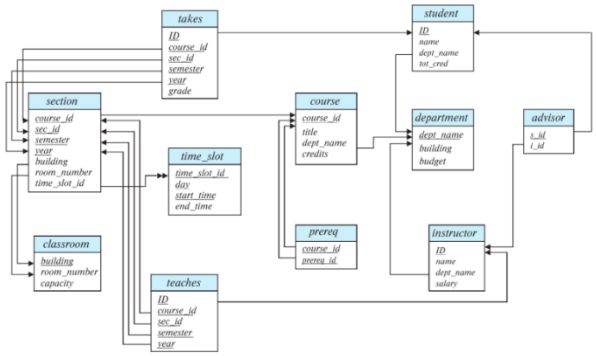
数据库测试

2025.4.17

42233100 林乐怡

一、university 数据库有以下关系模式：

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*)



1. 展示每个教师（ instructor ）的工号及其授课课程段（ section ）的数量。如果仅仅考虑授课的老师，请使用单表查询完成。

SQL：

```sql
SELECT id, COUNT(*) AS section_count
FROM teaches
GROUP BY id;
```

查询结果（部分）：

	id	section_count
1	77346	6
2	74420	6
3	14365	2
4	28400	2
5	95709	4
6	79081	6
7	43779	4
8	3199	4
9	4233	1
10	63287	2
11	41930	3
12	3335	2
13	80759	1
14	22591	13

2. 对于第 1 题，请确保即使没有授课的教师也要被输出。使用 JOIN 完成。

SQL：

```sql
SELECT instructor.id, COUNT(teaches.course_id) AS section_count
FROM instructor
LEFT JOIN teaches 1<->0..n: ON instructor.id = teaches.id
GROUP BY instructor.id;
```

查询结果（部分）：

	id	section_count
1	74426	0
2	4233	1
3	3335	2
4	90376	1
5	65931	1
6	78699	0
7	80759	1
8	15347	1
9	50330	1
10	63395	0
11	48507	1
12	64871	0
13	50885	0
14	42782	1
15	19368	3

3. 请使用标量子查询（scalar subquery）完成第 2 题。

SQL:

```sql
SELECT id,
       (SELECT COUNT(*)
        FROM teaches
        WHERE teaches.id = instructor.id) AS section_count
FROM instructor;
```

查询结果（部分）：

	id	section_count
1	63395	0
2	78699	0
3	96895	0
4	4233	1
5	4034	0
6	50885	0
7	79653	0
8	50330	1
9	80759	1
10	73623	1
11	97302	0
12	57180	0
13	74420	6
14	35579	0
15	31955	0

4. .解释为什么在 from 子句中追加 natural join section 并不会影响结果。

```sql
SELECT course_id, semester, year, sec_id, AVG(tot_cred)
FROM takes NATURAL JOIN student
WHERE year = 2017
GROUP BY course_id, semester, year, sec_id
HAVING count (ID) >= 2
```

原因：
上图查询结果为空，和追加 natural join section 后结果一致。查询的结果中没有来自 section 的列。在这个例子中，takes 与 section 的自然连接（基于 course_id, semester, year, sec_id）是一对一匹配，不改变行数、过滤条件或分组结果。

5. 使用 using 重写下面的查询：

```sql
SELECT *
FROM section NATURAL JOIN classroom;
```

找出 section 和 classroom 中相同的列：building 和 room_number，以达到代替 NATURAL JOIN 的效果。
使用 using 重写：

```sql
SELECT *
FROM section JOIN classroom 1..n<->1: USING (building, room_number);
```

二、应用题

考虑一个 emp_bonus 表，表示员工的奖金发送信息。其中，emp_no 表示员工工号 received 表示奖金发放日期， type 表示奖金类型，其中类型 1 表示其工资的 10%，类型 2 表示其工资的 20%，类型 3 表示其工资的 30%。

员工表 emp 的关系模式是 emp(emp_no, ename, sal, dept_no) ，分别是员工工号、姓名、工资和部门编号。

1. 创建两个关系，并添加测试数据，其中 emp_bonus 的内容严格按上表所示。
创建 emp_bonus 关系并添加数据（日期需要以特定形式存储，因此选择 VARCHAR（11）而不是 DATE。如果选择 DATE 则只能以类似 2005-03-17 存储）

```sql
CREATE TABLE emp_bonus(
    emp_no INT,
    received VARCHAR(11),
    type INT
);
INSERT INTO emp_bonus (emp_no, received, type) VALUES
( emp_no 7934, received '17-MAR-2005', type 1),
( emp_no 7934, received '15-FEB-2005', type 2),
( emp_no 7839, received '15-FEB-2005', type 3),
( emp_no 7782, received '15-FEB-2005', type 1);
```

	emp_no	received	type
1	7934	17-MAR-2005	1
2	7934	15-FEB-2005	2
3	7839	15-FEB-2005	3
4	7782	15-FEB-2005	1

创建 emp 关系并添加数据：

```
CREATE TABLE emp (
    emp_no INT PRIMARY KEY,
    ename VARCHAR(50),
    sal DECIMAL(10, 2),
    dept_no INT
);
INSERT INTO emp (emp_no, ename, sal, dept_no) VALUES
( emp_no 7934,  ename 'Miller',  sal 1300.00,  dept_no 42),
( emp_no 7839,  ename 'King',  sal 5000.00,  dept_no 42),
( emp_no 7782,  ename 'Clark',  sal 2450.00,  dept_no 42),
( emp_no 8000,  ename 'Smith',  sal 1000.00,  dept_no 10);
```

2. 请列出部门编号为 42 的所有员工的总工资及其总奖金

SQL：

```
SELECT e.dept_no,
       SUM(e.sal) AS total_salary, --计算总工资
       SUM(CASE
           WHEN b.type = 1 THEN e.sal * 0.10
           WHEN b.type = 2 THEN e.sal * 0.20
           WHEN b.type = 3 THEN e.sal * 0.30
           ELSE 0
       END) AS total_bonus -- 计算总奖金
FROM emp e
LEFT JOIN emp_bonus b
    ON e.emp_no = b.emp_no
WHERE e.dept_no = 42
GROUP BY e.dept_no;
```

查询结果：

	dept_no	total_salary	total_bonus
1	42	10050	2135