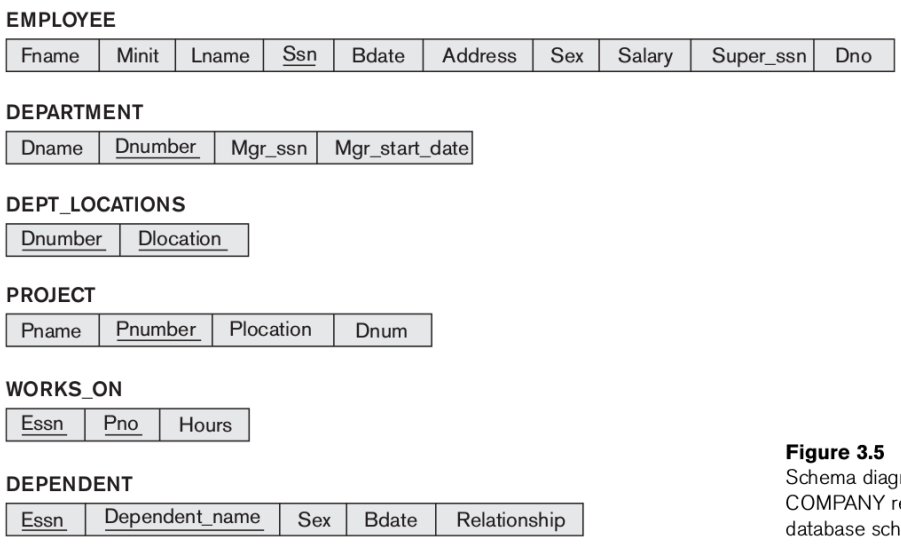

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: C

Question: A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project

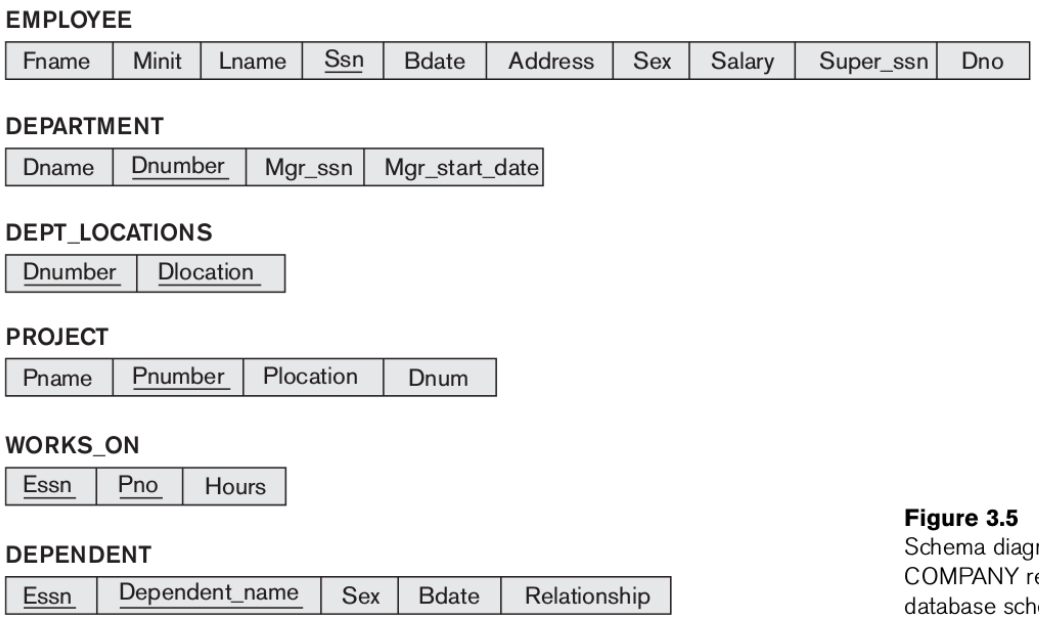

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: C

Question: A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project

```
CREATE VIEW VIEW_C (Pnumber, Pname, Dname, NREMP, Tot_hours
AS
SELECT Pnumber, Pname, Dname, COUNT(*), SUM(Hours)
FROM PROJECT, DEPARTMENT, WORKS_ON
WHERE Dnumber = Dnum
AND Pnumber = Pno
GROUP BY Pnumber, Pname, Dname
```

