

# 数据库模式

Figure 1: 员工数据库 (Employee Database)

| 关系名      | 属性   |
|----------|--|
| employee | pid (员工ID), person_name (员工姓名), street (街道), city (城市)                       |
| works    | person_name (员工姓名), pid (员工ID), company_name (公司名称), cid (公司ID), salary (薪资) |
| company  | cid (公司ID), company_name (公司名称), city (城市)                                   |

Figure 2: 银行数据库 (Bank Database)

| 关系名       | 属性  |
|-----------|---|
| branch    | branch_name (分行名称), branch_city (分行城市), assets (资产)                           |
| customer  | ID (客户ID), customer_name (客户姓名), customer_street (客户街道), customer_city (客户城市) |
| loan      | loan_number (贷款号), branch_name (分行名称), amount (金额)                            |
| borrower  | ID (客户ID), loan_number (贷款号)  |
| account   | account_number (账户号), branch_name (分行名称), balance (余额)                        |
| depositor | ID (客户ID), account_number (账户号)   |

Figure 3: 大学数据库 (University Database)

| 关系名        | 属性   | 主键                                      |
|------------|--|---|
| classroom  | building (教学楼), room_number (房间号), capacity (容量)   | (building, room_number)                 |
| department | dept_name (系名), building (教学楼), budget (预算)  | dept_name                               |
| course     | course_id (课程ID), title (课程名), dept_name (系名), credits (学分)  | course_id                               |
| instructor | ID (教师ID), name (姓名), dept_name (系名), salary (工资)  | ID                                      |
| section    | course_id (课程ID), sec_id (节次ID), semester (学期), year (年份), building (教学楼), room_number (房间号), time_slot_id (时间段ID) | (course_id, sec_id, semester, year)     |
| teaches    | ID (教师ID), course_id (课程ID), sec_id (节次ID), semester (学期), year (年份)   | (ID, course_id, sec_id, semester, year) |
| student    | ID (学生ID), name (姓名), dept_name (系名), tot_cred (总学分)   | ID                                      |
| takes      | ID (学生ID), course_id (课程ID), sec_id (节次ID), semester (学期), year (年份), grade (成绩)                                   | (ID, course_id, sec_id, semester, year) |
| advisor    | s_ID (学生ID), i_ID (教师ID)   | s_ID                                    |
| time_slot  | time_slot_id (时间段ID), day (星期), start_time (开始时间), end_time (结束时间)   | time_slot_id                            |
| prereq     | course_id (课程ID), prereq_id (先修课程ID)   | (course_id, prereq_id)                  |

# 题目

- 1.考虑图 1 中的员工数据库。哪些是适当的主键？
- 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。
  - 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。
- 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10000的贷款号。*b.找到每位有账户余额大于6000 的存款人的 ID。*
  - c. 找到每位在“Uptown”分行有账户余额大于\$6000 的存款人的 ID。
- 4.列举引入数据库中的空值的两个原因。
- 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。
  - b.找到位于“Watson”教学楼的每位教师的 ID 和姓名。
  - c.找到至少选修过一门“Comp. Sci.”系课程的每位学生的 ID 和姓名。
  - 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。
  - 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。

# 答案

## 问题1：考虑图1中的员工数据库。哪些是适当的主键？

对于Figure 1中的员工数据库，每个表的适当主键如下：

- **employee表**：主键是 pid （员工ID），因为员工ID唯一标识每个员工
- **works表**：主键是 (pid, cid) （员工ID + 公司ID的组合键），因为一个员工可能在多个公司工作，一个公司有多个员工，组合键确保唯一性
- **company表**：主键是 cid （公司ID），因为公司ID唯一标识每个公司

## 问题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

关系代数表达式：

$$\pi_{pid, person\_name}(employee \bowtie \pi_{pid}(works - \sigma_{company\_name='BigBank'}(works)))$$

解释：从works表中减去在"BigBank"工作的员工，然后与employee表连接，投影员工的ID和姓名。

### 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：

$$\pi_{pid, person\_name}(employee \bowtie \pi_{pid}(\sigma_{salary \geq salary2 \wedge pid \neq pid2}(works \times \rho_{pid2, person\_name2, company\_name2, cid2, salary2}(works))))$$

解释：通过笛卡尔积比较每个员工的薪资与其他员工的薪资，选择薪资大于等于其他员工薪资的员工（排除自己），然后与employee表连接投影ID和姓名。

### 问题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\$10000的贷款号

关系代数表达式：

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

解释：选择金额大于10000的贷款，并投影贷款号。

#### b. 找到每位有账户余额大于\$6000的存款人的ID

关系代数表达式：

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

解释：选择余额大于6000的账户，与存款人表连接，并投影存款人ID。

#### c. 找到每位在"Uptown"分行有账户余额大于\$6000的存款人的ID

关系代数表达式：

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

解释：选择分行名为"Uptown"且余额大于6000的账户，与存款人表连接，并投影存款人ID。

### 问题4：列举引入数据库中的空值的两个原因

引入空值（NULL）的原因主要包括：

1. **信息缺失**：当某个属性的值未知或暂时无法获取时，使用空值表示。例如，员工的生日未知。
2. **属性不适用**：当某个属性对特定元组不适用时，使用空值表示。例如，对于未婚员工，"配偶姓名"属性不适用。

### 问题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

关系代数表达式：

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

解释：选择部门为"Physics"的教师，并投影ID和姓名。

#### b. 找到位于"Watson"教学楼的每位教师的ID和姓名

关系代数表达式：

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

解释：教师表与部门表连接（基于dept\_name），选择教学楼为"Watson"的部门，并投影教师ID和姓名。

#### c. 找到至少选修过一门"Comp.Sci."系课程的每位学生的ID和姓名

关系代数表达式：

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

**解释：** 选择"Comp.Sci."部门的课程，与选课表连接得到学生ID，再与学生表连接得到学生ID和姓名。

#### **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**

**关系代数表达式：**

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

**解释：** 选择年份为2018的选课记录，投影学生ID，再与学生表连接得到学生ID和姓名。

#### **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**

**关系代数表达式：**

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

**解释：** 从所有学生ID和姓名中减去在2018年选课的学生ID和姓名，得到没有选课的学生ID和姓名。