数据科学与工程导论 Homework5

10235501435 张凯诚

1. 在新数据库中新建一张 user 表,插入几条数据,属性包含:唯一标识(id),姓名(name)性别(sex).年龄(age).联系方式(phone)，数据如下：('John Doe', 'Male', 25, '123-456-7890')，('Jane Smith', 'Female', 31, '987-654-3210')，('Bob Johnson', 'Male', 22, '555-123-4567')

```
postgres=# CREATE TABLE "user" (
postgres(#     id SERIAL PRIMARY KEY,
postgres(#     name VARCHAR(255),
postgres(#     sex VARCHAR(10),
postgres(#     age INT,
postgres(#     phone VARCHAR(20)
postgres(# );
CREATE TABLE
postgres=#
postgres=# INSERT INTO "user" (name, sex, age, phone) VALUES
postgres-# ('John Doe', 'Male', 25, '123-456-7890'),
postgres-# ('Jane Smith', 'Female', 31, '987-654-3210'),
postgres-# ('Bob Johnson', 'Male', 22, '555-123-4567');
INSERT 0 3
```

2. 写出 SQL 语句,查询 user 表中所有年龄在 20-30 范围内的用户

```
postgres=# SELECT * FROM "user" WHERE age BETWEEN 20 AND 30;
 id |    name     | sex  | age |    phone
----+-------------+------+-----+--------------
  1 | John Doe    | Male |  25 | 123-456-7890
  3 | Bob Johnson | Male |  22 | 555-123-4567
(2 行记录)
```

3. 写出 SQL 语句，向 user 表中添加自己的个人信息，并添加几条和你姓名同姓的虚拟信息

```
postgres=# INSERT INTO "user" (name, sex, age, phone) VALUES
postgres-# ('Zhang KaiCheng', 'Male', 19, '111-222-4444'),
postgres-# ('Zhang HaHa', 'Male', 28, '111-222-3333'),
postgres-# ('Zhang XiXi', 'Female', 24, '444-555-6666');
INSERT 0 3
```

4. 写出 SQL 语句,查询 user 表中年龄在 20-30 范围内,名字包含“你的姓氏”的用户,并按照年龄从大到小排序输出

```
postgres=# SELECT * FROM "user" WHERE age BETWEEN 20 AND 30 AND name LIKE 'Zhang%' ORDER BY age DESC;
 id |    name    |  sex   | age |    phone
----+------------+--------+-----+--------------
  5 | Zhang HaHa | Male   |  28 | 111-222-3333
  6 | Zhang XiXi | Female |  24 | 444-555-6666
(2 行记录)
```

5. 写出 SQL 语句,计算 user 表中所有用户的平均年龄

```
postgres=# SELECT AVG(age) AS average_age FROM "user";
     average_age
--------------------
 24.8333333333333333
(1 行记录)
```

6. 新建两张表 team 表(id,teamName)和 score 表(id,teamid,userid,score)。其中 score 表中的 teamid 为指向 team 表 id 的外键，userid 为指向 user 表 id 的外键

```
postgres=# CREATE TABLE "t" (
postgres(#     id SERIAL PRIMARY KEY,
postgres(#     teamName VARCHAR(255)
postgres(# );
CREATE TABLE
postgres=#
postgres=# CREATE TABLE "s" (
postgres(#     id SERIAL PRIMARY KEY,
postgres(#     teamid INT REFERENCES "t"(id),
postgres(#     userid INT REFERENCES "user"(id),
postgres(#     score INT
postgres(# );
CREATE TABLE
```

7. 在 team 表中和 score 表中插入合适的记录，写出 SQL 语句,查询 teamName 为“ECNU”的队伍中，年龄小于 20 的用户们，结果不得为空。

```
postgres=# INSERT INTO "t" (teamName) VALUES ('ECNU');
INSERT 0 1
postgres=# INSERT INTO "t" (teamName) VALUES ('SJU');
INSERT 0 1
postgres=# INSERT INTO "s" (teamid, userid, score) VALUES (1, 1, 70);
INSERT 0 1
postgres=# INSERT INTO "s" (teamid, userid, score) VALUES (1, 2, 90);
INSERT 0 1
postgres=# INSERT INTO "s" (teamid, userid, score) VALUES (2, 3, 80);
INSERT 0 1
postgres=# INSERT INTO "s" (teamid, userid, score) VALUES (1, 4, 60);
INSERT 0 1
postgres=# INSERT INTO "s" (teamid, userid, score) VALUES (2, 5, 60);
INSERT 0 1
postgres=# INSERT INTO "s" (teamid, userid, score) VALUES (1, 6, 60);
INSERT 0 1
postgres=# SELECT distinct u.*
postgres-# FROM "user" u
postgres-# JOIN "s" s ON u.id = s.userid
postgres-# JOIN "t" t ON t.id = s.teamid
postgres-# WHERE t.teamName = 'ECNU' AND u.age < 20;
 id |      name      | sex  | age |    phone
----+----------------+------+-----+-------------
  4 | Zhang KaiCheng | Male |  19 | 111-222-4444
(1 行记录)
```

8. 写出 SQL 语句,计算 teamName 为“ECNU”的总分(假设 score 存在 null 值,null 值

默认为 0 加入计算)。

```
postgres=# SELECT COALESCE(SUM(s.score), 0) AS total_score
postgres-# FROM "t" t
postgres-# JOIN "s" s ON t.id = s.teamid
postgres-# WHERE t.teamName = 'ECNU';
 total_score
-------------
         280
(1 行记录)
```

9. 写出 SQL 语句，删除 user 表中个人信息的记录。

```
postgres=# DELETE FROM "s" WHERE userid=4;
DELETE 1
postgres=# DELETE FROM "user" WHERE name = 'Zhang KaiCheng';
DELETE 1
```