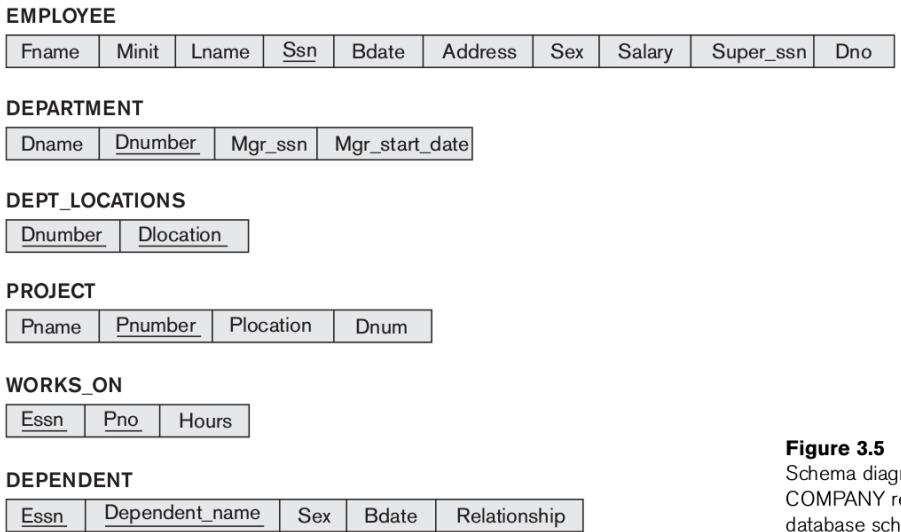

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: D

Question: A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

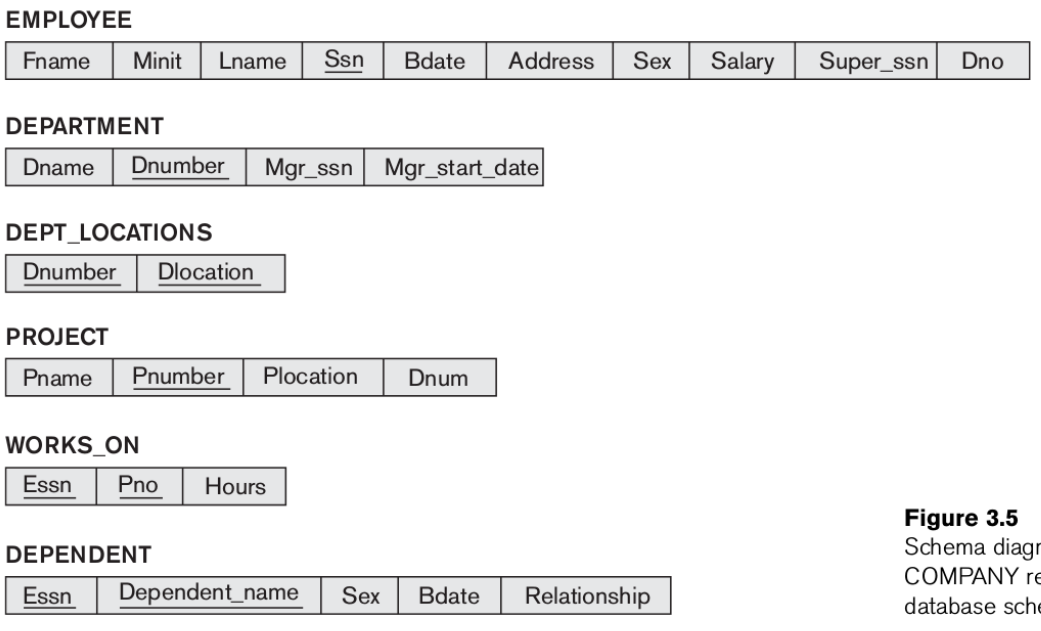

Figure 3.5
Schema diagram for the COMPANY relational database schema.

