employee (*ID*, *person_name, street, city*)
company (*company_name, city*)
works (ID, *company_name, salary*)
manages (ID, manager_id)

Figure 1

Consider the database in Figure 1, where the primary keys are underlined. The *manages* relation describes the manager (manager_id) of a certain employee (ID). Each manager is also an employee himself (herself). Construct the following SQL queries for this relational database.

1. Find the ID, name, and manager of each employee who works for “FirstBank”.

```sql
-- 1.   Find the ID, name, and manager of each employee who works for “FirstBank”.
select e.id as ID, e.person_name as name, m.person_name as manager from employee as e
inner join works as w on w.id = e.id
left join manages as mid on mid.id = e.id
left join employee as m on m.id = mid.manager_id
where w.company_name = 'FirstBank';
```

想法是從 employee 拿到名字，利用 ID 對應到 manages 拿到上司 ID 再對回去 employee 拿到上司名字，另外再對應題目提到的限制 company_name=’FirstBank’。第三行本人覺得要用 inner，想法是最終關注有再 FirstBank 上班的人，而人可能沒有工作。

2. Find the ID of each employee who does not work for “FirstBank”.

```sql
-- 2.   Find the ID of each employee who does not work for “FirstBank”.
select e.id as ID from employee as e
inner join works as w on w.id = e.id
where w.company_name <> 'FirstBank';
```

```sql
select w.id as ID from works as w
where w.company_name <> 'FirstBank';  2ms
```

此題只要求找ID，因此可以不需要employee，只需要對work限制公司名 (解二)。至於出現解一，只是單純第一次沒想那麼多就寫了： P

3. Find the ID and name of each employee who lives in the same city as the location of the company for which the employee works.

```
17  select e.id as ID, e.person_name as name from employee as e
18  inner join company as c on c.city = e.city
19  inner join works as w on w.id = e.id
20  where w.company_name = c.company_name;  4ms
```

需要ID, name因此需要employee。先不管是否與工作地點相同，與居住城市相同的公司inner join進來，再從works考慮是否工作與居住城市相同(使用where條件篩選)。

4. Find the ID of each employee who earns more than at least one employee of “SmallBank”.

(1) Please use “tuple variable”.

```
23  -- a. (1)   Please use “tuple variable”.
24  select distinct w_other.id as ID from works as w_other, works as w_smb
25  where w_other.salary>w_smb.salary and w_smb.company_name='SmallBank';
26  -- 特別：自己公司SmallBank內互比，賺的比其中一人多也算(ID = 10003)
```

可以看到第24行最後引用兩相同來源表格(tuple variable)，且題目沒註明同公司內不能互比，可能存在通公司AB員工互相比較也算在結果內。使用distinct是因為如果SmallBank員工不只一位，假設我賺的比所有SmallBank員工多，我會出現SmallBank員工數量的次數。

(2) Please use “nested subquery” in the WHERE clause.

```
29  select w.id as ID from works as w
30  where w.salary > (
31      select distinct min(sb.salary) from works as sb
32      where sb.company_name='SmallBank'
33  );  2ms
```

```
35  select w.id as ID from works as w
36  where w.salary > some(
37      select sb.salary from works as sb
38      where sb.company_name='SmallBank'
39  );  2ms
```

可以從where裡明顯看到包了一層select (nested subquery)，這裡引用min是因為內層select可能回傳多個值(我賺的比SmallBank員工多，且人數>=2)，因此取distinct min，依照題目要求，比一人多即可(比最低的高就好)。解二使用some使存在一筆資料符合就算。

Find the name of each company whose employees earn a higher salary, on average, than the average salary at “FirstBank”.

(1) Please use “having”

```sql
select w.company_name as company from works as w
group by w.company_name
having avg(w.salary) > (
    select avg(fb.salary) from works as fb
    where fb.company_name = 'FirstBank'
);
```

這題要取同公司名薪水的avg，因此必要的是group by company_name，第二個select意在取題目指名的FirstBank平均。

(2) Please use “with”.

```sql
with cte as (
    select w.company_name as company_name, avg(w.salary) as savg from works as w
    where w.company_name <> 'FirstBank'
    group by w.company_name
), cte2 as (
    select avg(w.salary) as savg from works as w
    where w.company_name = 'FirstBank'
    group by w.company_name
)
select c.company_name as company_name from cte as c
cross join cte2 as c2
where c.savg > c2.savg;  2ms
```

```sql
with cte as (
    select avg(fb.salary) as avgfb from works as fb
    where fb.company_name = 'FirstBank'
)
select w.company_name as company from works as w
group by w.company_name
having avg(w.salary) > (
    select avgfb from cte
);
```

```sql
with cte as (
    select fb.salary as avgfb from works as fb
    where fb.company_name = 'FirstBank'
)
select w.company_name as company from works as w, cte
group by w.company_name
having avg(w.salary) > avg(cte.avgfb);  3ms
```

這裡提供兩個做法，可以看到第一種偏向暴力，個例處理，而使用cross join在這裡也不會使表格列數變多(cte2只有一列)，最後在比較的只有cte列數。第二種單純只是在with內做好需要的avg，需要的時候select。解三在with不使用avg，為符合having必用aggregate function。

5. Delete all tuples in the *works* relation for employees of “SmallBank”.

```
69  delete from works as w
70  where w.company_name = 'SmallBank';
```

單純選定一個要delete的表格works，刪除題目要求的「與SmallBank有關的列」。

6. Add a new employee with the ID as “E01” and the name as “John”, but the address is currently unknown.

```
73  insert into employee values ('E01', 'John', null, null);
```

與題目要求的相同，於employee增加ID=E01, name=John, address=unknown者，未知者填入null(create table不能在address限制not null)。

7. Give each employee of “FirstBank” a 10-percent raise of salaries unless the salary becomes greater than $100000; in such cases, give only a 3-percent raise.

```
76  update works as w
77  set w.salary =
78      case
79          when w.salary * 1.10 <= 100000 then w.salary*1.1
80          else w.salary*1.03
81      end
82  where w.company_name = 'FirstBank'; AffectedRows: 4  8ms
```

此題指對存在的資料更新，因此使用update，對於加薪10%<=十萬者就*1.10，否則只有*1.03，且題意限定公司為FirstBank。

Note:

1. Please submit your homework in a single PDF file to Tronclass before 2025/10/22 23:59 (星期三)。

2. 解答2025/10/23公布。We do NOT accept late submission for this homework.