

数据管理基础

ch03_SQL语言
复习思考题
(参考答案)

智能软件与工程学院



1. 什么是子查询？什么是独立子查询？什么是相关子查询？它们之间的区别是什么？

- ❑ 一个**SELECT-FROM**语句称为一个‘查询块’。将一个查询块嵌套在另一个查询块中的查询称为‘嵌套查询’。在嵌套查询中，上层的查询块称为外层查询或父查询，下层查询块称为内层查询或子查询。
- ❑ 如果子查询的查询条件不依赖于父查询，称为‘不相关子查询’，又称为‘独立子查询’；如果子查询的查询条件依赖于父查询，称为‘相关子查询’。
- ❑ 不相关子查询的执行过程是“由里向外，逐层处理”，而相关子查询的执行过程是“由外向里，嵌套循环”。不相关子查询有更高的执行效率。

2. 关系代数中的‘差’运算在SQL中可以用什么查询谓词来表示？是否可以不使用子查询？

- ❑ 可以用 **NOT IN** 或者 **NOT EXISTS** 加上嵌套子查询来表示，也可以用两个子查询之间的**EXCEPT**差运算。在SQL查询表示中必须使用到子查询。

3. 在关系代数中用‘除法’表示的查询，在SQL中如何表示？是否可以用SQL中的量化比较谓词**ALL**来描述‘除法’的语义？是否可以不使用子查询？

- ❑ 可以用 **NOT EXISTS** 构造构造三层嵌套查询来表示，其中，第三层的**NOT EXISTS**谓词可以改用**NOT IN**谓词。也可以借用关系代数中‘除法’的推导公式，将其转换成SQL的查询表示。
- ❑ 一般情况下，无法用量化比较谓词**ALL**来描述‘除法’的查询语义。
- ❑ 不使用子查询，也无法表示‘除法’的查询语义。

4. 关于表的连接查询，在SQL语言中可以有哪几种表示方法？

- ❑ 可以有多种表示方法：
 - 单层的多表连接查询
 - 多层嵌套查询：涉及到嵌套子查询的谓词有 **IN**或**NOT IN**、**EXISTS**或**NOT EXISTS**、量词**SOME**或**ANY**或**ALL**。

ch03 - 复习思考题 - 参考答案

设有一个商品零售管理数据库MyCAP，关系模式如下：

关系	属性	关系模式
客户	客户编号，姓名，所在城市，优惠折扣	customers(cid, cname, city, discnt)
供应商	供应商编号，名称，所在城市，销售提成比例	agents(aid, aname, city, percent)
商品	商品编号，名称，存放城市，库存数量，单价	products(pid, pname, city, quantity, price)
订单	订单编号，销售日期，客户编号，供应商编号，商品编号，销售数量，销售金额	orders(ordno, orddate, cid, aid, pid, qty, dols)

其中：

- ① cid, aid, pid, ordno分别是客户、供应商、商品、订单关系的码；
- ② orddate是日期（DATE）类型的字段；
- ③ 可以根据ordno的大小来区分订单的先后，编号小的订单在前。

ch03 - 复习思考题 - 参考答案

5. 请用**SQL**语言来查询满足下述条件的客户的**cid**和**cname**:

Q₁: 没有通过纽约市的供应商去购买过商品;

Q₂: 只购买过两种不同(**pid**)商品;

Q₃: 所有订单的购买金额(**dols**)都超过**1000**美元;

Q₄: 通过纽约市的供应商去购买过所有价格超过**1**美元的商品;

6. 请用**SQL**语言来表示下述查询 (结果返回**cid**和**pid**)

Q₅: 客户**C**只购买过一次;

Q₆: 客户**C**只购买过商品**P** 1 次;

Q₇: 客户**C**只购买过商品**P**这一种商品;

7. 按照下述要求写出对应的**SQL**查询语句:

Q₈: 在所有有客户的城市中都被销售过的商品的编号;

