1. Tampilkan jumlah pegawai dari masing-masing departemen.

```
(SELECT D.DName as nama_departemen, COUNT(*) AS
jumlah_pegawai
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.Dno = D.Dnumber
GROUP BY D.DName)

UNION
(SELECT D.DName as nama_departemen, 0 AS jumlah_pegawai
FROM EMPLOYEE E, DEPARTMENT D
WHERE D.Dnumber not in (SELECT D.Dnumber FROM EMPLOYEE E,
DEPARTMENT D WHERE E.Dno = D.Dnumber)
GROUP BY D.DName);
```

2. Tampilkan nama employee yang bekerja sebagai manager dan juga memiliki anak perempuan

```
SELECT E.FName, E.LName
FROM EMPLOYEE E, DEPENDENT DEPEND, DEPARTMENT DEPART
WHERE E.SSn = DEPEND.Essn AND E.Ssn = DEPART.mgr_ssn AND
DEPEND.sex = 'F';
```

3. Tampilkan semua nama employee beserta salary dan nama departemennya. Jika employee tersebut memiliki dependent, tampilkan nama dependennya dan relasinya dengan employee

```
(SELECT E.Fname, E.LName, E.Salary, D.Dname AS
nama_departemen, '-' AS dependent_name, '-' AS relationship
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.Dno = D.Dnumber AND E.Ssn not in (SELECT
DEPEND.Essn FROM DEPENDENT DEPEND))

UNION
```

```
(SELECT E.FName, E.LName, E.SALARY, DEPART.DName AS
nama_departemen, STRING_AGG(DEPEND.Dependent_name, ', ') as
dependent_name, STRING_AGG(DEPEND.Relationship, ', ') as
relationship
FROM EMPLOYEE E, DEPARTMENT DEPART, DEPENDENT DEPEND
WHERE E.Dno = DEPART.Dnumber AND E.SSn = DEPEND.Essn
GROUP BY E.FName, E.LName, E.Salary, DEPART.DName);
```

4. Tampilkan semua nama employee yang memiliki gaji di atas rata rata gaji semua pegawai (catatan: gunakan EXISTS untuk query soal ini)

   Tidak bisa pakai exists?

```
SELECT E.FName, E.LName
FROM EMPLOYEE E
WHERE EXISTS (SELECT *
FROM EMPLOYEE
) AND E.SALARY > (SELECT AVG(E.SALARY) FROM EMPLOYEE E);
```

5. Tampilkan nama dari semua pegawai yang menjadi supervisor pegawai lainnya

```
SELECT DISTINCT E.FName, E.Lname
FROM EMPLOYEE E
WHERE E.SSN in (SELECT DISTINCT E.SUPER_Ssn FROM EMPLOYEE
E);
```

6. Tampilkan gaji (sebelum dan sesudah kenaikan) dari masing-masing employee jika untuk tiap jam dari proyek, employee mendapat tambahan gaji $1000/jam

```
(SELECT E.Fname, E.LName, E.Salary as gaji_sebelum,
SUM(E.salary + WO.hours*1000) AS gaji_sesudah
FROM EMPLOYEE E, WORKS_ON WO
WHERE E.SSn = WO.Essn
GROUP BY E.Fname, E.LName, E.Salary)

UNION

(SELECT E.Fname, E.LName, E.Salary as gaji_sebelum,
E.Salary as gaji_setelah
FROM EMPLOYEE E, WORKS_ON WO
WHERE E.Ssn not in (SELECT WO.Essn FROM WORKS_ON WO)
GROUP BY E.Fname, E.LName, E.Salary);
```

7. Untuk masing-masing employee yang memiliki lebih dari 1 proyek, tampilkan nama employee, ssn employee, dan jumlah proyek yang dikerjakan oleh masing-masing employee.

```
SELECT E.Fname, E.Lname, E.ssn, COUNT(WO.Pno)
FROM EMPLOYEE E, WORKS_ON WO
WHERE E.Ssn = WO.Essn
GROUP BY E.Fname, E.Lname, E.SSN
HAVING COUNT(WO.Pno) > 1;
```

8. Buat table view yang berisi data employee yang ditambah dengan informasi nama proyek dan lama jamnya yang sedang dikerjakan tiap employee beserta departemen tiap employee.

```
CREATE VIEW employee_new AS
SELECT E.*, P.PName, SUM(WO.Hours) AS lama_bekerja, D.Dname
FROM EMPLOYEE E, PROJECT P, WORKS_ON WO, DEPARTMENT D
WHERE E.Dno = D.DNumber AND E.Ssn = WO.Essn AND WO.Pno =
P.Pnumber
GROUP BY E.Fname, E.LName, E.Minit, E.Ssn, E.Bdate,
E.address, E.sex, E.Salary, E.super_ssn, E.dno , P.Pname,
D.Dname;
```

9. Dari tabel view yang sudah dibuat, tampilkan “employee_fname”, “employee_lname”, “proyek”, “departemen”

```
SELECT Fname, Lname, Pname, Dname
FROM employee_new;
```

10. Dari tabel view yang sudah dibuat, tampilkan “employee_fname”, “employee_lname” diikuti total durasi jam semua proyek yang dikerjakan masing-masing employee

```
SELECT Fname, Lname, Pname, lama_bekerja
FROM employee_new;
```