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 **Aggregate** operations:

(Q-01) What is total salary company pays? Also give count of employees, maximum, minimum, average salary the company pays to its employees.

$F_{COUNT(SSN),\ SUM(SALARY),\ MAX(SALARY),\ MIN(SALARY),\ AVG(SALARY)}$ (EMPLOYEE)

```
SELECT count(ssn), sum(salary), max(salary), min(salary), avg(salary) FROM employee;
```

(Q-02) Give department wise sum of salary. Also give count of employees, maximum, minimum, average salary that each department of the company pays.

$_{DNO}\ F_{COUNT(SSN),\ SUM(SALARY),\ MAX(SALARY),\ MIN(SALARY),\ AVG(SALARY)}$ (EMPLOYEE)

```
SELECT dno, count(ssn), sum(salary), max(salary), min(salary), avg(salary)
    FROM employee GROUP BY dno;
```

May also give names to aggregated columns -

r1(dno, no_emps, total_salary, max_salary, min_salary, avg_salary)
← $_{DNO}\ F_{COUNT(SSN),\ SUM(SALARY),\ MAX(SALARY),\ MIN(SALARY),\ AVG(SALARY)}$ (EMPLOYEE)

```
SELECT dno, count(ssn) AS no_emps, sum(salary) AS total_salary, max (salary) AS  max_salary,
min(salary) AS  min_salary, avg(salary) AS avg_salary
    FROM employee GROUP BY dno;
```

(Q-03) List DNO, Department Name, and No of Employees for each department of the company.

r1(dno, no_emps) ← $_{DNO}\ F_{COUNT(SSN)}$ (EMPLOYEE)
r2 ← r1 * DEPARTMENT
result ← $\pi_{DNO,\ DNAME,\ NO\_EMPS}$(r2)

```
SELECT dno, dname, no_emps FROM (SELECT dno, count(ssn) AS no_emps FROM employee
GROUP BY dno) AS r1 NATURAL JOIN department;
```

(Q-04) List DNO, Department Name, Manager Name, and Number of Employees for each department of the company.

$r1(dno, no\_emps) \leftarrow {}_{DNO}\mathcal{F}_{COUNT(SSN)}(EMPLOYEE)$
$r2 \leftarrow r1 * DEPARTMENT$
$r3 \leftarrow r2 \bowtie_{MGRSSN=SSN} EMPLOYEE$
$result \leftarrow \pi_{DNO, DNAME, FNAME, NO\_EMPS}(r3)$

```
SELECT r1.dno, dname, no_emps, fname AS manager_name FROM (SELECT dno, count(ssn)
    AS no_emps FROM employee GROUP BY dno) AS r1 NATURAL JOIN department
    JOIN employee ON (mgrssn=ssn);
```

(Q-05) List employee-ssn along with count of employees they are supervising for employees who are supervising more than 2 employees

$r1(superssn, no\_emps) \leftarrow {}_{SUPERSSN}\mathcal{F}_{COUNT(SSN)}(EMPLOYEE)$
$result \leftarrow \sigma_{NO\_EMPS > 2}(r1)$

```
SELECT superssn, count(ssn) FROM employee
    GROUP BY superssn
    HAVING count(ssn) > 2;
```

(Q-06) List DNO, Department Name, Number of Employees, and No of Projects it controls for each department of the company.

$r1(dno, no\_emps) \leftarrow {}_{DNO}\mathcal{F}_{COUNT(SSN)}(EMPLOYEE)$
$r2(dno, no\_projs) \leftarrow {}_{DNO}\mathcal{F}_{COUNT(PNO)}(PROJECTS)$
$r3 \leftarrow r1 * r2 * DEPARTMENT$
$result \leftarrow \pi_{DNO, DNAME, NO\_EMPS, NO\_PROJS}(r3)$

```
SELECT dno, dname, no_emps, no_projs FROM (SELECT dno, count(ssn) AS no_emps FROM
employee GROUP BY dno) AS r1 NATURAL JOIN (SELECT dno, count(pno) AS no_projs FROM
project GROUP BY dno) AS r2 NATURAL JOIN department;
```

(Q-07) List SSN, Name, SALARY, Department Name, and Number of Employees supervising for each employee of the company.

$r1(superssn, no\_emps) \leftarrow {}_{SUPERSSN}\mathcal{F}_{COUNT(SSN)}(EMPLOYEE)$
$r2 \leftarrow EMPLOYEE \text{ LEFT JOIN}_{E.SSN = r1.SUPERSSN} (r1)$
$r3 \leftarrow r2 * DEPARTMENT$
$result \leftarrow \pi_{E.SSN, FNAME, SALARY, DNAME, NO\_EMPS}(r3)$

```
SELECT e.ssn, fname, salary, dname, no_emps FROM (SELECT superssn, count(ssn) AS no_emps
FROM employee GROUP BY superssn) AS r1 RIGHT JOIN employee AS e ON (r1.superssn=e.ssn)
NATURAL JOIN department;
```