Q₉: 返回每一个客户的编号及其最后两份订单的订购日期;

Q₁₀: 查询满足下述条件的供应商编号和名称: 向所有折扣**discnt**最高的客户都销售过商品;

Q₁₁: 检索符合下述条件的商品的编号: 至少有一个客户通过与该客户位于同一个城市的供应商订购过该商品;

Q₁₂: 检索为居住在南京市的所有客户订购过同一种商品的供应商的编号;

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

5-Q₁: 查询满足下述条件的顾客的cid和cname: 没有通过纽约市的供应商去购买过商品;

Q₁ (参考答案1)

```
select cid, cname
from customers
where cid not in (
    select o.cid
    from orders o, agents a
    where o.aid=a.aid and a.city='纽约');
```

Q₁ (参考答案2)

```
select cid, cname
from customers c
where not exists (
    select *
    from orders o, agents a
    where o.aid=a.aid and o.cid=c.cid
    and a.city='纽约');
```

Q₁ (参考答案3)

```
( select cid, cname
  from customers )

except

( select c.cid, c.cname
  from customers c, orders o, agents a
  where o.aid=a.aid and o.cid=c.cid
    and a.city='纽约' );
```

□ 说明:

- ① 查询条件含有否定语义, 其查询命令中必须使用子查询;
- ② 在使用并、交、差等集合查询时, 子查询结果列中一般必须含有查询目标的码。

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

5-Q₂: 查询满足下述条件的顾客的cid和cname: 只购买过两种不同(pid)商品;

Q₂ (参考答案1) 不使用统计函数

```
select c.cid, c.cname
from customers c, orders x, orders y
where x.cid=c.cid and y.cid=c.cid and x.pid<>y.pid
and not exists (
    select *
    from orders z
    where z.cid=c.cid and z.pid<>x.pid
    and z.pid<>y.pid );
```

Q₂ (参考答案3) 使用 group by + having 查询

```
select c.cid, c.cname
from customers c, orders o
where o.cid=c.cid
group by c.cid, c.cname
having count(distinct pid)=2 ;
```

/ 在goup by子句中必须包含cid和cname这两个分组属性 */*

Q₂ (参考答案2) 在where子句中使用统计子查询

```
select c.cid, c.cname
from customers c
where 2 in ( select count(distinct pid)
            from orders o
            where o.cid=c.cid );
```

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

5-Q₃: 查询满足下述条件的顾客的cid和cname: 所有订单的销售金额(**dols**)都超过**1000**美元;

Q₃ (参考答案1) 不使用统计函数

```
select c.cid, c.cname
from customers c, orders x
where x.cid = c.cid and not exists (
    select *
    from orders y
    where y.cid = c.cid and y.dols <= 1000 );
```

Q₃ (参考答案3) 使用分组选择统计函数

```
select c.cid, c.cname
from customers c, orders o
where o.cid = c.cid
group by c.cid, c.cname
having min(o.dols) > 1000 ;
```

Q₃ (参考答案2) 不使用统计函数

```
select c.cid, c.cname
from customers c
where 1000 < all ( select o.dols
                  from orders o
                  where o.cid = c.cid );
```

Q₃ (参考答案4)

```
( select c.cid, c.cname
  from customers c, orders o
  where o.cid = c.cid )
except
( select c.cid, c.cname
  from customers c, orders o
  where o.cid=c.cid and o.dols <= 1000 );
```


顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

5-Q₄: 查询满足下述条件的顾客的cid和cname: 通过纽约市的供应商去购买过所有价格超过1美元的商品;

Q₄ (参考答案)

```
select cid, cname
from customers c
where not exists (
    select *
    from products p
    where p.price > 1 and not exists (
        select *
        from orders o, agents a
        where o.cid=c.cid and o.pid=p.pid and
              o.aid=a.aid and a.city='纽约' ));
```

说明:

