

数据管理基础

ch02_4_关系代数的应用
复习思考题
(参考答案)

智能软件与工程学院



复习思考题 1

1. 请用五种基本运算分别写出下列两个扩充运算的推导公式：
① ‘自然联接’运算 ② ‘除’运算
2. 理解笛卡尔积、 θ -联接、自然联接三者之间的联系和区别，并掌握：
① 在何种场景下适合使用自然联接？
② 如何正确使用‘自然联接’运算。
3. 如何理解‘除’运算的语义？请解释其推导公式的含义，并掌握如何正确书写‘除’运算表达式。
4. 理解笛卡尔积/ θ -联接/自然联接和‘除’运算之间的区别，并回答：
① 在什么情况下适合应用‘除’运算而不是联接运算？请举一例并写出使用除法和不使用除法的两种查询表达式。
② 请列举一个与情况①相似但又不能使用除法的例子，并写出其查询表达式。
5. 什么是关系的自联接？请列举一个使用自联接的例子，并写出查询表达式。
6. 什么是外联接？请列举一个使用外联接的例子，并写出查询表达式。

复习思考题 2

7. 设有一个公司产品零售数据库，其关系模式如下(带下划线的属性是关键字；以订单编号ordno的大小来区分订单的先后，ordno小的在前)。请用关系代数分别写出下述查询。

关系名	属性集	关系模式
顾客	<u>顾客编号</u> , 姓名, 居住城市, 折扣	C (<u>cid</u> , cname, city, discnt)
供应商	<u>供应商编号</u> , 名称, 所在城市, 佣金比例	A (<u>aid</u> , aname, city, percent)
商品	<u>商品编号</u> , 名称, 库存城市, 库存数量, 单价	P (<u>pid</u> , pname, city, stqty, price)
订单	<u>订单编号</u> , 订购日期, 顾客编号, 供应商编号, 商品编号, 订购数量, 销售金额	O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

(1) 查询满足条件的顾客的编号cid:

- ① 只购买过 p01 和 p02 两种商品;
- ② “只购买过商品p01” 或者 “只购买过商品p02”;
- ③ 只购买过一种商品。

(2) 查询满足条件的供应商的名字aname: 没有销售过商品;

(3) 查询满足条件的供应商的编号aid:

- ① 只向供应商自己所在城市中的顾客销售过商品;
- ② 在所有有顾客的城市中都向客户销售过商品;
- ③ 向供应商自己所在城市中的所有顾客都销售过商品。

复习思考题 3

关系名	属性集	关系模式
顾客	<u>顾客编号</u> , 姓名, 居住城市, 折扣	C (<u>cid</u> , cname, city, discent)
供应商	<u>供应商编号</u> , 名称, 所在城市, 佣金比例	A (<u>aid</u> , aname, city, percent)
商品	<u>商品编号</u> , 名称, 库存城市, 库存数量, 单价	P (<u>pid</u> , pname, city, stqty, price)
订单	<u>订单编号</u> , 订购日期, 顾客编号, 供应商编号, 商品编号, 订购数量, 销售金额	O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

- (4) 在每一个供应商自己的所有订单中，查询销售金额dols最高的订单，结果返回供应商编号、该供应商的销售金额最高订单的订单编号和销售金额。
- (5) 查询每一个顾客的第一份订单和最后一份订单（在第①和第②小题中，不考虑没有订单和只有一份订单的顾客；在第③小题中，需要返回所有顾客的查询结果）
- ① 结果关系中只有 cid, ordno, orddate 三个属性；
 - ② 结果关系中有顾客编号cid, 第一份订单的ordno和orddate, 最后一份订单的ordno和orddate等五个属性；
 - ③ 结果关系的关系模式同上面的第②小题，每一个顾客返回一条查询结果。如果是没有订单的顾客，结果元组上只有该顾客的编号；如果是只有一份订单的顾客，那么这份订单就作为该顾客的最后一份订单，他的第一份订单信息为空。（提示：使用外联接）

