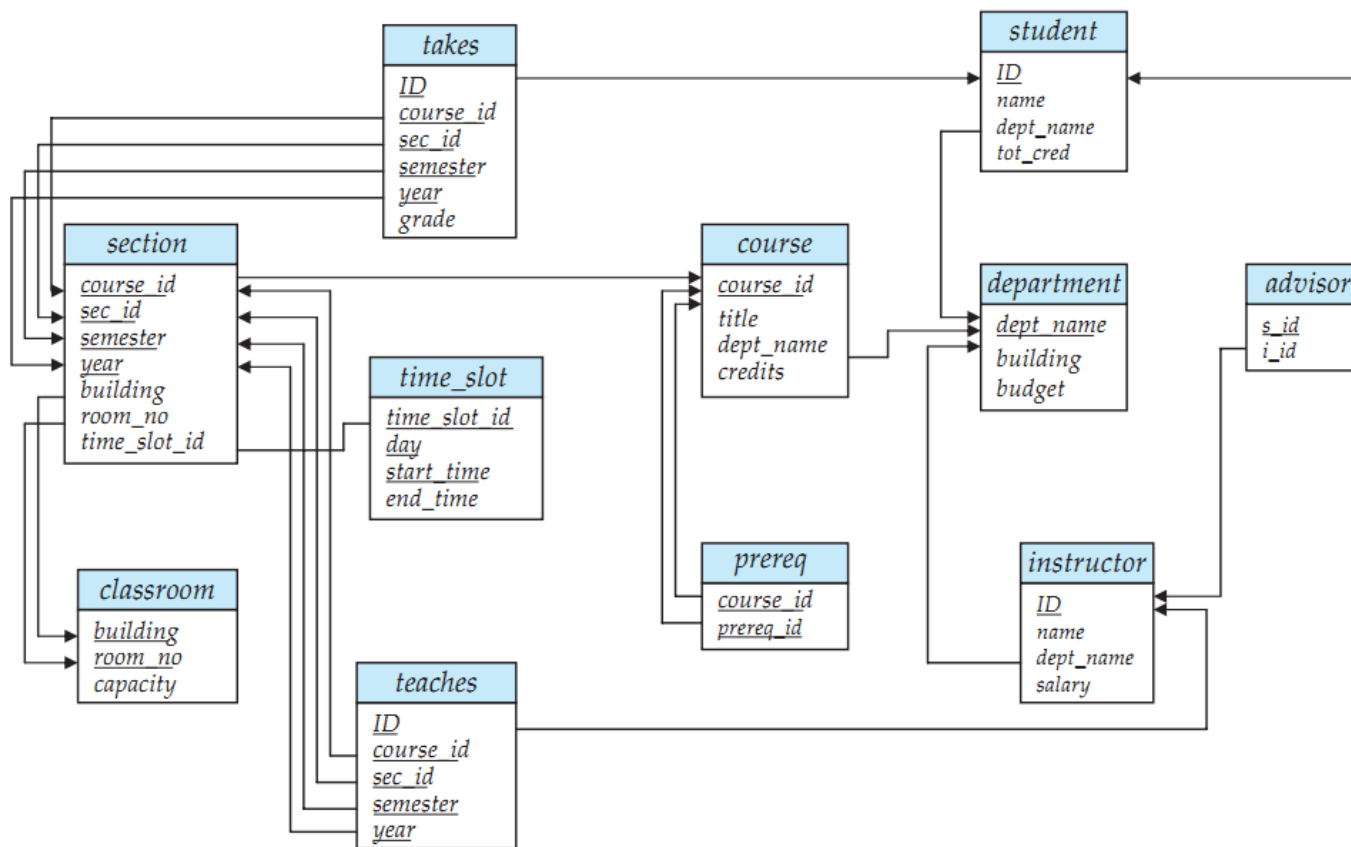


实验 3

仔细阅读下面的应用实例(大学机构的数据库设计)：（实验 3 与作业 3 合并，只需提交到励儒云）

- 大学分成多个系。每个系由自己唯一的名字（dept_name）来标识，坐落在特定的建筑物（building）中，有它的经费预算（budget）。
- 每一个系有一个开设课程列表。每门课程有课程号（course_id）、课程名（title）、系名（dept_name）和学分（credits），还可能有先修要求（prerequisites）。
- 教师由个人唯一的标识号（ID）来标识。每位教师有姓名（name）、所在的系（dept_name）和工资（salary）。
- 学生由个人唯一的标识号（ID）来标识。每位学生有姓名（name）、主修的系（dept_name）和已修学分数（tot_cred）。
- 大学维护一个教室列表，详细说明楼名（building）、房间号（room_number）和容量（capacity）。
- 大学维护开设的所有课程（开课）的列表。每次开课由课程号（course_id）、开课号（sec_id）、年（year）和学期（semester）来标识，与之相关联的有学期（semester）、年（year）、楼名（building）、房间号（room_number）和时段号（time_slot_id，即上课的时间）。
- 系有一个教学任务列表，说明每位教师的授课情况。在某开课年（year）和学期（semester）中某门开课号为（sec_id）的课程（course_id），由某开课教师（ID）开出。
- 大学有一个所有学生课程注册的列表，说明每位学生在哪些课程的哪次开课中注册了。学生（ID）在开课年开课学期注册了课程的某课次，经考试获得成绩。

对应模式图如下：（**建议使用 Web 版式视图，以显示全图**）



其对应的大学数据库模式如下：（红色部分为题目所涉及的关系）

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)

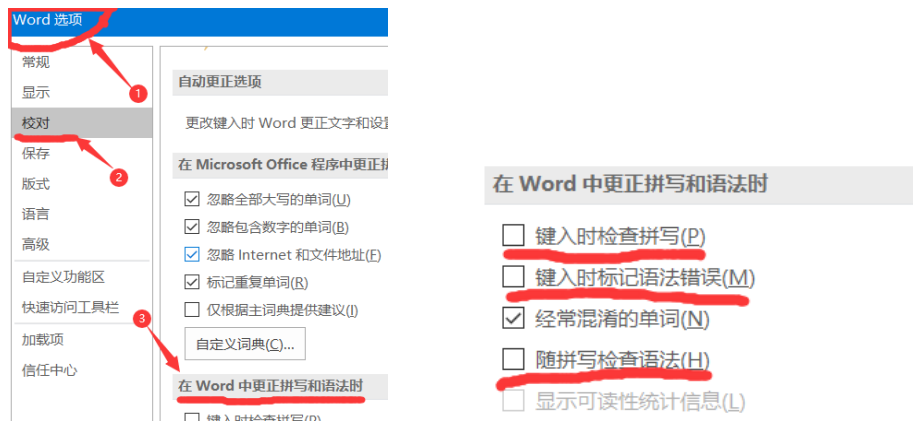
takes (ID , course id, sec id , semester, year , grade)

advisor (s_ID, i_ID)

time_slot (time slot id , day , start time, end_time)

prereq (course id , prereq id)

