Data Modeling and Implementation Techniques (CS 457)
Homework 4B

Exercise 8.15 Show the result of each of the sample queries in Section 8.5 as it would apply to the database state in Figure 5.6.

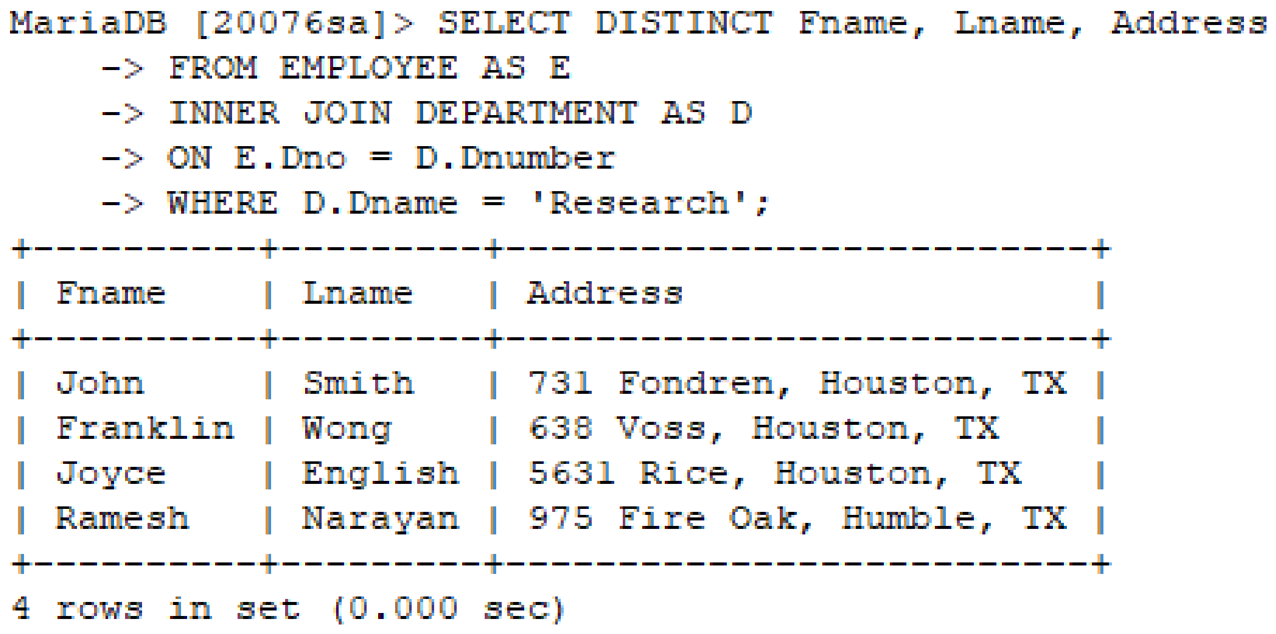
Ans.:

Query-1. Retrieve the name and address of all employees who work for the ‘Research’ department.

RESEARCH_DEPT ← σDname = ‘Research’ (DEPARTMENT)

RESEARCH_EMPS ← (RESEARCH_DEPT Dnumber = Dno EMPLOYEE)

RESULT ← πFname, Lname, Address (RESEARCH_EMPS)



```
MariaDB [20076sa]> SELECT DISTINCT Fname, Lname, Address
    -> FROM EMPLOYEE AS E
    -> INNER JOIN DEPARTMENT AS D
    -> ON E.Dno = D.Dnumber
    -> WHERE D.Dname = 'Research';
+----------+---------+--------------------------+
| Fname    | Lname   | Address                  |
+----------+---------+--------------------------+
| John     | Smith   | 731 Fondren, Houston, TX |
| Franklin | Wong    | 638 Voss, Houston, TX    |
| Joyce    | English | 5631 Rice, Houston, TX   |
| Ramesh   | Narayan | 975 Fire Oak, Humble, TX |
+----------+---------+--------------------------+
4 rows in set (0.000 sec)
```

