

<i>employee (pid, person name, street, city)</i>
<i>works (person name, pid, company name, cid, salary)</i>
<i>company (cid, company name, city)</i>

Figure 1 员工数据库

1. 考虑图 1 中的员工数据库。哪些是适当的主键？

employee (pid, person name, street, city): pid 是员工的唯一标识符，因此 pid 应该是主键。

works (person name, pid, company name, cid, salary): pid 和 cid 的组合可以唯一标识一个人在特定公司中的工作，因此 (pid, cid) 作为组合键是合适的主键。

company (cid, company name, city): cid 是公司表中每个公司的唯一标识符，因此 cid 是主键。

2. 考虑图 1 中的员工数据库。给出一个关系代数表达式来表示以下查询：

a. 查找每位不在“BigBank”工作的员工的 ID 和姓名。

- 首先，找出在“BigBank”工作的所有员工的 pid。
- 然后，将这些员工的 pid 从所有员工的 pid 中排除，最后返回剩下员工的 pid 和姓名。

关系代数表达式：

$$\pi_{pid, person\ name}(\sigma_{company\ name \neq "BigBank"}(employee \bowtie works \bowtie company))$$

b. 查找每位至少和数据库中的一位员工薪资一样多的员工的 ID 和姓名。

关系代数表达式：

$$\pi_{pid, person\ name}(\sigma_{salary \geq w.salary}(works \times works\ as\ w))$$

```
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)
```

Figure 2 银行数据库

3. 考虑图 2 中的银行数据库。给出关系代数表达式来表示以下查询：

a. 找到每个贷款金额大于 \$10,000 的贷款号。

- 这个查询涉及 `loan` 表中的 `amount` 字段，要求找到所有金额大于 10,000 的贷款号。

关系代数表达式：

$$\pi_{\text{loan_number}}(\sigma_{\text{amount} > 10000}(\text{loan}))$$

b. 找到每位有账户余额大于 \$6,000 的存款人的 ID。

- 这个查询需要结合 `account` 和 `depositor` 表，因为存款人的 ID 存在于 `depositor` 表中，而余额信息在 `account` 表中。

关系代数表达式：

$$\pi_{ID}(\sigma_{\text{balance} > 6000}(\text{account} \bowtie \text{depositor}))$$

c. 找到每位在 “Uptown” 分行有账户余额大于 \$6,000 的存款人的 ID。

- 这个查询还涉及 `branch` 表，因为需要过滤出特定分行的账户。先从 `branch` 表中找出分行为 “Uptown” 的账户，再结合其他表进行查询。

关系代数表达式：

$$\pi_{ID}(\sigma_{\text{balance} > 6000 \wedge \text{branch_name} = \text{Uptown}}((\text{account} \bowtie \text{branch}) \bowtie \text{depositor}))$$

```

classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)

```

Figure 3 大学数据库

5.使用大学数据库模式(Figure 3), 用关系代数编写以下查询:

a. 找到物理系的每位教师的 ID 和姓名。

- 这需从 `instructor` 表中找到 `dept_name` 为“Physics”的教师。

关系代数表达式:

$$\pi_{ID, name}(\sigma_{dept_name='Physics'}(instructor))$$

b. 找到位于“Watson”教学楼的每位系的教师的 ID 和姓名。

- 需结合 `instructor` 表和 `department` 表, 查询 `department` 表中 `building` 为“Watson”的教师。

关系代数表达式:

$$\pi_{ID, name}(\sigma_{building='Watson'}(instructor \bowtie department))$$

c. 找到至少选修过一门“Comp. Sci.”系的每位学生的 ID 和姓名。

- 需从 `course` 表找到 `dept_name` 为“Comp. Sci.”的课程, 然后结合 `takes` 和 `student` 表找到相关学生。

关系代数表达式:

$$\pi_{ID, name}(student \bowtie \sigma_{dept_name='Comp. Sci.'}(takes \bowtie course))$$

d. 找到在 2018 年至少选修过一门课程的每位学生的 ID 和姓名。

- 直接从 `takes` 表找到 `year = 2018` 的记录, 然后结合 `student` 表查询相关学生。

关系代数表达式：

$$\pi_{ID,name}(\sigma_{year=2018}(student \bowtie takes))$$

e. 找到在 2018 年没有选修过任何课程的每位学生的 ID 和姓名。

- 这个查询需要从 `student` 表中找到没有出现在 `takes` 表中 `year = 2018` 记录的学生。

关系代数表达式：

$$\pi_{ID,name}(student) - \pi_{ID,name}(\sigma_{year=2018}(student \bowtie takes))$$