- ① 两个子条件 $p.price > 1$ 和 $a.city='纽约'$ 不能写错位置;
- ② 如果想模仿Q₁-参考答案3的表示方法, 并不能简化最终的查询表达式;
- ③ 本题查询的目标顾客, 可以是通过纽约市的不同供应商去购买所有价格超过1美元的商品;
- ④ 如果将查询要求修改为“通过位于纽约市的同一个供应商去购买过所有价格超过1美元的商品”, 那么SQL查询该如何表示?
- ⑤ 可以将最内层的not exists谓词改为使用not in谓词。

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

6-Q₅: 请用SQL语言来表示下述查询 (结果返回cid和pid): 顾客C只购买过一次;

Q₅ (参考答案1)

```
select x.cid, x.pid
from orders x
where not exists
  ( select *
    from orders y
    where y.cid = x.cid and y.ordno <> x.ordno );
```

Q₅ (参考答案2)

```
select x.cid, x.pid
from orders x
where 1 = ALL ( select count(*)
                from orders y
                where y.cid = x.cid );
```

/ 因为子查询只返回单个查询结果, 因此本题也可以用 =SOME 或 IN 来代替 =ALL */*

Q₅ (参考答案3)

```
select cid, pid
from orders
where cid IN ( select cid
               from orders
               group by cid
               having count(*)=1 );
```

- ① 本题查询的是“只购买过一次商品的顾客”，结果返回满足条件的顾客的编号及其购买过的商品的编号，请注意区分Q₅、Q₆、Q₇这三个查询在语义上的区别；
- ② 在参考答案3中，为什么没有对两个 orders 表定义别名？
- ③ 在本题中，不能直接使用如下的 group by+having 查询方式，为什么？

```
select cid, pid from orders group by cid having count(*)=1 ;
```

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

6-Q₆: 请用SQL语言来表示下述查询 (结果返回cid和pid): **顾客C只购买过商品P 1 次;**

Q₆ (参考答案1)

```
select x.cid, x.pid
from orders x
where not exists (
    select *
    from orders y
    where y.cid = x.cid and y.pid = x.pid
    and y.ordno <> x.ordno );
```

Q₆ (参考答案2)

```
select x.cid, x.pid
from orders x
where 1 = ALL (
    select count(*)
    from orders y
    where y.cid = x.cid and y.pid = x.pid );
```

Q₆ (参考答案3)

```
select cid, pid
from orders
group by cid, pid
having count(*) = 1 ;
```

- ① 在本题参考答案2及参考答案3中，数字 '1' 可以是任意的数字n，也可以通过改变比较运算符'='来方便地表示“不超过n次”、“至少n次”、.....等查询语义；
- ② 不建议使用 (cid, pid) IN (select cid, pid) 表示方式！
- ③ 在本题中，以下的查询表达式为什么是错误的？

```
select cid, pid from orders
where cid IN ( select cid
    from orders
    group by cid, pid
    having count(*)=1 );
```

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

6-Q₇: 请用SQL语言来表示下述查询 (结果返回cid和pid): 顾客C只购买过商品P 1 次;

Q₇ (参考答案1)

```
select distinct x.cid, x.pid
from orders x
where not exists (
    select *
    from orders y
    where y.cid = x.cid and y.pid <> x.pid );
```

Q₇ (参考答案2)

```
select distinct x.cid, x.pid
from orders x
where 1 = ALL ( select count ( distinct y.pid )
                from orders y
                where y.cid = x.cid );
```

Q₇ (参考答案3)

```
select distinct cid, pid
from orders
where cid IN ( select cid
               from orders
               group by cid
               having count(distinct pid)=1 );
```

- ① 请体会在统计函数中加 **distinct** 的用法:
count (distinct pid)
- ② 在这个查询中, 在**select**子句中使用了**distinct**谓词, 为什么在Q₅和Q₆中没有?
- ③ 在本题中, 以下的查询表达式为什么是错误的?

```
select cid, pid
from orders
group by cid
having count(distinct pid) = 1 ;
```