Query-2. For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

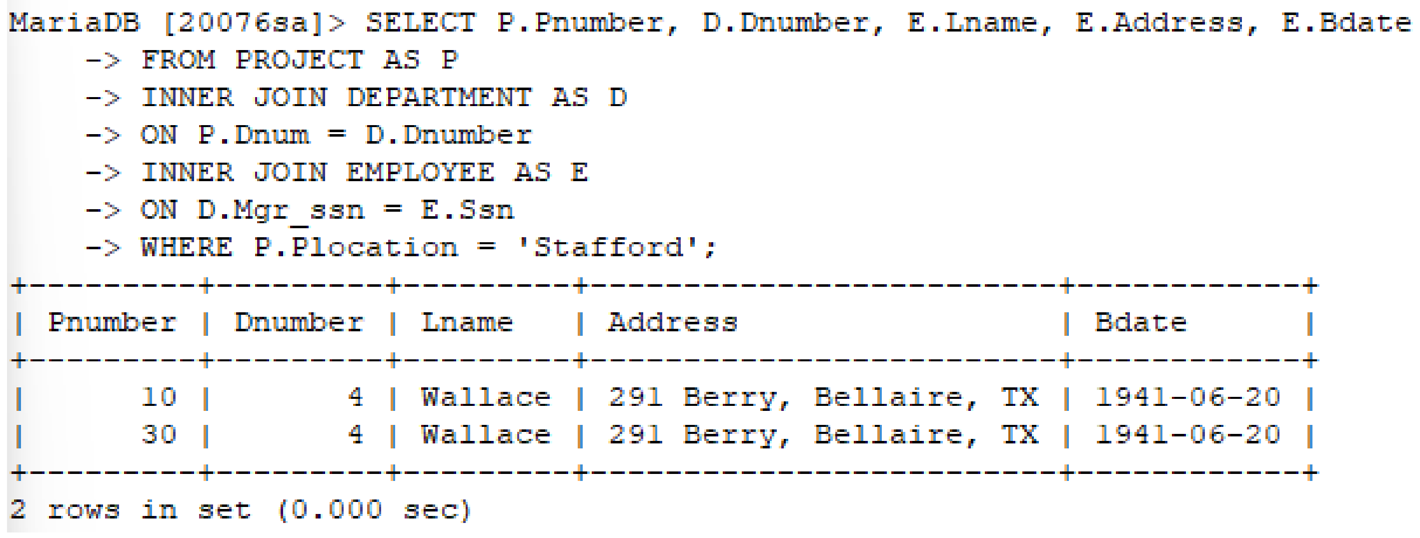
STAFFORD_PROJS ← σPlocation = ‘Stafford’(PROJECT)

CONTR_DEPTS ← (STAFFORD_PROJS Dnum = Dnumber DEPARTMENT)

PROJ_DEPT_MGRS ← (CONTR_DEPTS Mgr_ssn = Ssn EMPLOYEE)

RESULT ← πPnumber, Dnum, Lname, Address, Bdate (PROJ_DEPT_MGRS)

At first, we select the projects from Stafford, then join them with departments, and then join the result with the department managers.



```
MariaDB [20076sa]> SELECT P.Pnumber, D.Dnumber, E.Lname, E.Address, E.Bdate
    -> FROM PROJECT AS P
    -> INNER JOIN DEPARTMENT AS D
    -> ON P.Dnum = D.Dnumber
    -> INNER JOIN EMPLOYEE AS E
    -> ON D.Mgr_ssn = E.Ssn
    -> WHERE P.Plocation = 'Stafford';
+---------+---------+---------+------------------------+------------+
| Pnumber | Dnumber | Lname   | Address                | Bdate      |
+---------+---------+---------+------------------------+------------+
|      10 |       4 | Wallace | 291 Berry, Bellaire, TX | 1941-06-20 |
|      30 |       4 | Wallace | 291 Berry, Bellaire, TX | 1941-06-20 |
+---------+---------+---------+------------------------+------------+
2 rows in set (0.000 sec)
```

Query-3. Find the names of employees who work on all the projects controlled by department number 5.

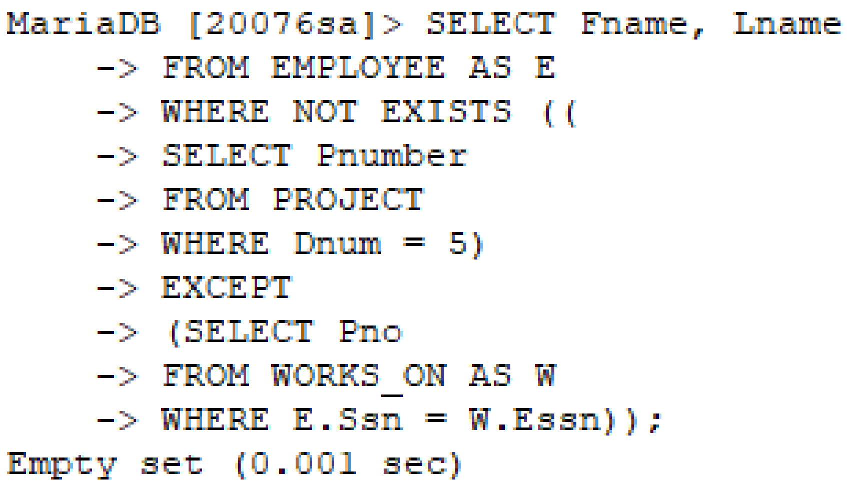
DEPT5_PROJS ← ρ(Pno) (πPnumber(σDnum = 5 (PROJECT)))

EMP_PROJ ← ρ(Ssn, Pno) (πEssn, Pno (WORKS_ON))

