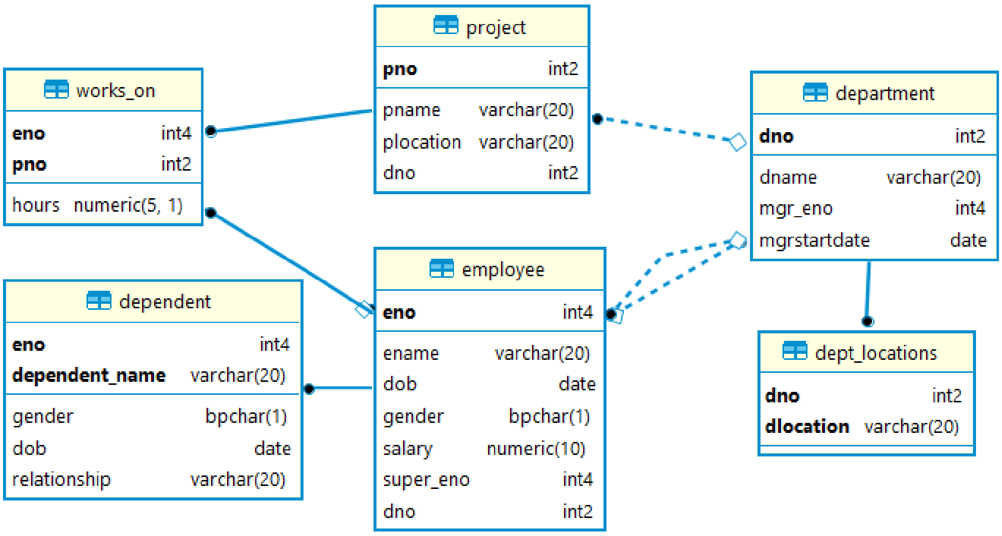
Querying exercises from Company3



Queries based on SET operations:

1. List Eno of non-managers, i.e., they are not manager of any department

```
select eno, ename, salary from employee where dno=4
union
select eno, ename, salary from employee where dno=5;
```

OR

```
select eno, ename, salary from employee where dno=4 or dno=5;
```

2. List Eno of non-managers, i.e., they are not manager of any department

$\pi_{ENO}(\text{EMPLOYEE})$ EXCEPT $\pi_{MGR\_ENO}(\text{DEPARTMENT})$

```
select eno from employee
except
select mgr_eno from department;
```

3. List Eno, Ename, Salary of non-managers, i.e. they are not manager of any department

$r1 \leftarrow \pi_{SSN}(EMPLOYEE)$ EXCEPT $\pi_{MGRSSN}(DEPARTMENT)$

$result \leftarrow \pi_{eno,\ ename,\ salary}(r * EMPLOYEE)$

```
select eno, ename, salary from employee
natural join
(select eno from employee
except
select mgr_eno from department) as r;
```

4. List SSN, FNAME of employees who do not work on any project.

$$r_1 \leftarrow \underline{\pi_{eno}(Emp) - \pi_{eno}(Works\_on)}$$

$$\pi_{eno,\ ename}(r_1 * EMP)$$

```
SELECT eno, ename FROM EMPLOYEE natural join
(select eno from employee
 except
 select eno from works_on
) as r;
```

OR

$$EMP \underset{eno}{\overset{SEMI-DIFFERNCE}{\text{———}}} WORKS\_ON$$

this means difference is computed such that eno of emp does not exist in eno of Works_ON

```
select eno, ename from employee
where eno NOT IN (select eno from works_on);
```

5. Find out Employees (eno, ename, salary) that are directly (works for) for dno=4 or indirectly associated with (work on projects managed by) dno=4

$$r_1 \leftarrow \pi_{eno}(\sigma_{dno=4}(EMP))$$

$$r_2 \leftarrow \pi_{eno}(\sigma_{dno=4}(WORKS\_ON * PROJ))$$

$$r \leftarrow r_1 \cup r_2$$

$$\pi_{eno,\, ename,\, salary}(r * EMP)$$

```
select eno, ename, salary from employee
natural join
(select eno from employee where dno=4
UNION
select eno from works_on natural join project
where dno=4) as r;
```

6. List SSN, FNAME of employees who work on at least one project.

$$r_1 \leftarrow \pi_{eno}(EMP) \cap \pi_{eno}(WORKS\_ON)$$

$$\pi_{eno,\, ename,\, salary}(r * EMP)$$

```
select eno, ename, dno, salary from employee
natural join
(select eno from employee
INTERSECT
select eno from works_on ) as r;
```

OR

$$EMP \ \frac{\text{SEMI INTERSECTION}}{\text{eno}} \ \text{WORKS-ON}$$

i.e; eno of emp exists IN eno of Works_ON

```
select eno, ename, salary from employee
where eno IN (select eno from works_on );
```

OR

$$\pi_{eno,\, ename,\, salary}(EMP * WORK\text{-}ON)$$

```
select distinct eno, ename, salary from employee
natural join works_on;
```

EQUI or NATURAL JOIN is also basically intersection only; JOIN happens on “matched” tuples. Matching here checks only for specified attributes in join condition. However non set operation do not remove the duplicates there for require using DISTINCT here.

7. Compute employees (`eno, ename, salary`) that are supervisors (supervises some employees) but not managers (that are manager of any department)

$$r_1 \leftarrow \pi_{super\text{-}eno}(EMP) - \pi_{mgr\text{-}eno}(DEP)$$

$$result \leftarrow \pi_{eno,\, ename,\, salary}(r_1 * EMP)$$

```
select distinct eno, ename, salary from employee
natural join
(select super_eno as eno from employee e
except
select mgr_eno from department) as r;
```

8. Compute employees (`eno, ename, salary`) that are supervisors or Managers but not both

$$r_2 \leftarrow \pi_{super\_eno}(Emp)$$

$$r_3 \leftarrow \pi_{mgr\_eno}(Dept)$$

$$\boxed{r_2 \cup r_3 - (r_2 \cap r_3)}$$

```
select eno, ename, salary from employee
join
((SELECT super_eno from employee
UNION
SELECT mgr_eno from department)
except
(SELECT super_eno from employee
intersect
SELECT mgr_eno from department)) as r
on (r.super_eno = eno);
```