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-Q₈: 在所有有顾客的城市中都被销售过的商品的编号;

Q₈ (参考答案1)

```
select pid
from products p
where not exists (
    select *
    from customers c
    where not exists (
        select *
        from customers x, orders o
        where x.cid=o.cid and o.pid=p.pid
        and x.city=c.city ) );
```

Q₈ (参考答案2)

```
select pid
from products p
where not exists (
    select *
    from customers c
    where c.city not in (
        select x.city
        from customers x, orders o
        where x.cid=o.cid and o.pid=p.pid ) );
```

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-Q₉: 返回每一个顾客的编号及其最后两份订单的订购日期;

Q₉ (参考答案1)

```
select x.cid, y.orddate as last_ord,
       x.orddate as prior_to_last_ord
from orders x, orders y
where x.cid=y.cid and x.ordno < y.ordno
     and not exists (
       select *
       from orders z
       where z.cid=x.cid and
             z.ordno > x.ordno and
             z.ordno <> y.ordno );
```

Q₉ (参考答案2)

```
select x.cid, y.orddate as last_ord,
       x.orddate as prior_to_last_ord
from orders x, orders y
where x.cid=y.cid and
     x.ordno < y.ordno and
     x.ordno >= all (
       select z.ordno
       from orders z
       where z.cid=x.cid and
             z.ordno <> y.ordno );
```

说明:

- ① 在题目里规定了: 以ordno的大小来区分订单的先后;
- ② 这里没有考虑另外两种情况: 只有一份订单的顾客, 以及没有购买过任何商品的顾客。

顾客 customers (<u>cid</u> , cname, city, discnt)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-Q₁₀: 查询满足下述条件的供应商编号和名称：向所有折扣discnt最高的顾客都销售过商品；

Q₁₀ (参考答案)

```

select aid, aname
from agents a
where not exists (
    select *
    from customers c
    where c.discnt in (select max(distinct) from customers)      /* 折扣最高的顾客 */
    and not exists (
        select *
        from orders o
        where o.aid=a.aid and o.cid=c.cid ) );
/* 不存在供应商a和顾客c之间的订单 */

```

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-Q₁₁: 检索符合下述条件的商品的编号: 至少有一个顾客通过与该顾客位于同一个城市的供应商订购过该商品;

Q₁₁ (参考答案1) 表连接查询

```
select o.pid
from orders o, customers c, agents a
where o.cid=c.cid and
      o.aid=a.aid and
      c.city=a.city ;
```

Q₁₁ (参考答案2) 使用自然连接操作符

```
select pid
from ( orders natural join customers )
     natural join agents ;
```

说明:

- ① 本题也可以使用嵌套查询来表示, 但意义不大;
- ② 在参考答案2中, (orders natural join customers) 与 agents 有两个同名的属性 aid 和 city, 它们之间的 natural join 恰好满足本题的查询要求;
- ③ 在参考答案2中, 也可以在FROM子句中使用.....join.....on..... 或者join.....using..... 来实现表的联接。例如, 订单与顾客关系通过属性cid进行等值连接, 可表示为:
orders join customers on cid

顾客 customers (<u>cid</u> , cname, city, discent)	商品 products (<u>pid</u> , pname, city, quantity, price)
供应商 agents (<u>aid</u> , aname, city, percent)	订单 orders (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-Q₁₂: 检索为居住在南京市的所有顾客订购过同一种商品的供应商的编号;

Q₁₂ (参考答案)

```
select distinct aid
from orders o1
where not exists (
    select *
    from customers c
    where c.city = '南京' and not exists (
        select *
        from orders o2
        where o2.aid=o1.aid and o2.cid=c.cid and o2.pid=o1.pid ) );
```

说明：请注意本题与前面的查询Q₁₀的区别。

- ① 最外层查询：查询对象不同（agents 与 orders）；
- ② 本题使用了 distinct 谓词；
- ③ 第三层子查询的where条件表达式不同。