复习思考题 4

关系	属性集	关系模式
学生	<u>学号</u> , 学生姓名, 就读院系, 年级	S (<u>sno</u> , sname, dept, grade)
课程	<u>课程号</u> , 课程名, 开课院系, 课程类型	C (<u>cno</u> , cname, dept, opt)
教师	<u>教师工号</u> , 教师姓名, 工作院系	T (<u>tno</u> , tname, dept)
选课	<u>学号</u> , <u>课程号</u> , 授课教师工号, 成绩, 修读年份	L (<u>sno</u> , <u>cno</u> , tno, score, years)

8. 设有一个如上所示的学生选课数据库，其中：带下划线的属性是关键字；课程类型分为‘必修’和‘选修’；同一门课同一个学生只能有一条选课记录；成绩采用百分制。请用关系代数写出下述查询。

查询满足下述条件的学生的学号和姓名：

- ① 只修读过自己就读院系开设的课程；
- ② 所有的修读课程成绩都合格（成绩 ≥ 60 ）；
- ③ 修读过‘计算机’系开设的所有‘必修’课程；
- ④ 修读了自己就读院系开设的所有‘必修’课程且在这些必修课程上的成绩都合格；
- ⑤ 2019级并且还没有修读通过自己就读院系开设的所有‘必修’课程。（存在自己就读院系开设的‘必修’课程，该同学还没有修读，或者虽然修读但成绩没达到合格。）

第1题 -- 参考答案

1. 请用五种基本运算分别写出下列两个扩充运算的推导公式:

① ‘自然联接’ 运算 ② ‘除’ 运算

① ‘自然联接’运算的推导公式

➤ 设有两个关系 R 和 S , 其关系模式如下, C_1, C_2, \dots, C_k 是它们两者的公共属性

$$\text{Head}(R) = \{ A_1, A_2, \dots, A_n, C_1, C_2, \dots, C_k \}$$

$$\text{Head}(S) = \{ B_1, B_2, \dots, B_m, C_1, C_2, \dots, C_k \}$$

➤ 结果关系的关系模式是:

$$\text{Head}(R \bowtie S) = \text{Head}(R) \cup \text{Head}(S) = \{ A_1, A_2, \dots, A_n, C_1, C_2, \dots, C_k, B_1, B_2, \dots, B_m \}$$

➤ 结果关系的推导公式是: $R \bowtie S = \pi_{A_1, A_2, \dots, A_n, C_1, C_2, \dots, C_k, B_1, B_2, \dots, B_m} (\sigma_{R.C_1=S.C_1 \wedge R.C_2=S.C_2 \wedge \dots \wedge R.C_k=S.C_k} (R \times S))$

② ‘除’运算的推导公式

➤ 设有两个关系 R 和 S , 其关系模式如下:

$$\text{Head}(R) = \{ A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m \}$$

$$\text{Head}(S) = \{ B_1, B_2, \dots, B_m \}$$

➤ 结果关系的关系模式是: $\text{Head}(R \div S) = \text{Head}(R) - \text{Head}(S) = \{ A_1, A_2, \dots, A_n \}$

➤ 结果关系的推导公式是: $R \div S = \pi_{A_1, A_2, \dots, A_n} (R) - \pi_{A_1, A_2, \dots, A_n} ((\pi_{A_1, A_2, \dots, A_n} (R) \times S) - R)$

第2题 -- 参考答案

2. 理解笛卡尔积、 θ -联接、自然联接三者之间的联系和区别，并掌握：①在何种场景下适合使用自然联接？②如何正确使用‘自然联接’运算。

都是关系的合并运算，都可以用来实现两个关系的合并。其中，‘笛卡尔积’是基本运算，‘ θ -联接’和‘自然联接’则是扩充运算，它们之间的联系和区别是：

❑ 笛卡尔积

- 是实现跨不同关系表进行数据访问的基础，在任意两个关系之间都可以用‘笛卡尔积’运算实现它们的合并；
- 设有一个 n 元关系 R 和 m 元关系 S ，笛卡尔积 $R \times S$ 的结果是一个 $(n+m)$ 元的关系，结果元组集合可形式化表示为： $R \times S = \{ (u, v) \mid u \in R \wedge v \in S \}$
- 在笛卡尔积的结果关系中，通常存在着很多无意义结果元组，需要通过后续选择运算过滤掉。

