

Querying exercises from Company Database

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

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 -

$r1(dno, no_emps, total_salary, max_salary, min_salary, avg_salary)$

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

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.

$r1(dno, no_emps) \leftarrow DNO \ F_{COUNT(SSN)} (EMPLOYEE)$

$r2 \leftarrow r1 * DEPARTMENT$

result $\leftarrow \pi_{DNO, DNAME, NO_EMPS}(r2)$

```
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) ←DNO FCOUNT(SSN)(EMPLOYEE)
r2 ← r1 * DEPARTMENT
r3 ← r2 ⋈MGRSSN=SSN EMPLOYEE
result ← πDNO, 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) ←SUPERSSN FCOUNT(SSN)(EMPLOYEE)
result ← σNO_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) ←DNO FCOUNT(SSN)(EMPLOYEE)
r2(dno, no_projs) ←DNO FCOUNT(PNO)(PROJECTS)
r3 ← r1 * r2 * DEPARTMENT
result ← πDNO, 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) ←SUPERSSN FCOUNT(SSN)(EMPLOYEE)
r2 ← EMPLOYEE LEFT JOINE.SSN = r1.SUPERSSN (r1)
r3 ← r2 * DEPARTMENT
result ← πE.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;
```