RESULT_EMP_SSNS ← EMP_PROJ ÷ DEPT5_PROJS

RESULT ← πLname, Fname (RESULT_EMP_SSNS * EMPLOYEE)

At first, create a table DEPT5_PROJS with project numbers of all projects from department 5. Then we create a table EMP_PROJ which has (Ssn, Pno), and apply the division. Also we have renamed the attributes. At the end, we join the result of the division, which holds only Ssn values, with the EMPLOYEE table to retrieve the Fname, Lname attributes from EMPLOYEE.



```
MariaDB [20076sa]> SELECT Fname, Lname
    -> FROM EMPLOYEE AS E
    -> WHERE NOT EXISTS ((
    -> SELECT Pnumber
    -> FROM PROJECT
    -> WHERE Dnum = 5)
    -> EXCEPT
    -> (SELECT Pno
    -> FROM WORKS_ON AS W
    -> WHERE E.Ssn = W.Essn));
Empty set (0.001 sec)
```

Query-4. Make a list of project numbers for projects that involve an employee whose last name is ‘Smith’, either as a worker or as a manager of the department that controls the project.

SMITHS(Essn) ← πSsn (σLname = ‘Smith’(EMPLOYEE))

SMITH_WORKER_PROJS ← πPno(WORKS_ON * SMITHS)

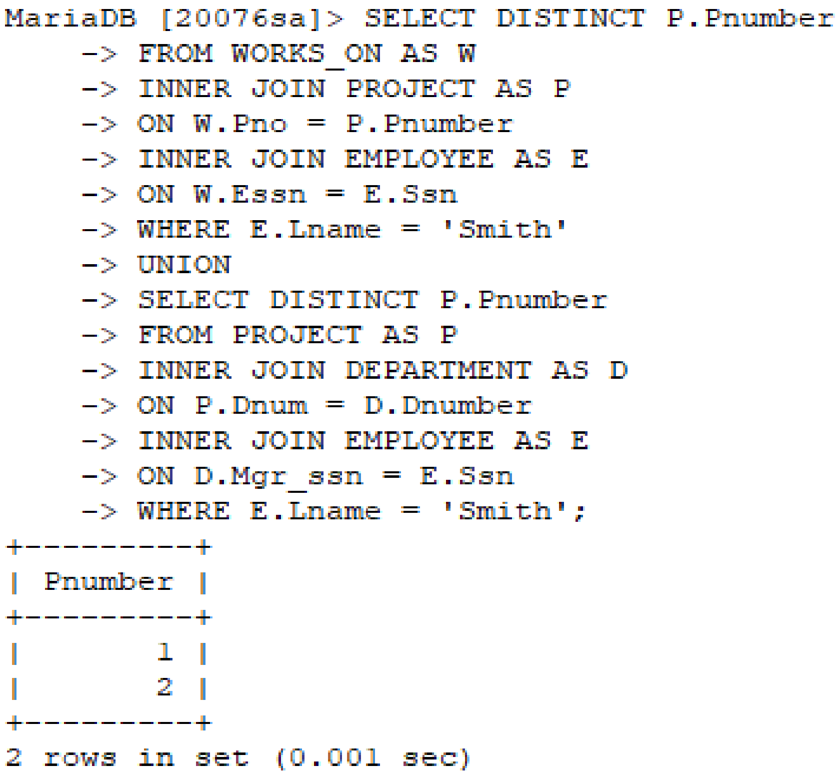
MGRS ← πLname, Dnumber (EMPLOYEE Ssn = Mgr_ssn DEPARTMENT)

SMITH_MANAGED_DEPTS (Dnum) ← πDnumber (σLname = ‘Smith’ (MGRS))

SMITH_MGR_PROJS (Pno) ← πPnumber (SMITH_MANAGED_DEPTS * PROJECT)

RESULT ← (SMITH_WORKER_PROJS ∪ SMITH_MGR_PROJS)

First, we retrieved the project numbers for those which involve an employee name Smith as a worker in SMITH_WORKER_PROJS. Then we retrieved the project numbers those which have an employee Smith as a manager of department which controls the project SMITH_MGR_PROJS. And then, we used UNION operation between SMITH_WORKER_PROJS and SMITH_MGR_PROJS.



