Querying exercises from Company database

EMPLOYEE (ssn, fname, minit, lname, bdate, dno, gender, superssn)
Foreign Keys: dno REFERENCES department (dno),
Foreign Key: superssn REFERENCES employee (ssn)
DEPARTMENT (dno, dname, mgrssn, mgrstartdate)
Foreign Keys: mgrssn REFERENCES employee (ssn)
DEP_LOCATIONS (dno, dlocation)
Foreign Keys: dno REFERENCES department (dno),
PROJECT (pno, pname, plocation, dno)
Foreign Keys: dno REFERENCES department (dno),
WORKS_ON (essn, pno, hours)
Foreign Keys: essn REFERENCES employee (ssn)
Foreign Keys: pno REFERENCES project (pno)
DEPENDENT (essn, dep_name, gender, bdate date, relationship)
Foreign Keys: essn REFERENCES employee (ssn)

Queries based on Selection and Project operations:

(Q-01) List employees working for dno=4

$\sigma_{DNO = 4}$ (EMPLOYEE)

```
SELECT * FROM employee WHERE dno = 4;
```

(Q-02) List employees having salary > 30000

$\sigma_{SALARY > 30000}$ (EMPLOYEE)

```
SELECT * FROM employee WHERE salary> 30000;
```

(Q-03) List first name, last name, and salary of all employees

$\pi_{LNAME, FNAME, SALARY}$(EMPLOYEE)

```
SELECT fname, lname, salary FROM employee;
```

(Q-04) List first name, last name, and salary of employees that work for dno=5.

$\pi_{LNAME, FNAME, SALARY}(\sigma_{DNO=5}$ (EMPLOYEE))

```
SELECT fname, lname, salary FROM employee WHERE dno=5;
```

(Q-05) List employees having salary >= 10000 and <= 30000

$\sigma_{salary >= 10000 AND salary <= 30000}$(EMPLOYEE)

```
SELECT * FROM employee WHERE salary >= 10000 AND salary <= 30000;
```

(Q-06) List (fname, dno, salary) of employees having salary >= 30000

$r1 \leftarrow \sigma_{salary >= 30000}(EMPLOYEE)$
$result \leftarrow \pi_{fname, dno, salary}(r1)$
OR
$result \leftarrow \pi_{fname, dno, salary}(\sigma_{salary >= 30000}(EMPLOYEE))$

```
SELECT fname, dno, salary FROM employee WHERE salary >= 30000;
```

(Q-07) List Fname, Salary of all Female Employees

$\pi_{fname, salary}(\sigma_{GENDER='F'}(EMPLOYEE))$

```
SELECT fname, salary FROM employee WHERE gender='F';
```

(Q-08) List employees either working dno=4 and salary > 25000 or working dno=5 and salary > 30000.

$\sigma_{(DNO=4 \text{ AND } SALARY>25000) \text{ OR } (DNO=5 \text{ AND } SALARY > 30000)}(EMPLOYEE)$

```
SELECT * FROM employee WHERE (dno=4 AND salary > 25000) OR
    (dno=5 AND salary >= 30000);
```

(Q-09) List all employees supervised by employee having ssn = 123

$\sigma_{(superssn=123)}(EMPLOYEE)$

```
SELECT * FROM employee WHERE superssn=123;
```

Queries based on **JOIN** operations:

(Q-10) List Fname, Dname of all employees

$r1 \leftarrow EMPLOYEE \bowtie_{employee.dno=department.dno} DEPARTMENT$
$result \leftarrow \pi_{fname, dname}(r1)$

```
SELECT fname, dname FROM employee AS e
    JOIN department AS d ON (e.dno=d.dno);
```

(Q-11) List Fname, Salary of all Female Supervisors

$s \leftarrow \pi_{SUPERSSN}(EMPLOYEE)$
$r1 \leftarrow s \bowtie_{s.superssn=ssn} EMPLOYEE$
$r2 \leftarrow \sigma_{GENDER='F'}(r1)$
$result \leftarrow \pi_{fname, salary}(r2)$

```
SELECT fname, salary FROM employee AS e JOIN
    (SELECT DISTINCT superssn FROM employee) AS r1
    ON (r1.superssn=e.ssn)
    WHERE gender='F';
```