8. 在使用字符匹配查询谓词**like**时，<匹配串>中可以使用哪些通配符？它们的含义分别是什么？

❑ <匹配串>可以是一个完整的字符串，也可以含有通配符%（可匹配任意长度（包括长度为0）的字符串）和_（可匹配任意单个字符）

➤ 例如：**a%b**表示以“a开头、以b结尾”的任意长度字符串

➤ 例如：**a_b**表示“以a开头、以b结尾”的长度为3的任意字符串

❑ 也可以使用‘换码字符’将通配符转义为普通字符

➤ 例如：**'a_b%' escape '\'** 表示以“a开头、第三个字符为b”的任意长度字符串

9. 关于聚集函数

- ① 在**SQL**语言中，引入了哪几种聚集函数？
- ② 哪一种聚集函数的统计结果是与被统计对象的元素类型无关？
- ③ 在被统计的对象集合中存在取值为空值的元素时，对聚集函数的计算将会产生什么样的影响？
- ④ **count(*)**、**count(cid)**、**count(distinct cid)**三者之间有什么区别？？

- ① 五种统计函数：元素个数统计**COUNT**，最大值统计**MAX**，最小值统计**MIN**，计算总和**SUM**，计算平均值**AVG**
- ② **COUNT**与被统计对象的元素类型无关，**SUM**和**AVG**只能统计数值类型，**MAX**和**MIN**可以统计字符、数值类型。
- ③ 在使用聚集函数对一个集合中的元素进行统计计算时，将忽略其中的‘空’值元素，然后再进行统计计算。
- ④ 它们返回的结果语义如下：
 - **COUNT(*)**：返回集合中的元组个数
 - **COUNT(cid)**：返回结果元组中**cid**的取值非空的元组个数
 - **COUNT(distinct cid)**：返回结果元组中**cid**的取值非空的不同**cid**值的个数

10. 在使用到group by子句的SQL查询中，目标列可以由哪些部分组成？

❑ 在select目标列中，只能包含以下两类结果列：

- 必须包含group by子句中的所有分组属性
- 可以没有统计计算，也可以调用一个或多个统计函数

11. 关于视图 (view)

- ① SQL语言中的‘基本表’和关系模型中的‘关系’有什么区别？
 - ② 在SQL语言中，‘视图’和‘基本表’有什么联系与区别？
 - ③ 在视图创建命令中，**WITH CHECK OPTION**选项的作用是什么？
 - ④ 在视图创建命令和基表创建命令中，定义内容有什么区别？
 - ⑤ 在SQL中创建视图的作用是什么？
-
- ① SQL语言中的‘基本表’是关系模型中的‘关系’在数据库管理系统中的实现，但它们之间存在着区别：关系必须满足元组的唯一性；但在基本表中，如果既没有定义主码(primary key)也没有定义非空的唯一码(unique)，允许在表中存在重复的行。
 - ② ‘基本表’是关系在数据库管理系统中的实现，需要为其中的元组分配物理存储空间；‘视图’是建立在‘基本表’上的导出表（虚表），在数据库中只有视图的定义信息，不需要为视图中的数据分配物理存储空间。
 - ③ 在视图创建命令中使用**WITH CHECK OPTION**选项，其作用是：如果允许在该视图上执行更新操作，则其更新后的结果元组仍然必需满足视图的定义条件。即通过该视图插入或修改后的新元组能够通过该视图上的查询操作查出来。
 - ④ 在基表创建命令中，需要定义列的数据类型，以及列和表上的数据完整性约束；在视图创建命令中，只需要给出视图对应的子查询，不需要有任何关于数据类型及完整性约束的定义。
 - ⑤ 视图的作用：能够简化用户的操作；能使不同用户以不同方式看待同一数据，适应数据库共享的需要；提供了一定程度的逻辑独立性；能够对机密数据提供安全保护；利用视图可以更清晰的表达查询。