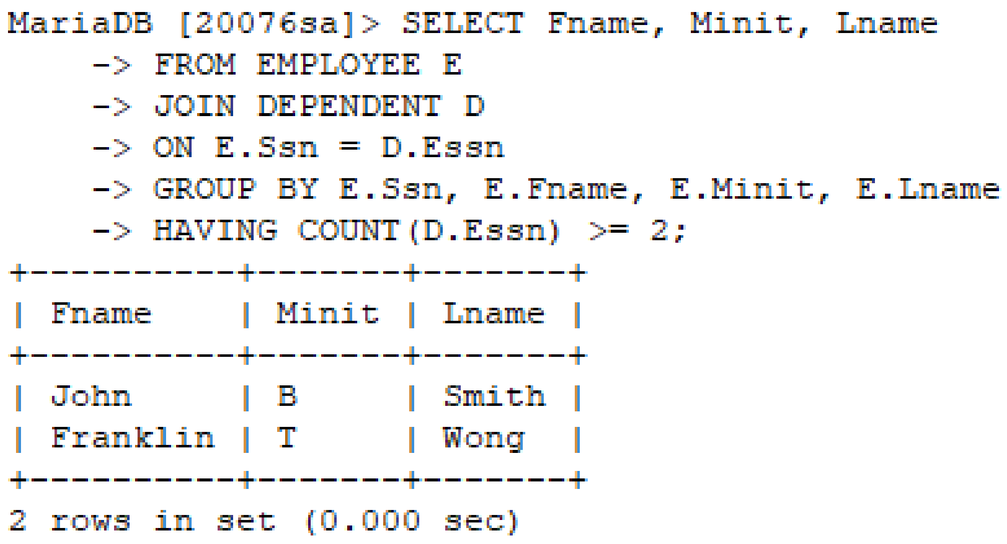
```
MariaDB [20076sa]> SELECT DISTINCT P.Pnumber
    -> FROM WORKS_ON AS W
    -> INNER JOIN PROJECT AS P
    -> ON W.Pno = P.Pnumber
    -> INNER JOIN EMPLOYEE AS E
    -> ON W.Essn = E.Ssn
    -> WHERE E.Lname = 'Smith'
    -> UNION
    -> SELECT DISTINCT P.Pnumber
    -> FROM PROJECT AS P
    -> INNER JOIN DEPARTMENT AS D
    -> ON P.Dnum = D.Dnumber
    -> INNER JOIN EMPLOYEE AS E
    -> ON D.Mgr_ssn = E.Ssn
    -> WHERE E.Lname = 'Smith';
+---------+
| Pnumber |
+---------+
|       1 |
|       2 |
+---------+
2 rows in set (0.001 sec)
```

Query-5. List the names of all employees with two or more dependents.

D1 (Ssn, No_of_dependents)← Essn ℑ COUNT Dependent_name (DEPENDENT)

D2 ← σNo_of_dependents > 2 (D1)

RESULT ← πLname, Fname (D2 * EMPLOYEE)



```
MariaDB [20076sa]> SELECT Fname, Minit, Lname
    -> FROM EMPLOYEE E
    -> JOIN DEPENDENT D
    -> ON E.Ssn = D.Essn
    -> GROUP BY E.Ssn, E.Fname, E.Minit, E.Lname
    -> HAVING COUNT(D.Essn) >= 2;
+----------+-------+-------+
| Fname    | Minit | Lname |
+----------+-------+-------+
| John     | B     | Smith |
| Franklin | T     | Wong  |
+----------+-------+-------+
2 rows in set (0.000 sec)
```

\

Query-6. Retrieve the names of employees who have no dependents.

ALL_EMPLOYEES ← πSsn (EMPLOYEE)

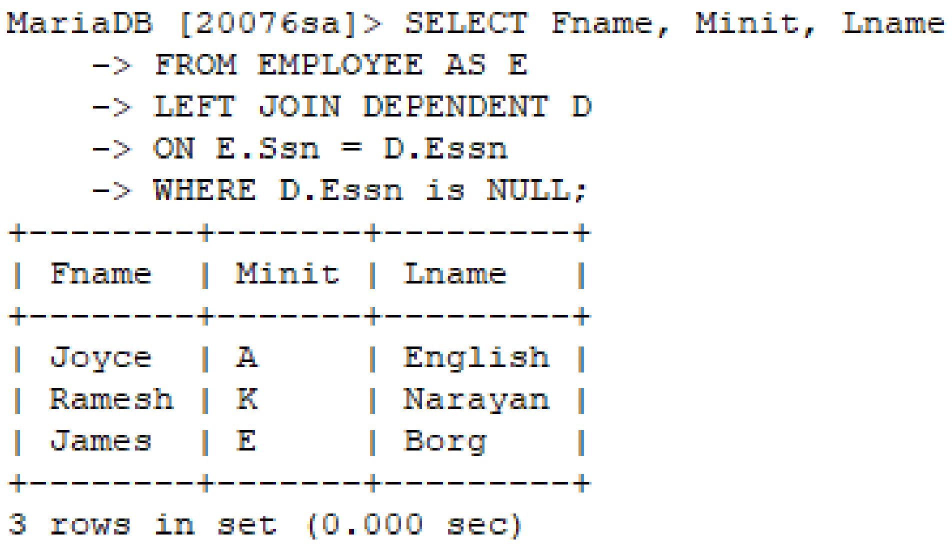
EMPS_WITH_DEPS (Ssn) ← πEssn (DEPENDENT)

EMPS_WITHOUT_DEPS ← (ALL_EMPLOYEES – EMPS_WITH_DEPS)