❑ θ -联接

- 通过用户定义的联接条件 F 来实现两个关系的合并；
- 可以把一次‘ θ -联接’运算看成是一次“笛卡尔积”+一次“选择运算”， θ -联接结果关系的关系模式与笛卡尔积的结果关系相同，结果元组集合则是笛卡尔积结果集的一个子集， θ -联接的推导公式如下： $R \bowtie_F S = \sigma_F(R \times S)$

❑ 自然联接

- 将两个关系通过它们的“所有同名属性的相等比较”来进行合并的一种关系运算；
- 自然联接结果关系的关系模式与笛卡尔积、 θ -联接的结果关系的模式不同；
- 也可以用“笛卡尔积+选择+投影”运算来实现自然联接的查询结果。（推导公式见第1题）

第2题 -- 参考答案 (cont.)

(cont.) 执行关系之间的联接查询时，一般通用方法是通过‘笛卡尔积’或者‘ θ -联接’来实现，更常用的方法是‘自然联接’。

❑ 一般方法：笛卡尔积 or θ -联接

- 当两个关系之间“不存在同名属性”，或者“联接条件不是基于同名属性的相等比较”时，需要采用‘笛卡尔积’或者‘ θ -联接’实现关系联接查询。
- 在笛卡尔积和 θ -联接的结果关系中可能存在同名属性，使用时需要加以区别。

❑ 常用方法：自然联接

- 自然联接的联接条件是隐含在运算符中的（所有同名属性的相等比较）
- 如果在两个关系之间“存在多对同名属性，而本次查询又不需要‘所有’的同名属性都相等”，此时有两种选择：
 - ① 不使用自然联结，转而采用笛卡尔积或 θ -联接来实现关系的合并；
 - ② 先对其中的一个关系执行投影运算，过滤掉其中不需要进行相等比较的那些同名属性，然后再使用自然联接运算。

❑ 难点（关系的自联接）：‘自联接’是指一个关系自己和自己进行联接运算，这个通常无法使用‘自然联接’，只能使用 θ -联接或者笛卡尔积。一般方法如下：

- 使用赋值运算定义‘同质不同名’的两个中间关系（元组集合相同，但关系名不同），当然也可以对中间关系中的属性进行重命名；
- 然后，再使用笛卡尔积或 θ -联接实现两个中间关系的合并；
- 也可以通过有目的地对中间关系中属性进行重命名，然后用自然联结实现中间关系的合并。

第3题 -- 参考答案

3. 如何理解‘除’运算的语义？请解释其推导公式的含义，并掌握如何正确书写‘除’运算表达式。

□ 关于‘除’运算的结果元组的语义：设在关系R和S之间可以执行‘除’运算 $R \div S$ ，结果关系为T。如果一个元组 $t \in T$ ，那么对于除数关系S中的每一个元组s，都有 $(t,s) \in R$

□ 关于‘除’运算的推导公式是 $R \div S = \pi_{A_1, A_2, \dots, A_n}(R) - \pi_{A_1, A_2, \dots, A_n}((\pi_{A_1, A_2, \dots, A_n}(R) \times S) - R)$

□ 其中每一步的计算目的如下：

1) $T_{max} := \pi_{A_1, A_2, \dots, A_n}(R)$ // T_{max} 是最大可能的结果元组集合

2) $R_{max} := T_{max} \times S$ // R_{max} 与关系R是同类关系，如果 $R_{max} \subseteq R$ ，那么 $R \div S$ 的结果关系就是 T_{max}

3) $T_1 := R_{max} - R$ // 事实上，会有很多 R_{max} 中的元组没有出现在关系R中

4) $T_2 := \pi_{A_1, A_2, \dots, A_n}(T_1)$ // T_2 就是关系 T_{max} 中不满足除运算结果要求的那些元组

// 即：对于关系 T_2 中的任一个元组q，至少能在关系S中找到一个元组s，

// 并且由元组q和s所构成的元组(q,s)不在关系R中出现。

5) $R \div S := T_{max} - T_2$

□ 在使用‘除’运算时，正确的使用方法如下：

- 我们将查询结果中的客观对象称为‘目标对象’，用于定义查询条件的客观对象称为‘条件对象’；
- 在被除数关系中必须包含目标对象和条件对象的关键字；
- 除数关系中只含条件对象的关键字；
- 被除数和除数关系中不能含其它‘不必要’的多余属性。

