2152118 史君宝 数据库原理 第二次作业

1. 第一题：

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.

a. Find the ID of each customer of the bank who has an account but not a loan.

b. Find the ID of each customer who lives on the same street and in the same city as customer '12345'.

c. Find the name of each branch that has at least one customer who has an account in the bank and who lives in "Harrison".

表 3.18：

branch(*branch_name*, *branch_city*, *assets*)
customer (*ID*, *customer_name*, *customer_street*, *customer_city*)
loan (*loan_number*, *branch_name*, *amount*)
borrower (*ID*, *loan_number*)
account (*account_number*, *branch_name*, *balance*)
depositor (*ID*, *account_number*)

Figure 3.18 Banking database.

解答：
(a) 查找银行中每个有帐户但没有贷款的 ID。
```
select ID from depositor EXCEPT select ID from borrower;
```

(b) 查找与客户'12345'住在同一街道和同一城市的每个客户的 ID。
```
select F.ID from customer as F, customer as S where F.customer_street = S.customer_street and F.customer_city = S.customer_city and S.ID = 12345;
```

(c)找出至少有一个客户在银行有账户并且住在“Harrison”的每个分行的名称。
```
select distinct branch_name from account, depositor, customer where customer.ID = depositor.ID and depositor.account_number = account.account_number and customer_city = 'Harrison'
```

2. 第二题：

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

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

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

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

表 3.18：

branch(*branch_name*, *branch_city*, *assets*)
customer (*ID*, *customer_name*, *customer_street*, *customer_city*)
loan (*loan_number*, *branch_name*, *amount*)
borrower (*ID*, *loan_number*)
account (*account_number*, *branch_name*, *balance*)
depositor (*ID*, *account_number*)

Figure 3.18 Banking database.

(a)找到在位于“Brooklyn”的每个分行拥有帐户的每个客户。
select distinct customer.ID from customer,depositor,account,branch where customer.ID = depositor.ID and depositor.account_number = account.account_number and account.branch_name = branch.branch_name and branch_city = “Brooklyn”;

(b)找出银行中所有贷款金额的总和。
select sum(amount) from account;

(c)找出资产大于至少一个位于“布鲁克林”的分支机构的所有分支机构的名称。
select branch_name from branch where assets > (select MIN(assets) from branch where branch_city = 'Brooklyn');

3. 第三题：

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

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

b. Find ID and name of each employee who lives in the same city and on the same street as does her or his manager.

c. Find ID and name of each employee who earns more than the average salary of all employees of her or his company.

d. Find the company that has the smallest payroll.

图 3.19：

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

Figure 3.19 Employee database.

(a)查找每个员工的 ID 和姓名，这些员工居住在与其工作的公司所在的城市。

```
Select ID, person_name from employee join works ON employee.ID = works.ID
join company ON works.company_name = company.company_name where
employee.city = company.city;
```

(b)找到每个与她或他的经理住在同一城市和同一街道的员工的 ID 和名字。

```
select ID, person_name from employee join manages on employee.manager_id
= manages.ID as A, employee as B where A.city = B.city and A.street =
B.street;
```

(c)找出每个收入超过公司所有员工平均工资的员工的 ID 和名字。

```
SELECT e.ID, e.person_name
FROM employee e
JOIN works w ON e.ID = w.ID
JOIN company c ON w.company_name = c.company_name
WHERE w.salary > (
    SELECT AVG(w2.salary)
    FROM works w2
    WHERE w2.company_name = w.company_name
);
```

(d)找到工资最少的公司。

```
select company.company_name from company join works on
company.company_name = works.company_name group by company.company_name
having MIN(works.salary) = (select MIN(w2.salary) from works w2);
```