2.12

1. 适合的主码

branch: branch\_name

customer: ID

loan: loan\_number

borrower: ID, loan\_number

account: account\_number

depositor: ID, account\_number

1. 适当的外码

borrower中的loan\_number引用于loan中的loan\_number

depositor 中的account\_number引用于account中的account\_number

loan 中的 branch\_name 引用于branch中的branch\_name

account 中的 branch\_name 引用于branch中的branch\_name

borrower 中的ID引用于customer中的ID

depositor 中的ID引用于customer中的ID

2.15

1. Πloan\_number(σaccount >10000(loan))
2. ΠID(σbalance > 6000∧depositor.account\_number = account.account\_number(depositor × account))
3. ΠID(σbalance > 6000∧depositor.account\_number = account.account\_number∧branch\_name = ‘Uptown’(depositor × account))

3.15

-- a，找出ID，姓名，在布鲁克林所有支行都有账户的人

select customer.ID, customer\_name

from customer, depositor

where customer.ID = depositor.ID

    and not exists(

        (

            select branch\_name

            from branch

            where branch\_city = 'Brooklyn'

        )

        except

        (

            select branch\_name

            from account

            where account.account\_number = depositor.account\_number

        )

    );

-- b 找出贷款总额

select sum(amount)

from loan;

-- c找出资产大于某一家布鲁克林分行的分行

select branch\_name

from branch as B1

where assets > some(

    select assets

    from branch as B2

    where branch\_city = 'Brooklyn'

);

3.22

-- 改写unique(select title from course)

-- 判断课程title和加上distinct关键字后的count是否相同

where (select count(title) from course) = (select count(distinct title) from course);

3.28

-- 找老师，老师讲授了其所在系的所有课程，按照姓名排序

-- 系的所有课 - 教师讲得所有课 = 空集

select I.name

from instructor as I

where not exists(

    (

        select course\_id

        from course

        where I.dept\_name = course.dept\_name

    )

    except

    (

        select course\_id

        from teaches

        where teaches.ID = I.ID

    )

)

order by I.name;