第4题 -- 参考答案

4. 理解笛卡儿积/ θ -联接/自然联接和‘除’运算之间的区别，并回答：

- ① 在什么情况下适合应用除运算而不是联接运算？请举一例并写出使用除法和不使用除法的两种查询表达式。
- ② 请列举一个与情况①相似但又不能使用除法的例子，并写出其查询表达式。

□ 我们将查询结果中的客观对象称为‘目标对象’，用于定义查询条件的客观对象称为‘条件对象’。在判断某个目标对象 t （结果元组 t ）是否属于结果关系时，

- 如果只需要从条件对象集合中找到一个条件对象 c （元组 c ），并且目标对象 t 和条件对象 c 之间满足查询条件，那么就直接使用‘联接’运算（包括笛卡尔积、 θ -联接和自然联接）；
- 如果需要条件对象集合中的所有条件对象 c 和目标对象 t 之间都满足查询条件，那么就使用‘除’运算。

顾客 C (<u>cid</u> , cname, city, discent)	商品 P (<u>pid</u> , pname, city, stqty, price)
供应商 A (<u>aid</u> , aname, city, percent)	订单 O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

① 例：向‘南京’市的所有顾客都销售过商品的供应商的编号；

- (使用除运算): $\pi_{aid, cid}(O) \div \pi_{cid}(\sigma_{city='南京'}(C))$
- (不使用除运算): $\pi_{aid}(O) - \pi_{aid}(\pi_{aid}(A) \times \pi_{cid}(\sigma_{city='南京'}(C))) - \pi_{aid, cid}(O)$

② 例：通过顾客自己所在城市中的所有供应商都购买过商品的顾客的编号；

- 本题不能使用除运算来表示，正确表示方法是: $\pi_{cid}(O) - \pi_{cid}(\pi_{cid, aid}(A \bowtie C) - \pi_{cid, aid}(O))$

第 5&6 题 -- 参考答案

5. 什么是关系的自联接？请列举一个使用自联接的例子，并写出查询表达式。

所谓的‘自联接’，就是指一个关系自己与自己进行‘联接’查询的情况。例如：在每一个城市中查询 percent 值最高的供应商，结果返回这些供应商所在的城市名称、供应商编号及其 percent 值。该查询的表示方法是：令 $M := A, N := A$ ，查询表达式为：

$$\pi_{city, aid, percent}(A) - \pi_{M.city, M.aid, M.percent}(\sigma_{M.city=N.city \wedge M.percent < N.percent}(M \times N))$$

顾客 C (<u>cid</u> , cname, city, discent)	商品 P (<u>pid</u> , pname, city, stqty, price)
供应商 A (<u>aid</u> , aname, city, percent)	订单 O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

6. 什么是外联接？请列举一个使用外联接的例子，并写出查询表达式。

所谓的‘外联接’，是一种特殊的关系联接运算。外联接结果关系的关系模式与它们之间自然联接的结果关系模式相同，外联接的结果元组集合既包含着它们自然联接的结果元组，也包含着联接对象关系中那些没有出现在自然联接结果集中的元组。在外联接的结果关系上，使用关系代数运算可以重构原来的联接对象关系。

例如：查询顾客与供应商位于同一个城市中的订单，结果返回城市名称、顾客编号、供应商编号、订单编号。结果集中也要包含顾客和供应商位于同一个城市、但他们之间没有产生过销售订单的情况（返回的订单编号为‘空值’）。

我们用‘left-outer-join’表示‘左外联接’运算，该查询可表示为：

$$\pi_{C.city, C.cid, A.aid, O.ordno}((C \bowtie A) \text{ left-outer-join } O)$$

第 7 题 -- 参考答案 1

C (<u>cid</u> , cname, city, discent)	A (<u>aid</u> , aname, city, percent)
P (<u>pid</u> , pname, city, stqty, price)	O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-(1) 查询满足条件的顾客的编号cid:

① 只购买过 p01 和 p02 两种商品;

$$\pi_{cid}(\sigma_{pid = 'p01'}(O)) \cap \pi_{cid}(\sigma_{pid = 'p02'}(O)) - \pi_{cid}(\sigma_{pid \neq 'p01' \wedge pid \neq 'p02'}(O))$$

② “只购买过商品p01” 或者 “只购买过商品p02”;

$$(\pi_{cid}(\sigma_{pid = 'p01'}(O)) - \pi_{cid}(\sigma_{pid \neq 'p01'}(O))) \cup (\pi_{cid}(\sigma_{pid = 'p02'}(O)) - \pi_{cid}(\sigma_{pid \neq 'p02'}(O)))$$

③ 只购买过一种商品。

$$\text{令 } R:=O, S:=O, \text{ 该查询可表示为: } \pi_{cid}(O) - \pi_{R.cid}(\sigma_{R.cid=S.cid \wedge R.pid \neq .pid}(R \times S))$$

7-(2) 查询满足条件的供应商的名字aname: 没有销售过商品;

$$\pi_{aname}((\pi_{aid}(A) - \pi_{aid}(O)) \bowtie A)$$

第7题 -- 参考答案2

C (<u>cid</u> , cname, city, discent)	A (<u>aid</u> , aname, city, percent)
P (<u>pid</u> , pname, city, stqty, price)	O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-(3) 查询满足条件的供应商的编号aid:

① 只向供应商自己所在城市中的顾客销售过商品;

$$\pi_{aid}(O) - \pi_{A.aid}(\sigma_{A.aid = O.aid \wedge O.cid = C.cid \wedge A.city \neq C.city}(A \times O \times C))$$

② 在所有有顾客的城市中都向客户销售过商品;

答案1 (仅使用基本运算符)

$$\pi_{aid}(O) - \pi_{aid}(\pi_{A.aid, C.city}(A \times C) - \pi_{O.aid, C.city}(\sigma_{O.cid = C.cid}(O \times C)))$$

答案2 (使用到自然连接运算) $\pi_{aid}(O) - \pi_{aid}(\pi_{A.aid, C.city}(A \times C) - \pi_{O.aid, C.city}(O \bowtie C))$

答案3 (使用到除法运算) $\pi_{O.aid, C.city}(O \bowtie C) \div \pi_{city}(C)$

③ 向供应商自己所在城市中的所有顾客都销售过商品。

$$\pi_{aid}(O) - \pi_{aid}(\pi_{aid, cid}(A \bowtie C) - \pi_{aid, cid}(O))$$

第 7 题 -- 参考答案 3

C (<u>cid</u> , cname, city, discent)	A (<u>aid</u> , aname, city, percent)
P (<u>pid</u> , pname, city, stqty, price)	O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

7-(4) 在每一个供应商自己的所有订单中，查询销售金额dols最高的订单，结果返回供应商编号、该供应商的销售金额最高订单的订单编号和销售金额。

令 $M:=O$ $N:=O$ ，该查询可表示为：

$$\pi_{aid, ordno, dols} (O) - \pi_{M.aid, M.ordno, M.dols} (\sigma_{M.aid=N.aid \wedge M.dols < N.dols} (M \times N))$$

第7题 -- 参考答案 4

7-(5) 查询每一个顾客的第一份订单和最后一份订单（在第①和第②小题中，不考虑没有订单和只有一份订单的顾客；在第③小题中，需要返回所有顾客的查询结果）

① 结果关系中只有 **cid, ordno, orddate** 三个属性；

➤ 查询每个客户的最后一份订单 E

（step 1）查询每个客户的‘历史订单’（每个客户在最后一份订单之前的订单）

令 $O_1 := O, O_2 := O, M := \pi_{O1.cid, O1.ordno, O1.orddate}(\sigma_{O1.cid=O2.cid \wedge O1.ordno < O2.ordno}(O_1 \times O_2))$

（step 2）查询每个客户的最后一份订单 E: $E(cid, lastord, lastdate) := \pi_{cid, ordno, orddate}(O) - M$