SQL Views: D

Question: A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

```
CREATE VIEW VIEW_D (Pnumber, Pname, Dname, NREMP, TOTHours)
AS
SELECT Pnumber, Pname, Dname, COUNT(*), SUM(Hours)
FROM PROJECT, DEPARTMENT, WORKS_ON
WHERE Dnumber = Dnum
AND Pnumber = Pno
GROUP BY Pnumber, Pname, Dname
HAVING COUNT(*) > 1
```

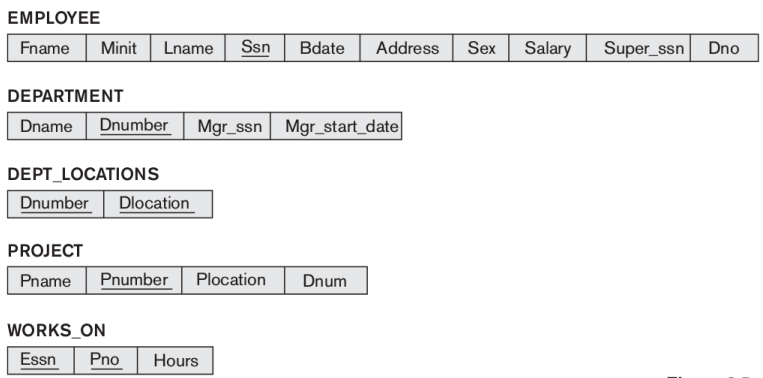

Figure 3.5

SQL Views Strike Back: A

Question: is it allowed? If so, what is the corresponding query?

```
SELECT *
FROM DEPT_SUMMARY
```

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```

SQL Views Strike Back: A

Question: is it allowed? If so, what is the corresponding query?

```
SELECT *
FROM DEPT_SUMMARY
```

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```

Corresponding query:

```
SELECT Dno, COUNT(*), SUM(Salary), AVG(Salary)
FROM EMPLOYEE
GROUP BY Dno
```

SQL Views Strike Back: B

Question: is it allowed? If so, what is the corresponding query?

```
SELECT D, C
FROM DEPT_SUMMARY
WHERE TOTAL_S>100000;
```

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```

SQL Views Strike Back: B

Question: is it allowed? If so, what is the corresponding query?

```
SELECT D, C
FROM DEPT_SUMMARY
WHERE TOTAL_S>100000;
```

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```

Corresponding query:

```
SELECT Dno, COUNT(*)
FROM EMPLOYEE
GROUP BY Dno
HAVING SUM(Salary)>100000
```

SQL Views Strike Back: C

Question: is it allowed? If so, what is the corresponding query?

```
UPDATE DEPT_SUMMARY
SET D=3
WHERE D=4;
```

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```

SQL Views Strike Back: C

Question: is it allowed? If so, what is the corresponding query?

```
UPDATE DEPT_SUMMARY
SET D=3
WHERE D=4;
```

Answer: Nope, there is a referential problem: the employee's department number refers to the department relation and also updates over aggregates are generally not allowed.

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```

SQL Views Strike Back: D

Question: is it allowed? If so, what is the corresponding query?

```
DELETE FROM DEPT_SUMMARY
WHERE C>4;
```

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```

SQL Views Strike Back: D

Question: is it allowed? If so, what is the corresponding query?

```
DELETE FROM DEPT_SUMMARY
WHERE C>4;
```

Answer: Deletion by an aggregated column is not allowed

Rewritten version:

```
DELETE FROM EMPLOYEE
WHERE Dno IN
(SELECT Dno
 FROM EMPLOYEE
 GROUP BY Dno
 HAVING COUNT(*) > 4)
```

View:

```
CREATE VIEW DEPT_SUMMARY (D, C, Total_s, Average_s)
AS SELECT Dno, COUNT (*), SUM (Salary), AVG (Salary)
FROM  EMPLOYEE
GROUP BY Dno;
```