RESULT ← πLname, Fname (EMPS_WITHOUT_DEPS * EMPLOYEE)

At first, we're retrieving all employee Ssns in ALL_ EMPLOYEES. Then creating a table with the Ssns of those who are having minimum one dependent in EMPS_WITH_DEPS. We use the SET DIFFERENCE operation then to get employees Ssns with no dependents in EMPS_WITHOUT_DEPS, and at the end joining this with EMPLOYEE to retrieve the desired details.
This query can be written as:

πLname, Fname ((πSsn (EMPLOYEE) – ρSsn (πEssn (DEPENDENT))) * EMPLOYEE)



```
MariaDB [20076sa]> SELECT Fname, Minit, Lname
    -> FROM EMPLOYEE AS E
    -> LEFT JOIN DEPENDENT D
    -> ON E.Ssn = D.Essn
    -> WHERE D.Essn is NULL;
+--------+-------+---------+
| Fname  | Minit | Lname   |
+--------+-------+---------+
| Joyce  | A     | English |
| Ramesh | K     | Narayan |
| James  | E     | Borg    |
+--------+-------+---------+
3 rows in set (0.000 sec)
```

Query-7. List the names of managers who have at least one dependent.

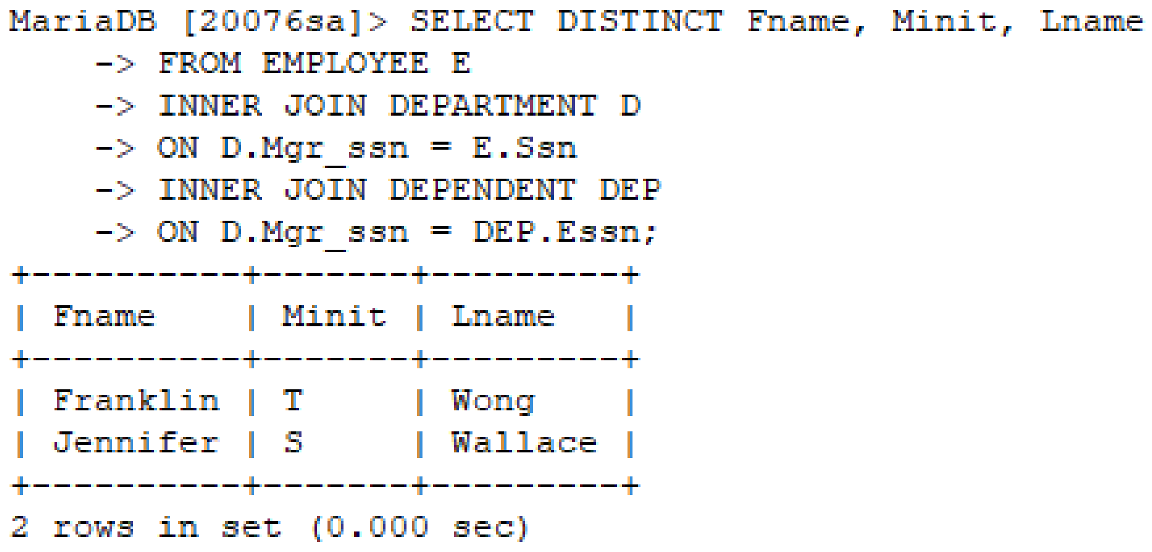
MANAGRS (Ssn) ← πMgr_ssn (DEPARTMENT)

EMPS_WITH_DEPS (Ssn) ← πEssn (DEPENDENT)

MANAGRS_WITH_DEPS ← (MGRS ∩ EMPS_WITH_DEPS)

RESULT ← πLname, Fname (MGRS_WITH_DEPS * EMPLOYEE)

we retrieve the Ssns of managers in MANAGRS, and employees' Ssns with at least one dependent in EMPS_WITH_DEPS, and then we apply the Set-Interaction to get the managers' Ssns of with having one dependent minimum.

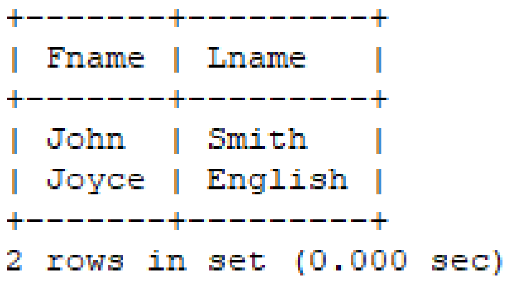


```
MariaDB [20076sa]> SELECT DISTINCT Fname, Minit, Lname
    -> FROM EMPLOYEE E
    -> INNER JOIN DEPARTMENT D
    -> ON D.Mgr_ssn = E.Ssn
    -> INNER JOIN DEPENDENT DEP
    -> ON D.Mgr_ssn = DEP.Essn;
+----------+-------+---------+
| Fname    | Minit | Lname   |
+----------+-------+---------+
| Franklin | T     | Wong    |
| Jennifer | S     | Wallace |
+----------+-------+---------+
2 rows in set (0.000 sec)
```

