数据库原理第三次作业

42233099 曾慧鑫

1.1 新建一个 university 数据库，并执行 largeRelationsInsertFile.sql，导入数据。

```
postgres=# CREATE DATABASE university;
CREATE DATABASE
```

图 1-1

图 1-2

图 1-3

1.2 运行第 2 次作业的题目三代码。注意：把原题目中的会计改成 History。

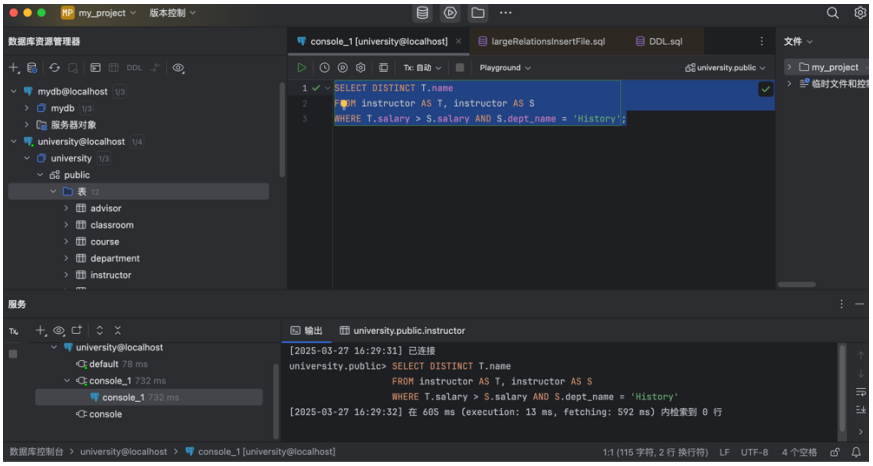

图 1-4

运行代码可以得到：在表中没有人比"History" 系的教师工资高或者在表中"History" 系没有教师。

2. 参考 Pattern Matching，在 PG 中使用至少三种方法实现找到所有以 S 开头教师的名字。

2.1 使用 LIKE

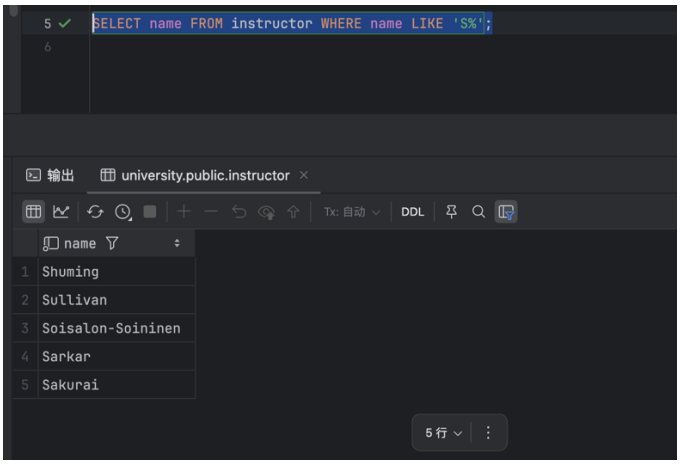

图 2-1

2.2 使用 SIMILAR TO 正则表达式

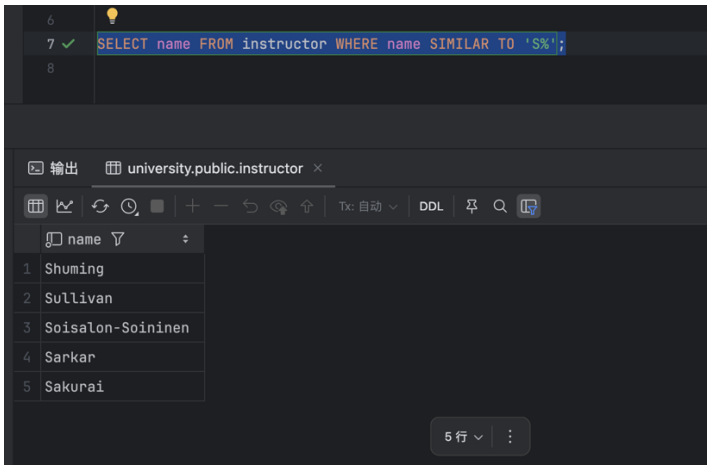

图 2-2

2.3 使用 POSIX 正则表达式

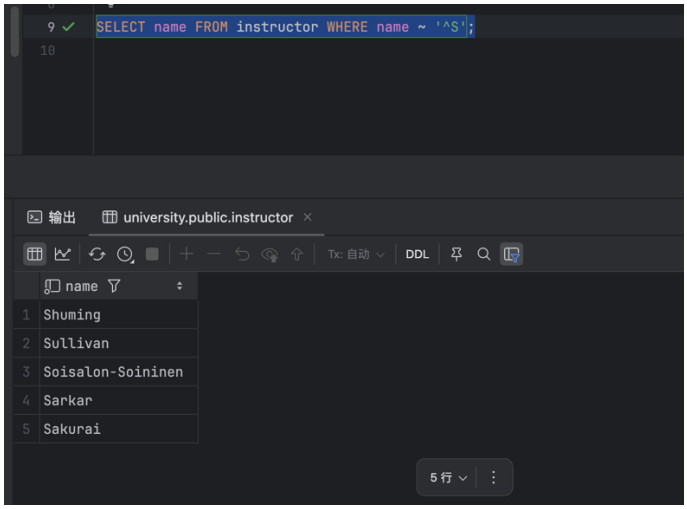

图 2-3

3.1 题目二已经实现，见图 2-1、2-2、2-3

3.2 列出所有的数据库

输入指令 \list

```
postgres=# \list
                                                    List of databases
    Name     |   Owner    | Encoding | Locale Provider |   Collate   |    Ctype    |
le | ICU Rules |   Access privileges
-------------+------------+----------+-----------------+-------------+-------------+-
---+-----------+-----------------------
 mydb        | zenghuixin | UTF8     | icu             | en_US.UTF-8 | en_US.UTF-8 |
S  |           |
 postgres    | postgres   | UTF8     | icu             | en_US.UTF-8 | en_US.UTF-8 |
S  |           |
 template0   | postgres   | UTF8     | icu             | en_US.UTF-8 | en_US.UTF-8 |
S  |           | =c/postgres          +
            |           |          |                 |             |             |
    |           | postgres=CTc/postgres
 template1   | postgres   | UTF8     | icu             | en_US.UTF-8 | en_US.UTF-8 |
S  |           | =c/postgres          +
            |           |          |                 |             |             |
    |           | postgres=CTc/postgres
 university  | zenghuixin | UTF8     | icu             | en_US.UTF-8 | en_US.UTF-8 |
S  |           |
 zenghuixin  | zenghuixin | UTF8     | icu             | en_US.UTF-8 | en_US.UTF-8 |
S  |           |
(6 rows)

:...skipping...
