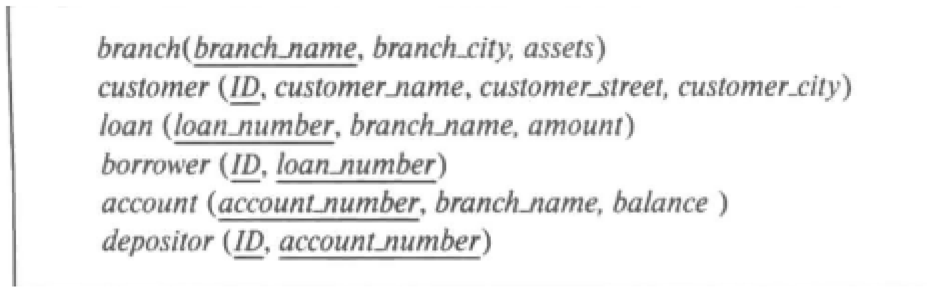
数据库作业


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)

3.8

a.找出银行中有账户但无贷款的每位客户的 ID

```
(select ID
from customer)
except
(select ID
from borrower

```

b.找出与客户 '12345' 居住在同 个城市、同一个街道的每位客户的 ID

```
SELECT ID
FROM customer
WHERE (customer_street, customer_city) IN
    (SELECT customer_street, customer_city FROM customer WHERE ID = 123
  AND ID <> 1234;
```

c．找出每个这样的支行的名称：在这些支行中至少有一位居住在 “Harrison" 的客户开设了账户

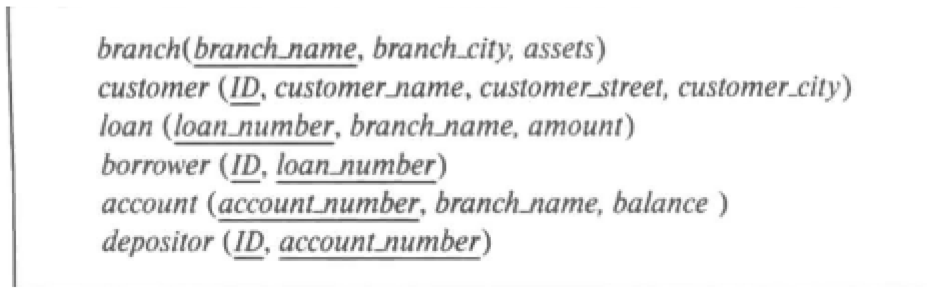
```
with cus_id as
SELECT ID
FROM customer
WHERE customer_city = "Harrison"

with cus_account as
SELECT account_number
```

```
FROM depositer
WHERE ID in cus_id

SELECT branch_name
FROM account
WHERE account_number in cus_account
```

3.15



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)

Brooklyn所有支行都有账户的人

```
SELECT ID
FROM customer AS c1
WHERE NOT EXISTS (
   (SELECT branch_name
    FROM branch
    WHERE branch_city = 'Brooklyn')

   EXCEPT

   (SELECT a.branch_name
    FROM account AS a
    JOIN depositor AS d ON a.account_number = d.account_number
    WHERE d.ID = c1.ID)
);
```

找出银行所有贷款的总和

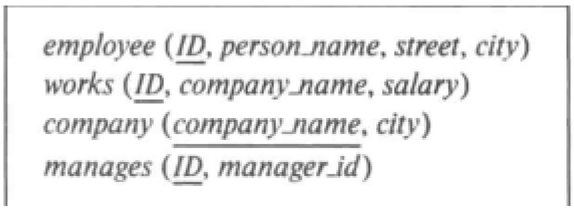
```
SELECT sum(amount)
```

```
FROM loan
```

找出资产比位于“Brooklyn”的至少一家支行要多的所有支行的名称

```
SELECT branch_name
FROM branch
WHERE assest >some (
                    SELECT assest
                    FROM branch
                    WHERE branch_city = "Brooklyn")
```