Exercise 8.16 Specify the following queries on the COMPANY relational database schema shown in Figure 5.5 using the relational operators discussed in this chapter. Also show the result of each query as it would apply to the database state in Figure 5.6.

a) **Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.**

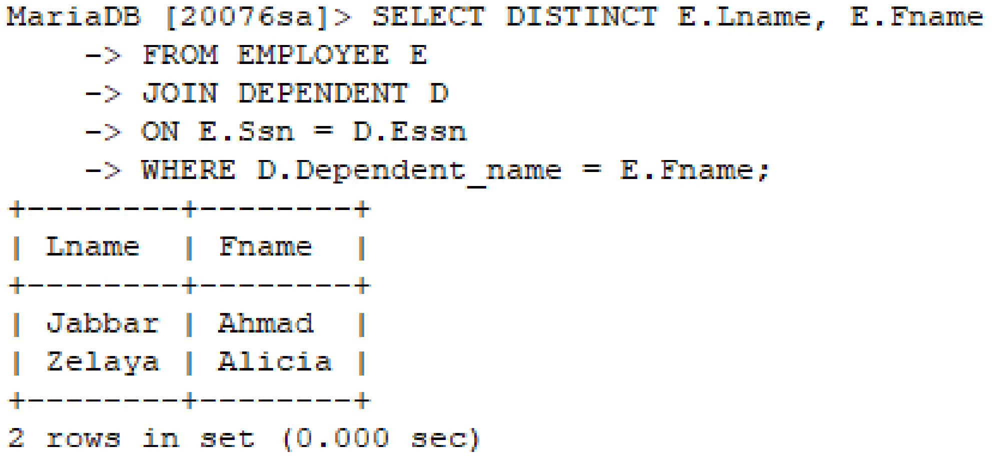
EMP_W_X <-- $\sigma_{Pname = 'ProductX'}$ (PROJECT)) $\bowtie_{(Pnumber),(Pno)}$ (WORKS_ON)
EMP_WORK_10 <-- (EMPLOYEE) ⋈ (Ssn) , (Essn) ($\sigma_{HOURS >10}$ (EMP_W_X))
RESULT<-- $\pi_{Lname, Fname}$ ($\sigma_{DNO = 5}$ (EMP_WORK_10))



```
+-------+---------+
| Fname | Lname   |
+-------+---------+
| John  | Smith   |
| Joyce | English |
+-------+---------+
2 rows in set (0.000 sec)
```

b) **List the names of all employees who have a dependent with the same first name as themselves.**

E <-- (EMPLOYEE) $_{(Ssn, Fname)}$ ⋈ $_{(Essn, Dependent\_name)}$ (DEPENDENT)
R <-- π_{Fname} (E)



```
MariaDB [20076sa]> SELECT DISTINCT E.Lname, E.Fname
    -> FROM EMPLOYEE E
    -> JOIN DEPENDENT D
    -> ON E.Ssn = D.Essn
    -> WHERE D.Dependent_name = E.Fname;
+--------+--------+
| Lname  | Fname  |
+--------+--------+
| Jabbar | Ahmad  |
| Zelaya | Alicia |
+--------+--------+
2 rows in set (0.000 sec)
```

c) **<u>Find the names of all employees who are directly supervised by ‘Franklin Wong’</u>.**

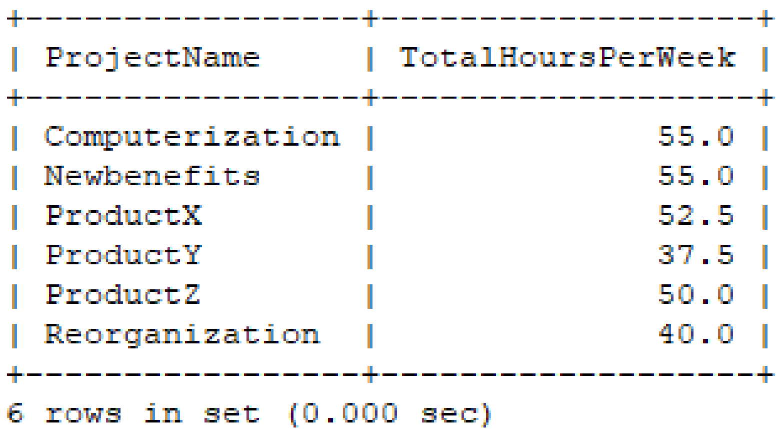
WONG_SSN <-- π_{Ssn} ($\sigma_{\text{Fname = 'Franklin' AND Lname = 'Wong'}}$ (EMPLOYEE))
WONG_EMPS <-- (EMPLOYEE) $\bowtie_{\text{(Super\_ssn), (Ssn)}}$(WONG_SSN)
RESULT <-- $\pi_{\text{Fname, Lname}}$ (WONG_EMPS)