                                                    List of databases
    Name     |   Owner    | Encoding | Locale Provider |   Collate   |    Ctype    |
```

图 3-1

3.3 列出当前数据库的所有表

输入指令 \c university 确保链接 university 数据库

\dt

```
postgres=# \c university
You are now connected to database "university" as user "zenghuixin".
university=# \dt
           List of relations
 Schema |    Name     | Type  |  Owner
--------+-------------+-------+----------
 public | advisor     | table | postgres
 public | classroom   | table | postgres
 public | course      | table | postgres
 public | department  | table | postgres
 public | instructor  | table | postgres
 public | prereq      | table | postgres
 public | section     | table | postgres
 public | student     | table | postgres
 public | takes       | table | postgres
 public | teaches     | table | postgres
 public | time_slot   | table | postgres
 public | 表_name     | table | postgres
(12 rows)

university=#
```

图 3-2

3.4 显示某张表的关系模式

输入指令 \d instructor（表的名字）

```
university=# \d instructor
                    Table "public.instructor"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 id        | character varying(5)  |           | not null |
 name      | character varying(20) |           | not null |
 dept_name | character varying(20) |           |          |
 salary    | numeric(8,2)          |           |          |
Indexes:
    "instructor_pkey" PRIMARY KEY, btree (id)
Check constraints:
    "instructor_salary_check" CHECK (salary > 29000::numeric)
Foreign-key constraints:
    "instructor_dept_name_fkey" FOREIGN KEY (dept_name) REFERENCES department(dept_name) ON DELETE SET NULL
Referenced by:
    TABLE "advisor" CONSTRAINT "advisor_i_id_fkey" FOREIGN KEY (i_id) REFERENCES instructor(id) ON DELETE SET NULL
    TABLE "teaches" CONSTRAINT "teaches_id_fkey" FOREIGN KEY (id) REFERENCES instructor(id) ON DELETE CASCADE
```

图 3-3