12. 设有一个教务管理数据库，其中的关系模式如下：

关系	属性	关系模式
学生	学号, 姓名, 就读院系, 入学年份	Student (sno, sname, dept, regyear)
教师	工号, 姓名, 工作院系	Teacher (tno, tname, dept)
课程	课程号, 课程名, 开课院系, 学分, 课程类型	Course (cno, cname, dept, credit, optional)
选课	学号, 课程号, 授课教师工号, 选修年份, 选修学期, 成绩	SC (sno, cno, tno, scyear, semester, grade)
授课	主讲教师工号, 课程号, 授课年份, 授课学期, 课时数	TC (tno, cno, tcyear, semester, hours)

其中：

- 学号、工号、课程号分别是学生、教师、课程表的主码；
- 同一门课同一个学生只能有一条选课记录，一门课一位老师在同一年内的同一个学期内只能有一条授课记录；
- 成绩全部采用百分制，成绩大于或等于60分才能获得该门课程的学分；
- 单门课程的学分绩计算公式是：成绩 $grade \times 0.05 \times$ 学分 $credit$ ；
- 课程类型分‘核心’和‘选修’两种。

ch03 - 复习思考题 - 参考答案

(第12题) 请用SQL语言来表示下述数据访问请求。

1. 查询满足下述条件的课程的课程名及开课院系：课程名中包含有‘数据库’；
2. 查询满足下述条件的课程的课程号和课程名：‘开课院系’为空；
3. 查询满足下述条件的教师的工号和姓名：在2021至2023年间没有承担过授课任务；
4. 查询满足下述条件的教师的工号和姓名：只讲授过自己工作所在院系开设的课程；
5. 查询满足下述条件的学生的学号和姓名：修读并获得了‘计算机’系开设的所有‘核心’课的学分；
6. 查询满足下述条件的学生的学号、姓名和就读院系：2020年之前入学，但还没有修读完自己就读院系开设的所有‘核心’课；
7. 查询每一位老师第一次上课和最后一次上课的年份，结果返回教师工号和起、止授课年份；
8. 查询每一位教师的授课次数及累计课时数，结果只返回累计课时数满1000小时的教师的工号、授课次数、累计课时数，并按照累计课时数的降序输出结果；
9. 针对所有‘核心’课程都及格的同学，统计每一个人的平均学分绩（不含不及格课程），结果返回学号、就读院系、平均学分绩，并按照‘就读院系的升序’和‘平均学分绩的降序’输出结果；
10. 创建一个学生学习情况统计视图，视图中的属性包括：学号，姓名，就读院系，已获得的学分总数，‘核心’课程的平均学分绩。

ch03 - 复习思考题 12 - 参考答案

关系	属性	关系模式
学生	学号, 姓名, 就读院系, 入学年份	Student (sno, sname, dept, regyear)
教师	工号, 姓名, 工作院系	Teacher (tno, tname, dept)
课程	课程号, 课程名, 开课院系, 学分, 课程类型	Course (cno, cname, dept, credit, optional)
选课	学号, 课程号, 授课教师工号, 选修年份, 选修学期, 成绩	SC (sno, cno, tno, scyear, semester, grade)
授课	主讲教师工号, 课程号, 授课年份, 授课学期, 课时数	TC (tno, cno, tcyear, semester, hours)

1. 查询满足下述条件的课程的课程名及开课院系：课程名中包含有‘数据库’；

```
select cname, dept from Course where cname like '%数据库%';
```

2. 查询满足下述条件的课程的课程号和课程名：‘开课院系’为空；

```
select cno, cname from Course where dept IS NULL;
```

ch03 - 复习思考题 12 - 参考答案

关系	属性	关系模式
学生	学号, 姓名, 就读院系, 入学年份	Student (sno, sname, dept, regyear)
教师	工号, 姓名, 工作院系	Teacher (tno, tname, dept)
课程	课程号, 课程名, 开课院系, 学分, 课程类型	Course (cno, cname, dept, credit, optional)
选课	学号, 课程号, 授课教师工号, 选修年份, 选修学期, 成绩	SC (sno, cno, tno, scyear, semester, grade)
授课	主讲教师工号, 课程号, 授课年份, 授课学期, 课时数	TC (tno, cno, tcyear, semester, hours)