Result:

Fname	Lname
John	Smith
Ramesh	Narayan
Joyce	English

d) **<u>For each project, list the project name and the total hours per week (by all employees) spent on that project</u>.**

PROJ_HOURS (Pno, TOT_HRS) <-- $\sigma_{\text{Pno , SUM HOURS}}$ (WORKS_ON)
RESULT <-- $\pi_{\text{Pname, TOT\_HRS}}$ ((PROJ_HOURS) $\bowtie_{\text{(Pno), (Pnumber)}}$ (PROJECT))



```
+-----------------+-------------------+
| ProjectName     | TotalHoursPerWeek |
+-----------------+-------------------+
| Computerization |              55.0 |
| Newbenefits     |              55.0 |
| ProductX        |              52.5 |
| ProductY        |              37.5 |
| ProductZ        |              50.0 |
| Reorganization  |              40.0 |
+-----------------+-------------------+
6 rows in set (0.000 sec)
```

e) **<u>Retrieve the names of all employees who work on every project</u>.**

PROJ_EMPS $_{\text{(Pno, Ssn)}}$ <-- $\pi_{\text{Pno, Essn}}$ (WORKS_ON)
ALL_PROJS $_{\text{(Pno)}}$ <-- π_{Pnumber} (PROJECT)
EMPS_ALL_PROJS <-- PROJ_EMPS ⋈ ALLPROJS (* DIVISION operation *)
RESULT <-- $\pi_{\text{Fname, Lname}}$ (EMPLOYEE * EMP_ALL_PROJS)

Result:

FNAME LNAME

f) **<u>Retrieve the names of all employees who do not work on any project</u>.**

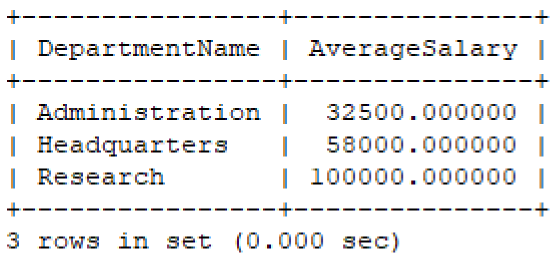
ALL_EMPS <-- π_{Ssn} (EMPLOYEE)
WORKING_EMPS $_{\text{(Ssn)}}$ <-- π_{Essn} (WORKS_ON)
NON_WORKING_EMPS <-- ALL_EMPS - WORKING_EMPS (*DIFFERENCE*)
RESULT <-- $\pi_{\text{Fname, Lname}}$ (EMPLOYEE * NON_WORKING_EMPS)

Result:

FNAME LNAME

g) **<u>For each department, retrieve the department name and the average salary of all employees working in that department</u>.**

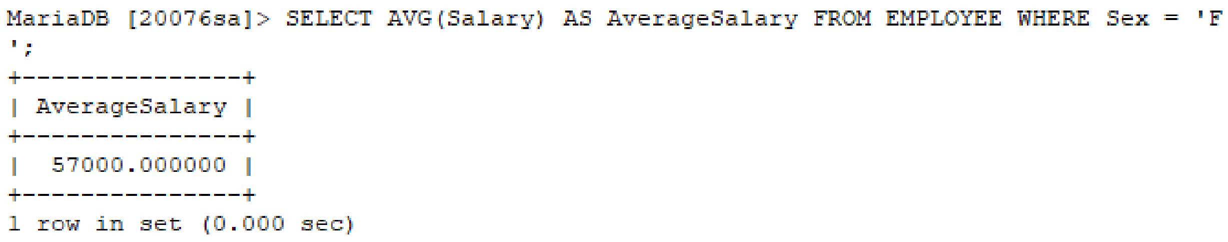
DEPT_AVG_SALS $_{\text{(Dnumber, AVG\_SAL)}}$ <-- σ_{Dno} $\bowtie$ $_{\text{AVG SALARY}}$ (EMPLOYEE)
RESULT <-- $\pi_{\text{Dnumber, AVG\_SAL}}$ (DEPT_AVG_SALS * DEPARTMENT)



```
+----------------+---------------+
| DepartmentName | AverageSalary |
+----------------+---------------+
| Administration |  32500.000000 |
| Headquarters   |  58000.000000 |
| Research       | 100000.000000 |
+----------------+---------------+
3 rows in set (0.000 sec)
```

h) **Retrieve the average salary of all female employees.**

RESULT $_{(\text{AVG\_F\_SAL})}$ <-- $\pi_{\text{AVG SALARY}}$ ($\sigma_{\text{SEX = 'F'}}$ (EMPLOYEE))



```
MariaDB [20076sa]> SELECT AVG(Salary) AS AverageSalary FROM EMPLOYEE WHERE Sex = 'F
';
+---------------+
| AverageSalary |
+---------------+
|  57000.000000 |
+---------------+
1 row in set (0.000 sec)
```