(Q-12) List Fname, Salary of all Female Managers

$r1 \leftarrow EMPLOYEE \bowtie_{employee.ssn=department.mgrssn} DEPARTMENT$
$result(dname, mgr\_name) \leftarrow \pi_{dname,\ fname}(\sigma_{GENDER='F'}(r1))$

```
SELECT dname, fname AS mgr_name FROM employee AS e
    JOIN department AS d ON (e.ssn=d.mgrssn)
    WHERE gender='F';
```

(Q-13) List Employee that are working on projects monitored by DNO=4

$\pi_{EMPLOYEE.*}((\sigma_{DNO=4}(PROJECT)) * WORKS\_ON) * EMPLOYEE)$

```
SELECT employee.* FROM project NATURAL JOIN works_on
    JOIN employee ON(ssn=essn) WHERE project.dno=4
```

(Q-14) List Employee Name, SupervisorName (if any)

$\pi_{e.fname,s.fname}(EMPLOYEE\ e\ LEFT\ JOIN\ EMPLOYEE\ s\ ON\ (e.superssn=s.ssn))$

```
SELECT e.fname, s.fname FROM employee AS e LEFT JOIN employee AS s
    ON(e.superssn=s.essn)
```

(Q-15) List employees(ssn, fname) that work on projects managed by department 'Research'.

$r1 \leftarrow \sigma_{DNAME='Research'}(DEPARTMENT)$
$r2 \leftarrow r1 * PROJECT * WORKS\_ON$
$r3 \leftarrow r2 \bowtie_{ESSN=SSN} EMPLOYEE$
$RESULT \leftarrow \pi_{SSN,\ FNAME}(r3)$

```
SELECT ssn, fname FROM (
(SELECT * FROM department WHERE dname = 'Research') AS r1
    NATURAL JOIN project NATURAL JOIN works_on) AS r2
            JOIN employee AS e ON (ssn=essn);
```

Queries based on **SET** operations:

(Q-16) List ssn of non-managers, i.e. they are not manager of any department

$\pi_{SSN}(EMPLOYEE)\ EXCEPT\ \pi_{MGRSSN}(DEPARTMENT)$

```
SELECT ssn FROM employee
EXCEPT
SELECT DISTINCT mgrssn FROM department;
```

(Q-17) List ssn, and name of non-managers

NM ← π_{SSN}(EMPLOYEE) EXCEPT π_{MGRSSN}(DEPARTMENT)
RES ← $\pi_{SSN,FNAME}$(EMPLOYEE * NM)

```
SELECT ssn, fname FROM employee
    NATURAL JOIN
(SELECT ssn FROM employee EXCEPT SELECT mgrssn FROM department) AS nm;
```

OR using Semi DIFFERENCE

```
SELECT ssn, fname FROM employee
    WHERE ssn NOT IN (SELECT mgrssn FROM department);
```

*Note: operator * between two relations here, represent Natural Join.*

(Q-18) List SSN, FNAME of employees who do not work on any project.

NW ← π_{SSN}(EMPLOYEE) EXCEPT π_{ESSN}(WORKS_ON)
RES ← $\pi_{SSN,FNAME}$(EMPLOYEE * NW)

```
SELECT ssn, fname FROM employee
    NATURAL JOIN
(SELECT ssn FROM employee EXCEPT SELECT essn FROM works_on) AS nw;
```

OR using Semi DIFFERENCE

```
SELECT ssn, fname FROM employee
    WHERE ssn NOT IN (SELECT essn FROM works_on); //SEMI-DIFFERENCE
```

(Q-19) Find out Employees that either work for DNO=4 or associated with a department as manager

e1 ← $\sigma_{DNO=4}$(EMPLOYEE)
e2 ← $\pi_{EMPLOYEE.*}$($\sigma_{department.dno=4}$(EMPLOYEE * DEPARTMENT))
result ← e1 UNION e2

OR

```
SELECT * FROM employee WHERE dno=4
UNION
SELECT employee.* FROM employee NATURAL JOIN department
    WHERE department.dno=4;
```

(Q-20) List SSN, FNAME of employees who work on at least one project.

EW ← π_{SSN}(EMPLOYEE) INTERSECT π_{ESSN}(WORKS_ON)
result ← $\pi_{SSN,FNAME}$(EMPLOYEE * EW)

```
SELECT ssn, fname FROM employee
    NATURAL JOIN
(SELECT ssn FROM employee INTERSECT SELECT essn FROM works_on) AS ew;
```

```
SELECT ssn, fname FROM employee
    WHERE ssn IN (SELECT essn FROM works_on); //SEMI-JOIN or SEMI-INTERSECT
```