3. 查询满足下述条件的教师的工号和姓名：在2021至2023年间没有承担过授课任务；

```
select tno, tname
from Teacher
where tno not in ( select tno
                  from TC
                  where tcyear between 2021 and 2023 );
```

- 也可以在SC表上查询2021至2023年间担任过授课任务的教师工号；
- 也可以使用 NOT EXISTS，或者两个子查询之间的差运算来表示该查询。

ch03 - 复习思考题 12 - 参考答案

4. 查询满足下述条件的教师的工号和姓名：只讲授过自己工作所在院系开设的课程；

```
select x.tno, x.tname
from Teacher x
where x.tno IN ( select tno  from TC ) and
      not exists ( select *
                  from TC t, Course c
                  where t.tno=x.tno and t.cno=c.cno and x.dept<>c.dept );
```

➤ 因为结果集含教师的码，也可用两个子查询之间的差运算来获得查询结果（EXCEPT操作），该查询可表示如下：

```
( select x.tno, x.tname
  from teach t, Teacher x
  where t.tno=x.tno )
except
( select y.tno, y.tname
  from teach t, course c, Teacher y
  where t.tno=y.tno and t.cno=c.cno and y.dept<>c.dept );
```


ch03 - 复习思考题 12 - 参考答案

关系	属性	关系模式
学生	学号, 姓名, 就读院系, 入学年份	Student (sno, sname, dept, regyear)
教师	工号, 姓名, 工作院系	Teacher (tno, tname, dept)
课程	课程号, 课程名, 开课院系, 学分, 课程类型	Course (cno, cname, dept, credit, optional)
选课	学号, 课程号, 授课教师工号, 选修年份, 选修学期, 成绩	SC (sno, cno, tno, scyear, semester, grade)
授课	主讲教师工号, 课程号, 授课年份, 授课学期, 课时数	TC (tno, cno, tcyear, semester, hours)

5. 查询满足下述条件的学生的学号和姓名：修读并获得了‘计算机’系开设的所有‘核心’课的学分；

```
select S.sno, S.sname
from Student S
where not exists ( select *
                    from Course C
                    where C.dept='计算机' and C.optional='核心' and
                        not exists ( select *
                                    from SC R
                                    where R.sno=S.sno and R.cno=C.cno and R.grade>=60 ) ) ;
```

ch03 - 复习思考题 12 - 参考答案

关系	属性	关系模式
学生	学号, 姓名, 就读院系, 入学年份	Student (sno, sname, dept, regyear)
教师	工号, 姓名, 工作院系	Teacher (tno, tname, dept)
课程	课程号, 课程名, 开课院系, 学分, 课程类型	Course (cno, cname, dept, credit, optional)
选课	学号, 课程号, 授课教师工号, 选修年份, 选修学期, 成绩	SC (sno, cno, tno, scyear, semester, grade)
授课	主讲教师工号, 课程号, 授课年份, 授课学期, 课时数	TC (tno, cno, tcyear, semester, hours)

6. 查询满足下述条件的学生的学号、姓名和就读院系：2020年之前入学，但还没有修读完自己就读院系开设的所有‘核心’课

```
select S.sno, S.sname, S.dept
from Student S
where S.regyear < 2020 and exists (
    select * from Course C where C.dept=S.dept and C.optional='核心' and
    not exists (select * from SC R where C.cno=R.cno and R.sno=S.sno and R.grade>=60 ) );
```

ch03 - 复习思考题 12 - 参考答案

关系	属性	关系模式
学生	学号, 姓名, 就读院系, 入学年份	Student (sno, sname, dept, regyear)
教师	工号, 姓名, 工作院系	Teacher (tno, tname, dept)
课程	课程号, 课程名, 开课院系, 学分, 课程类型	Course (cno, cname, dept, credit, optional)
选课	学号, 课程号, 授课教师工号, 选修年份, 选修学期, 成绩	SC (sno, cno, tno, scyear, semester, grade)
授课	主讲教师工号, 课程号, 授课年份, 授课学期, 课时数	TC (tno, cno, tcyear, semester, hours)