➤ 使用与上面类似的方法，查询每个客户的第一份订单 F

（step 3）查询每个客户的‘后继订单’（每个客户在第一份订单之后的订单）

令 $O_1 := O, O_2 := O, N := \pi_{O2.cid, O2.ordno, O2.orddate}(\sigma_{O1.cid=O2.cid \wedge O1.ordno < O2.ordno}(O_1 \times O_2))$

（step 4）查询每个客户的第一份订单 F: $F(cid, ordno, orddate) := \pi_{cid, ordno, orddate}(O) - N$

① 答：令 $lastorder(cid, ordno, orddate) := E$ ，本题查询可表示为： $lastorder \cup F$

说明：本题查询结果中，不含没有订单的顾客；如果是只有一份订单的顾客，结果集中只返回一条结果元组。

第7题 -- 参考答案5

7-(5) 查询每一个顾客的第一份订单和最后一份订单（在第①和第②小题中，不考虑没有订单和只有一份订单的顾客；在第③小题中，需要返回所有顾客的查询结果）

② 结果关系中有顾客编号cid, 第一份订单的ordno和orddate, 最后一份订单的ordno和orddate等五个属性；

➤ 查询每个客户的最后一份订单 E

(step 1) 查询每个客户的‘历史订单’（每个客户在最后一份订单之前的订单）

令 $O_1 := O$, $O_2 := O$, $M := \pi_{O1.cid, O1.ordno, O1.orddate}(\sigma_{O1.cid=O2.cid \wedge O1.ordno < O2.ordno}(O_1 \times O_2))$

(step 2) 查询每个客户的最后一份订单E: $E(cid, lastord, lastdate) := \pi_{cid, ordno, orddate}(O) - M$

➤ 使用与上面类似的方法，查询每个客户的第一份订单 F

(step 3) 查询每个客户的‘后继订单’（每个客户在第一份订单之后的订单）

令 $O_1 := O$, $O_2 := O$, $N := \pi_{O2.cid, O2.ordno, O2.orddate}(\sigma_{O1.cid=O2.cid \wedge O1.ordno < O2.ordno}(O_1 \times O_2))$

(step 4) 查询每个客户的第一份订单F: $F(cid, ordno, orddate) := \pi_{cid, ordno, orddate}(O) - N$

② 每个客户返回一条元组，包括 cid, 第一份订单ordno和orddate, 最后一份订单的lastord和lastdate, 本题查询可表示为：

$\pi_{F.cid, F.ordno, F.orddate, E.lastord, E.lastdate}(\sigma_{F.cid=E.cid \text{ and } F.ordno \neq E.lastord}(F \times E))$

第7题 -- 参考答案6

7-(5) 查询每一个顾客的第一份订单和最后一份订单（在第①和第②小题中，不考虑没有订单和只有一份订单的顾客；在第③小题中，需要返回所有顾客的查询结果）

③ 结果关系的关系模式同上面的第②小题，每一个顾客返回一条查询结果。如果是没有订单的顾客，结果元组上只有该顾客的编号；如果是只有一份订单的顾客，那么这份订单就作为该顾客的最后一份订单，他的第一份订单信息为空。（提示：使用外联接）

（step 1）查询每个客户的‘历史订单’（每个客户在最后一份订单之前的订单）

令 $O_1 := O$, $O_2 := O$, $H := \pi_{O1.cid, O1.ordno, O1.orddate}(\sigma_{O1.cid=O2.cid \wedge O1.ordno < O2.ordno}(O_1 \times O_2))$

（step 2）查询每个客户的最后一份订单E: $E(cid, lastord, lastdate) := \pi_{cid, ordno, orddate}(O) - H$

（step 3）查询每个客户的‘中间订单’（在历史订单H中，查询每个客户在第一份订单之后的订单）

令 $H_1 := H$, $H_2 := H$, $M := \pi_{H2.cid, H2.ordno, H2.orddate}(\sigma_{H1.cid=H2.cid \text{ and } H1.ordno < H2.ordno}(H_1 \times H_2))$

