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*)

Fig.0 大学数据库

1. 使用 Fig.0 的大学模式（university schema）编写的 SQL 查询：
   a) 查找每位至少选修过一门计算机科学课程的学生的 ID 和姓名，确保结果中没有重复的姓名。
   b) 查找每位未选修过 2017 年之前开设的任何课程的学生的 ID 和姓名。
   c) 对于每个系（department），找到该系教师的最高薪水。
   d) 找到在所有系（departments）中计算出的每个系最高薪水的最低值。
2. 考虑以下查询：

```
with dept_total (dept_name, value) as
        (select dept_name, sum(salary)
        from instructor
        group by dept_name),
dept_total_avg(value) as
        (select avg(value)
        from dept_total)
select dept_name
from dept_total, dept_total_avg
where dept_total.value >= dept_total_avg.value;
```

   请将此查询重新编写，而不使用 WITH 子句。
3. 不使用"unique"构造，重写 where 语句：

```
where unique (select title from course)
```

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*)

Fig. 1 银行数据库

4. 考虑 Fig.1 中的银行数据库，其中主键已经用下划线标记。构建以下针对这个关系数据库的 SQL 查询。
   a) 查找银行的每位顾客的 ID，他们拥有账户但没有贷款。
   b) 查找与顾客 '12345' 同住在相同街道和城市的每位顾客的 ID。
   c) 查找每个至少有一位在银行拥有账户并住在“Harrison”的顾客的分行的名称。
5. 考虑 Fig.1 中的银行数据库，其中主键已经用下划线标记。构建以下针对这个关系数据库的 SQL 查询。
   a) 找到每位在“Brooklyn”的每个分行都有账户的顾客。
   b) 找到银行中所有贷款金额的总和。
   c) 找到具有资产超过至少一个位于“Brooklyn”的分行的资产的所有分行名称。