7. 查询每一位老师第一次上课和最后一次上课的年份, 结果返回教师工号和起、止授课年份;

```
select x.tno, x.tcyear, y.tcyear
from TC x, TC y
where x.tno=y.tno and
      not exists ( select * from TC R where R.tno=x.tno and R.tcyear < x.tcyear ) and
      not exists ( select * from TC R where R.tno=y.tno and R.tcyear > y.tcyear );
```

➤ 说明: 查询结果中不含没有承担过授课任务的教师; 如果一位教师只在某一年承担过授课任务, 那么授课的起止年份是同一年。

ch03 - 复习思考题 12 - 参考答案

关系	属性	关系模式
学生	学号, 姓名, 就读院系, 入学年份	Student (sno, sname, dept, regyear)
教师	工号, 姓名, 工作院系	Teacher (tno, tname, dept)
课程	课程号, 课程名, 开课院系, 学分, 课程类型	Course (cno, cname, dept, credit, optional)
选课	学号, 课程号, 授课教师工号, 选修年份, 选修学期, 成绩	SC (sno, cno, tno, scyear, semester, grade)
授课	主讲教师工号, 课程号, 授课年份, 授课学期, 课时数	TC (tno, cno, tcyear, semester, hours)

8. 查询每一位教师的授课次数及累计课时数，结果只返回累计课时数满1000小时的教师的工号、授课次数、累计课时数，并按照累计课时数的降序输出结果；

```
select tno, count(*), sum(hours)
from TC
group by tno
having sum(hours) > 1000
order by sum(hours) DESC ;
```

ch03 - 复习思考题 12 - 参考答案

9. 针对所有‘核心’课程都及格的同学，统计每一个人的平均学分绩，结果返回学号，就读院系，平均学分绩，并按照‘就读院系的升序’和‘平均学分绩的降序’输出结果；

```
select S.sno, S.dept, AVG(R.grade*0.05*C.credit) as avgofcred
from Student S, SC R, Course C
where S.sno=R.sno and R.cno=C.cno and
      S.sno not in (select x.sno
                    from SC x, Course y
                    where x.cno=y.cno and y.optional='核心' and x.grade < 60)
group by S.sno, S.dept
order by S.dept ASC, avgofcred DESC;
```

➤ 说明：

- ① 在本题中，“所有核心课程都及格”被理解为“选修了的核心课程都及格”，如果要按照“选修了所有的核心课程且都及格”来理解的话，需要另外表示。
- ② 结果列需要包含所有的分组属性。

10. 创建一个学生学习情况统计视图，视图中的属性包括：学号，姓名，就读院系，已获得的学分总数，‘核心’课程的平均学分绩

```
create view stud_view as
  select S.sno, S.sname, S.dept, Y.sumofcred, Z.avgofcred
  from Student S,
    ( select R.sno, sum(C.credit)
      from SC R, Course C
      where R.cno=C.cno and R.grade>=60
      group by R.sno ) AS Y(sno, sumofcred) ,
    ( select R.sno, AVG(R.grade*0.05*C.credit)
      from SC R, Course C
      where R.cno=C.cno and C.optional='核心'
      group by R.sno ) AS Z(sno, avgofcred)
  where S.sno=Y.sno and S.sno=Z.sno ;
```

说明：

- ① 因为两个统计函数所统计的范围不同，所以不能在同一个子查询中同时完成这两个统计操作，所以在FROM子句中嵌入了两个子查询，分别对应两种不同的统计需求；
- ② 也可以将两个子查询分别创建为两个视图，再在视图上创建本题的视图。