（step 4）查询每个客户的第一份订单F: $F(cid, ordno, orddate) := H - M$

③ 用 left-outer-join 表示‘左外联接’运算，本题查询可表示为：

$(\pi_{cid}(C) \text{ left-outer-join } E) \text{ left-outer-join } F$

第8题 -- 参考答案 1

关系	属性集	关系模式
学生	<u>学号</u> , 学生姓名, 就读院系, 年级	S (<u>sno</u> , sname, dept, grade)
课程	<u>课程号</u> , 课程名, 开课院系, 课程类型	C (<u>cno</u> , cname, dept, opt)
教师	<u>教师工号</u> , 教师姓名, 工作院系	T (<u>tno</u> , tname, dept)
选课	<u>学号</u> , <u>课程号</u> , 授课教师工号, 成绩, 修读年份	L (<u>sno</u> , <u>cno</u> , tno, score, years)

8. 设有一个如上所示的学生选课数据库，其中：带下划线的属性是关键字；课程类型分为‘必修’和‘选修’；同一门课同一个学生只能有一条选课记录；成绩采用百分制。请用关系代数写出下述查询。查询满足下述条件的学生的学号和姓名：

1. 只修读过自己就读院系开设的课程；

解： $\pi_{sno, sname} ((\pi_{sno}(L) - \pi_{sno}(\sigma_{S.sno=L.sno \wedge C.cno=L.cno \wedge S.dept \neq C.dept}(L \times S \times C))) \bowtie S)$

2. 所有的修读课程成绩都合格（成绩 ≥ 60 ）；

解： $\pi_{sno, sname} ((\pi_{sno}(L) - \pi_{sno}(\sigma_{score < 60}(L))) \bowtie S)$

第8题 -- 参考答案2

关系	属性集	关系模式
学生	<u>学号</u> , 学生姓名, 就读院系, 年级	S (<u>sno</u> , sname, dept, grade)
课程	<u>课程号</u> , 课程名, 开课院系, 课程类型	C (<u>cno</u> , cname, dept, opt)
教师	<u>教师工号</u> , 教师姓名, 工作院系	T (<u>tno</u> , tname, dept)
选课	<u>学号</u> , <u>课程号</u> , 授课教师工号, 成绩, 修读年份	L (<u>sno</u> , <u>cno</u> , tno, score, years)

3. 修读过‘计算机’系开设的所有‘必修’课程；

解1（不使用‘除’运算）：

$$\pi_{sno,sname}((\pi_{sno}(L) - \pi_{sno}((\pi_{sno}(S) \times \pi_{cno}(\sigma_{dept='计算机' \wedge opt='必修'}(C))) - \pi_{sno,cno}(L)))) \bowtie S)$$

解2（使用‘除’运算）： $\pi_{sno,sname,cno}(L \bowtie S) \div \pi_{cno}(\sigma_{dept='计算机' \wedge opt='必修'}(C))$

4. 修读了自己就读院系开设的所有‘必修’课程且在这些必修课程上的成绩都合格；

解（本题不能使用‘除’运算）：

$$\pi_{sno,sname}((\pi_{sno}(L) - \pi_{sno}((\pi_{S.sno,C.cno}(\sigma_{opt='必修'}(S \bowtie C))) - \pi_{sno,cno}(\sigma_{score \geq 60}(L)))) \bowtie S)$$

第 8 题 -- 参考答案 3

关系	属性集	关系模式
学生	<u>学号</u> , 学生姓名, 就读院系, 年级	S (<u>sno</u> , sname, dept, grade)
课程	<u>课程号</u> , 课程名, 开课院系, 课程类型	C (<u>cno</u> , cname, dept, opt)
教师	<u>教师工号</u> , 教师姓名, 工作院系	T (<u>tno</u> , tname, dept)
选课	<u>学号</u> , <u>课程号</u> , 授课教师工号, 成绩, 修读年份	L (<u>sno</u> , <u>cno</u> , tno, score, years)

5. 2019级并且还没有修读通过自己就读院系开设的所有‘必修’课程。（存在自己就读院系开设的‘必修’课程，该同学还没有修读，或者虽然修读了但成绩没达到合格。）

解（本题不能使用‘除’运算）：

$$\pi_{sno, sname}((\pi_{S.sno, C.cno}(\sigma_{grade=2019 \wedge opt='必修'}(S \bowtie C)) - \pi_{sno, cno}(\sigma_{score \geq 60}(L))) \bowtie S)$$