划线部分为关系的主码。（建议关闭语法错误提示，以增加可读性，方法如下）



上述关系中红色部分为题目所涉及的关系，请用 SQL 完成以下查询：

（数据库脚本在励儒云下载。

注意：有两个数据库，这两个数据库表相同，只是数据不同，有些查询在小数据集中才有结果，有些题目在大数据集中有结果。

共 SQL Server、MYSQL 两个版本, 每个版本各有两个文件：

❶一个数据库对象脚本（创建了两个数据库 college_big、college_small）

❷一个数据脚本，分别为上述两个数据库导入数据。

在查询分析器中先执行❶，再执行❷

）。

- a) 找出 Comp. Sci. 系开设的具有 3 个学分的课程名称。（请将 'Comp. Sci.' 复制到查询的条件中）

```
SELECT title
FROM course
WHERE credits=3 AND dept_name='Comp. Sci.'
```

- b) 找出名叫 Einstein 的教师所教的所有学生的标识，保证结果中没有重复。

```
SELECT s_ID
FROM advisor
WHERE i_ID IN (SELECT ID
FROM instructor
WHERE name='Einstein')
```

- c) 找出教师的最高工资。

```
SELECT MAX(salary)
FROM instructor
```

- d) 找出工资最高的所有教师（可能有不止一位教师具有相同的工资）。

```
SELECT *
```

```
FROM instructor
WHERE salary IN (SELECT MAX(salary)
FROM instructor)
```

- e) 找出 2009 年秋季开设的每个课程段的选课人数。

```
SELECT count(s_ID)
FROM advisor
WHERE i_ID IN (SELECT ID
FROM takes
WHERE YEAR='2009' AND semester='Fall')
```

- f) 从 2009 年秋季开设的所有课程段中，找出最多的选课人数。

- g) 找出在 2009 年秋季拥有最多选课人数的课程段。

- h) 找出所有至少选修了一门 Comp. Sci. 课程的学生姓名，保证结果中没有重复的姓名；

- i) 找出所有没有选修在 2009 年春季之前开设的任何课程的学生 ID 和姓名；

- j) 找出每个系教师的最高工资值。可以假设每个系至少有一位教师。

- k) 从前述查询 (j) 中所计算出的每个系最高工资中选出最低值。