3.16



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

找出每位雇员的姓名和ID：该雇员所居住的城市与其工作的公司所在的城市一样

```
SELECT e.ID, e.person_name
FROM employee AS e
JOIN works AS w ON e.ID = w.ID
JOIN company AS c ON w.company_name = c.company_name
WHERE e.city = c.city;
```

找出所居住的城市街道与其经理形同的每位雇员的ID和姓名

```
SELECT e.ID, e.person_name
FROM employee AS e
JOIN manages AS m ON e.ID = m.ID
JOIN employ AS e2 ON m.manages_id= e2.ID
WHERE e.city = e2.city and e.street = e2.street
```

找出工资高于其所在公司的所有雇员平均工资的每位雇员的id和姓名

```
with company_salary(company_name,avg_salary) as
(
SELECT company_name,avg(salary) as avg_salary
FROM works
group by company_name
)

SELECT ID,person_name
FROM employee as e
join works as w on e.ID = w.ID
WHERE w.salary >all(
                SELECT avg_salary
                FROM company_salary as c
                WHERE c.company_name = w.company_name
                )
```

找出工资总和最小的公司

```
with company_salary(company_name,sum_salary) as
(
SELECT company_name,sum(salary) as sum_salary
FROM works
group by company_name
)

SELECT company_name
FROM company_salary
WHERE sum_salary=
(
    SELECT min(sum_salary)
        FROM company_salary
)
```

3.17

为First Bank Corporation 的所有雇员增长10%的工资

```
update works
set salary = salary * 1.1
```

```
WHERE company_name = "First Bank Corporation"
```

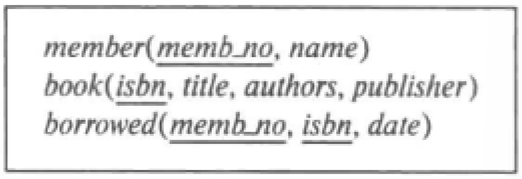
为First Bank Corporation 的所有经理增长10%的工资

```
update works
set salary = salary * 1.1
WHERE ID in （
            SELECT manager_id
            FROM manages
            )
```

删除“Small Bank Corporation”的雇员在works关系中的所有元组

```
delete from works
where company_name = "Small Bank Corporation"
```

3.21



member(*memb_no*, *name*)
book(*isbn*, *title*, *authors*, *publisher*)
borrowed(*memb_no*, *isbn*, *date*)

a . 找出借阅了至少一本由“MCGraw-hill”出版的书的每位会员的编号和姓名

```
SELECT memb_no ,name
FROM member as m
join borrowed as b on m.memb_no = b.memb_no
join book as bo on b.isbn = bo.isbn
WHERE publisher = "MCGraw-hill"
```

b. 找出借阅了所有由“MCGraw-hill”出版的书的每位会员的会员编号和姓名

```
SELECT m.memb_no, m.name
FROM member as  m
WHERE NOT EXISTS (
   SELECT b.isbn
   FROM book as b
```

```
    WHERE b.publisher = 'McGraw-Hill'
    except (
        SELECT isbn
        FROM borrowed br
        WHERE br.memb_no = m.memb_no

    )
);
```

c.对于每家出版商，找出借阅了超过5本由该出版商出版的书的每位会员的会员编号和姓名

```
SELECT m.memb_no ,m.name,bo.publisher
FROM member as m
join borrowed as b on m.memb_no = b.memb_no
join book as bo on b.isbn = bo.isbn
group by memb_no ,name, publisher
having count(distinct isbn )  > 5
```

d.找出会员借阅书籍的平均数量。考虑这样的情况：如果某会员没有借阅任何书籍，那么该会员根本不会出现在borrowed关系中，但是该会员应参与平均运算

```
with member_books(memb_no ,book_count) as
(
SELECT m.memb_no ,count(b.isbn) as book_count
FROM member as m
left join borrowed as b on m.memb_no = b.memb_no
group by memb_no
)
SELECT avg(book_count)
FROM member_books
```