i) **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.**

EMP_PRO_HOUSE $_{(\text{ Ssn })}$ <-- π_{Essn} (WORKS_ON J$_{(\text{ Pno }),\ (\text{ Pnumber })}$ ($\sigma_{\text{PLocation = 'Houston'}}$ (PROJECT)))
DEP_NO_HOU <-- π_{Dnumber} (DEPARTMENT) - π_{Dnumber} ($\sigma_{\text{Dlocation = 'Houston'}}$ (DEPARTMENT))
EMP_DEP_NO_HOU <-- π_{Ssn} (EMPLOYEE $\bowtie_{(\text{ Pno }),\ (\text{ Dnumber })}$ (DEP_NO_HOU))
RESULT_EMPS <-- EMP_ PRO_HOUSE - EMP_DEP_NO _HOU (* this is set DIFFERENCE *)
RESULT <-- $\pi_{\text{Lname, Fname, Address}}$ (EMPLOYEE * RESULT_EMPS)

Result:

LNAME FNAME ADDRESS
Wallace Jennifer 291 Berry, Bellaire, TX

j) **List the last names of all department managers who have no dependents**

DEPT_MANGS $_{(\text{ Ssn })}$ <-- $\pi_{\text{Mgr\_ssn}}$ (DEPARTMENT)
EMPS_WITH_DEPENDENTS $_{(\text{ Ssn })}$ <-- π_{Essn} (DEPENDENT)
RESULT_EMPS <-- DEPT_MANGS - EMPS_WITH_DEPENDENTS
RESULT <-- π_{Lname} (EMPLOYEE * RESULT_EMPS)

Result:

LNAME
Borg

Exercise 8.18 Consider the LIBRARY relational database schema shown in Figure 8.14, which is used to keep track of books, borrowers, and book loans. Referential integrity constraints are shown as directed arcs in Figure 8.14, as in the notation of Figure 5.7. Write down relational expressions for the following queries:

a. How many copies of the book titled The Lost Tribe are owned by the library branch whose name is 'Sharpstown'?

B <-- BOOKCOPIES * LIBRARY_BRANCH * BOOK
RESULT <-- $\pi_{\text{No\_of\_copies}}$ ($\sigma_{\text{BranchName = 'Sharpstown' and Title = 'The Lost Tribe'}}$) B

b. How many copies of the book titled The Lost Tribe are owned by each library branch?

$\Pi_{\text{BranchId, No\_of\_copies}}$ (($\sigma_{\text{Title = 'The Lost Tribe'}}$ (BOOK)) * BOOKCOPIES)

c. Retrieve the names of all borrowers who do not have any books checked out.

NO_COUT_B <-- π_{CardNo} (BORROWER) - π_{CardNo} (BOOK_LOANS)
RESULT <-- π_{Name} (BORROWER * NO_ COUT_B)

d. For each book that is loaned out from the Sharpstown branch and whose Due_date is today, retrieve the book title, the borrower's name, and the borrower's address.

A <-- π_{BranchId} ($\sigma_{\text{BranchName = 'Sharpstown'}}$ (LIBRARY_BRANCH))
B_FROM_S <-- $\pi_{\text{BookId, CardNo}}$ (($\sigma_{\text{DueDate = 'today'}}$ (BOOKLOANS)) * A)
RESULT <-- $\pi_{\text{Title, Name, Address}}$ (BOOK * BORROWER * B_FROM_S)

e. **For each library branch, retrieve the branch name and the total number of books loaned out from that branch.**

S (BranchId, Total) <-- $_{\text{BranchId } \bowtie \text{ COUNT}}$ (BookId, CardNo) (BOOK_LOANS)
RESULT <-- $\pi_{\text{BranchName, Total}}$ (S * LIBRARY_BRANCH)

f. **Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.**

R (CardNo, TotalCheckout) <-- $_{\text{CardNo } \bowtie \text{ COUNT (BookId)}}$ (BOOK_LOANS)
R5 <-- $\sigma_{\text{TotalCheckout} > 5}$ (R)
RESULT <-- $\pi_{\text{Name, Address, TotalCheckout}}$ (B5 * BORROWER)

g. **For each book authored (or coauthored) by Stephen King, retrieve the title and the number of copies owned by the library branch whose name is Central.**

Steph (BookId, Title) <-- ($\sigma_{\text{AuthorName = 'Stephen King'}}$ (BOOK_AUTHORS)) * BOOK
CENTRAL (BranchId) <-- $\sigma_{\text{BranchName = 'Central'}}$ (LIBRARY_BRANCH)
RESULT <-- $\pi_{\text{Title, No\_of\_copies}}$ (Steph * BOOKCOPIES * CENTRAL)