

HW01

□ 设有一个公司产品零售数据库，其关系模式如下(带下划线的属性是各个关系的码)

关系名	属性集	关系模式
顾客	顾客编号, 姓名, 居住城市, 折扣	C (<u>cid</u> , cname, city, discount)
供应商	供应商编号, 名称, 所在城市, 佣金比例	A (<u>aid</u> , aname, city, percent)
商品	商品编号, 名称, 库存数量, 单价	P (<u>pid</u> , pname, qty, price)
订单	订单编号, 订单日期, 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)

□ 请用关系代数分别写出下述查询。

1. 查询满足下述条件的订单，结果返回订单编号、顾客所在城市、供应商所在城市：订单上的顾客和供应商不在同一个城市中；
2. 查询满足下述条件的顾客的编号和名称：只有一份订单；
3. 查询满足下述条件的供应商的编号：在所有有顾客的城市中都销售过商品；(请写出两种不同的表示方法：使用除法和不使用除法)
4. 查询满足下述条件的供应商a和城市c的组合：供应商a向位于城市c中的所有顾客都销售过商品。结果只需要返回供应商的编号和城市名称两个属性。
5. 查询每一位供应商的第一份订单和最后一份订单，结果返回供应商编号、第一份订单的订单编号和日期、最后一份订单的订单编号和日期。(不考虑没有订单或只有一份订单的供应商)

$$1. \pi_{ordno, C.city, A.city} \left((A \bowtie C) \bowtie O \right)_{A.city \neq C.city}$$

$$2. \pi_{cid, cname} \left(C \bowtie \left(\pi_{cid}(O) - \pi_{cid}(O \bowtie_{cid_1 = cid_2, ordno_1 \neq ordno_2} O) \right) \right)$$

$$3. \text{除法: } \pi_{aid, city}(O \bowtie C) \div \pi_{city}(C)$$

不用除法:

$$K = \pi_{aid, city}(O \bowtie C)$$

$$\pi_{aid}(K) - \pi_{aid}(\pi_{aid}(A) \times \pi_{city}(C) - K)$$

$$4. R = \pi_{aid, cid, city}(O \bowtie C)$$

$$\pi_{aid, city}(R) - \pi_{aid, city}(\pi_{aid}(A) \times \pi_{aid, city}(C) - R)$$

有供应商未销售过顾客的城市

$$5. F(aid, fid, ftime) = \pi_{aid, ordno, orddate} \left(\left(\pi_{ordno}(O) - \pi_{ordno} \left(O \bowtie_{aid_1 = aid_2, orddate_1 > orddate_2} O \right) \right) \bowtie O \right)$$

$$L(aid, lid, ltime) = \pi_{aid, ordno, orddate} \left(\left(\pi_{ordno}(O) - \pi_{ordno} \left(O \bowtie_{aid_1 = aid_2, orddate_1 < orddate_2} O \right) \right) \bowtie O \right)$$

结果: $F \bowtie L$

□ 设有一个如下表所示的学生选课数据库，其中：带下划线属性是码；课程类型分为‘核心’和‘公选’；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，课程成绩大于或等于60分为‘通过’。

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

□ 请用关系代数写出下述查询。

6. 查询所有的跨院系课程选修情况(学生的就读院系与课程的开课院系不同)，结果返回学生的学号和就读院系、课程的课程号和开课院系、成绩；
7. 查询满足下述条件的教师的姓名：只讲授过自己工作院系开设的课程；
8. 查询满足下述条件的教师的编号和工作院系：讲授过自己工作院系开设的所有‘核心’课程；
9. 查询满足下述条件的学生的学号和就读院系：4年级。并且还存在于自己就读院系开设的‘核心’课

- 设有一个如下表所示的学生选课数据库，其中：带下划线属性是码；课程类型分为‘核心’和‘公选’；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，课程成绩大于或等于60分为‘通过’。

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

- 请用关系代数写出下述查询。

- 查询所有的跨院系课程选修情况（学生的就读院系与课程的开课院系不同），结果返回学生的学号和就读院系、课程的课程号和开课院系、成绩；
- 查询满足下述条件的教师的姓名：只讲授过自己工作院系开设的课程；
- 查询满足下述条件的教师的编号和工作院系：讲授过自己工作院系开设的所有‘核心’课程；
- 查询满足下述条件的学生的学号和就读院系：4年级，并且还存在于自己就读院系开设的‘核心’课没有通过（没有选修或课程成绩低于60分）；
- 查询满足下述条件的学生的学号和姓名：在选修的每一门课程上，都取得了该门课程的最高分。

$$6. \pi_{sno, sdept, cno, cdept, score} \left(\sigma_{S.dept \neq C.dept} (S \bowtie_{S.sno=L.sno} L \bowtie_{L.cno=C.cno} C) \right)$$

$$7. \pi_{tname} (L) - \pi_{tname} \left(\sigma_{C.dept \neq T.dept} (T \bowtie_{T.tno=L.tno} L \bowtie_{L.cno=C.cno} C) \right)$$

跨院系教师

$$CORE = \sigma_{opt = \text{核心}} (C) \quad \text{核心课}$$

$$SHOULD = \pi_{tno, cno} (T \bowtie CORE) \quad \text{应该有的匹配}$$

$$FACT = \pi_{tno, cno} (L) \quad \text{实际开的课}$$

$$\left(\pi_{tno} (SHOULD) - \pi_{tno} (SHOULD - FACT) \right) \bowtie \pi_{tno, dept} (T)$$

$$9. TS = \sigma_{grade = 4\text{年级}} (S) \quad \text{4年级}$$

$$CORE = \sigma_{opt = \text{核心}} (C) \quad \text{核心课}$$

$$CORETS = \pi_{sno, cno, dept} (TS \bowtie CORE) \quad \text{应有的}$$

$$PASTS = \pi_{sno, cno} \left(\sigma_{score \geq 60} (CORETS \bowtie L) \right) \quad \text{通过的}$$

$$\text{结果: } \pi_{sno} \left(\pi_{sno, cno} (CORETS) - PASTS \right) \bowtie \pi_{sno, dept} (TS)$$

$$10. \left(\pi_{sno} (L) - \pi_{sno} \left(L \bowtie_{\substack{cno_1=cno_2 \\ score_1 < score_2 \\ sno_1 \neq sno_2}} L \right) \right) \bowtie \pi_{sno, sname} (S)$$

分数比别人小的学生
不存在比他分数高的学生 \rightarrow All MAX