

# DB - HW3

3.8

Consider the bank database of Figure 3.18, where the primary keys are underlined. Construct the following SQL queries for this relational database.

1. Find the ID of each customer of the bank who has an account but not a loan.
2. Find the ID of each customer who lives on the same street and in the same city as customer '12345'.
3. Find the name of each branch that has at least one customer who has account in the bank and who lives in "Harrison".

Answer:

1.

```
SELECT ID
FROM   depositor
EXCEPT (SELECT ID
          FROM   borrower);
```

2.

```
SELECT S.ID
FROM   customer AS S JOIN customer AS C USING (customer_street, customer_city)
WHERE  C.ID = '12345';
```

3.

```
SELECT DISTINCT branch_name
FROM   account NATURAL JOIN depositor NATURAL JOIN customer
WHERE  customer_city = 'Harrison';
```

3.9

Consider the relational database of Figure 3.19, where the primary keys are underlined. Give an expression in SQL for each of the following queries.

1. Find the ID, name and city of residence of each employee who works for "First Bank Corporation".
2. Find the ID, name and city of residence of each employee who works for "First Bank Corporation" and earns more than \$10000.
3. Find the ID of each employee who does not work for "First Bank Corporation".
4. Find the ID of each employee who earns more than every employee of "Small Bank Corporation".
5. Assume that companies may be located in several cities. Find the name of each company that is located in every city with "Small Bank Corporation" is located.
6. Find the name of the company that has the most employees (or companies, in the case where there is a tie for the most).
7. Find the name of each company whose employees earn a higher salary, on average, than the average salary at "First Bank Corporation".

Answer:

1.

```
SELECT ID, person_name, city
FROM   employee, works
WHERE  works.company_name = 'First Bank Corporation'
AND    works.employee_name = employee.employee_name;
```

2.

```
SELECT ID, person_name, city
FROM   employee
WHERE  ID
IN     (SELECT ID
        FROM   works
        WHERE  company_name = 'First Bank Corporation'
        AND    salary      > 10000);
```

3.

```
SELECT ID
FROM   works
WHERE  company_name <> 'First Bank Corporation';
```

4.

```
WITH   e_total_salary
AS     (SELECT   ID, SUM(salary) AS total_salary
        FROM     works
        GROUP BY ID)
SELECT ID
FROM   e_total_salary
WHERE  total_salary > ALL
      (SELECT total_salary
        FROM   e_total_salary, works
        WHERE  works.company_name <> 'Small Bank Corporation'
        AND    e_total_salary.ID = works.ID);
```

5.

```
SELECT C.company_name
FROM   company AS C
WHERE  NOT EXISTS
      ((SELECT city
        FROM   company
        WHERE  company_name = 'Small Bank Corporation')
       EXCEPT
       (SELECT city
        FROM   company T
        WHERE  C.company_name = T.company_name));
```

6.

```

SELECT    company_name
FROM      works
GROUP BY  company_name
HAVING    COUNT(DISTINCT ID) >= ALL
          (SELECT COUNT(DISTINCT ID)
           FROM    works
           GROUP BY company_name);

```

7.

```

SELECT    company_name
FROM      works
GROUP BY  company_name
HAVING    AVG(salary) >
          (SELECT AVG(salary)
           FROM    works
           WHERE   company_name = 'First Bank Corporation');

```

3.10

Consider the relational database of Figure 3.19. Give an expression in SQL for each of the following:

1. Modify the database so that the employee whose ID is '12345' now lives in "Newtown".
2. Give each manager of "First Bank Corporation" a 10 percent raise unless the salary becomes greater than \$100000; in such cases, give only a 3 percent raise.

Answer:

1.

```

UPDATE employee
SET    city = 'Newtown'
WHERE  ID = '12345';

```

2.

```

UPDATE works
SET    salary = salary * 1.1
WHERE  ID IN
      (SELECT ID
       FROM  manages)
AND    company_name = 'First Bank Corporation'
AND    salary * 1.1 <= 100000;

UPDATE works
SET    salary = salary * 1.03
WHERE  ID IN
      (SELECT ID
       FROM  manages)
AND    company_name = 'First Bank Corporation'
AND    salary * 1.1 > 100000;

```

3.15

Consider the bank database of Figure 3.18, where the primary keys are underlined. Construct the following SQL queries for the relational database.

1. Find each customer who has an account at *every* branch located in "Brooklyn".

2. Find the total sum of all loan amounts in the bank.

3. Find the names of all branches that have assets greater than those of at least one branch located in “Brooklyn”.

Answer:

1.

```
SELECT C.customer_name
FROM   depositor AS E, customer AS C
WHERE  NOT EXISTS
      ((SELECT branch_name
        FROM   branch
        WHERE  branch_city = 'Brooklyn')
      EXCEPT
      (SELECT A.branch_name
        FROM   depositor AS D, account AS A
        WHERE  D.account_number = A.account_number
        AND    E.ID              = D.ID))
AND    E.ID = C.ID;
```

2.

```
SELECT SUM(amount)
FROM   loan;
```

3.

```
SELECT branch_name
FROM   branch
WHERE  assets > SOME
      (SELECT assets
        FROM   branch
        WHERE  branch_city = 'Brooklyn');
```