Consider the following company dataset, where Emp.mgrno is a foreign key for Emp(eno) used to model the management chain in the company, Work.empno is a foreign key for Emp.eno, and Work.deptno is a foreign key for Dept.dno. Assume that salaries and budgets are given in dollars (per year) and ages are given in years. Write each of the queries that follow in SQL.

**Emp**(eno, ename, salary, age, mgrno)  -- the usual info about employees

**Work**(empno, deptno, pcttime)  -- emps can split their time between depts

**Dept**(dno, dname, budget)  -- the usual info about departments

1. **(2 points)** Print the employee numbers and names of employees who earn more per year than their manager does:

   ```sql
   SELECT e.eno, e.ename
   FROM Emp e, Emp m
   WHERE e.mgrno = m.eno
   AND e.salary > m.salary
   ```

2. **(3 points)** Print the names of employees who spend at least some part of their time working in all of the departments that have a budget over $1M.

   ```sql
   SELECT e.ename  -- Select emp names where ...
   FROM Emp e
   WHERE NOT EXISTS (  -- ... there doesn't exist a $1M+ department where ...
      SELECT d.dno
      FROM Dept d
      WHERE d.budget > 1000000
   )

   AND NOT EXISTS (  -- ... this emp doesn't work for that dept! (:-))
      SELECT *
      FROM Work w
      WHERE w.deptno = d.dno
      AND w.empno = e.eno )
   ```
Q: Why this one is incorrect?

```
SELECT e.eno, e.ename
FROM Emp e
WHERE NOT EXISTS (
    SELECT * FROM Work w, Dept d
    WHERE w.empno = e.eno
    AND w.deptno = d.dno
    AND d.budget <= 1000000);
```

3. (2 points) Print a list of the employee numbers, employee names, percent times, department numbers, and department names of all employees. There may be some employees who aren’t assigned to any department at the moment; be sure to include them on the list as well.

```
SELECT e.eno, e.ename, w.pcttime, w.deptno, d.dname
FROM Emp e
LEFT OUTER JOIN Work w ON e.eno = w.empno
LEFT OUTER JOIN Dept d ON w.deptno = d.dno;
```

4. (3 points) Create an employment percentage view that, for each different percent-time value (pcttime), lists the percent-time together with the average age and the low and high salaries for employees who work for some department for that percentage of their time.

```
CREATE VIEW employment_percentage
    (pcttime, avg_age, low_salary, high_salary)
AS
SELECT w.pcttime, avg(e.age), min(e.salary), max(e.salary)
FROM Work w, Emp e
WHERE w.empno = e.eno
GROUP BY w.pcttime;
```