0. 所有的题目结果中，给出SQL语句和执行结果。

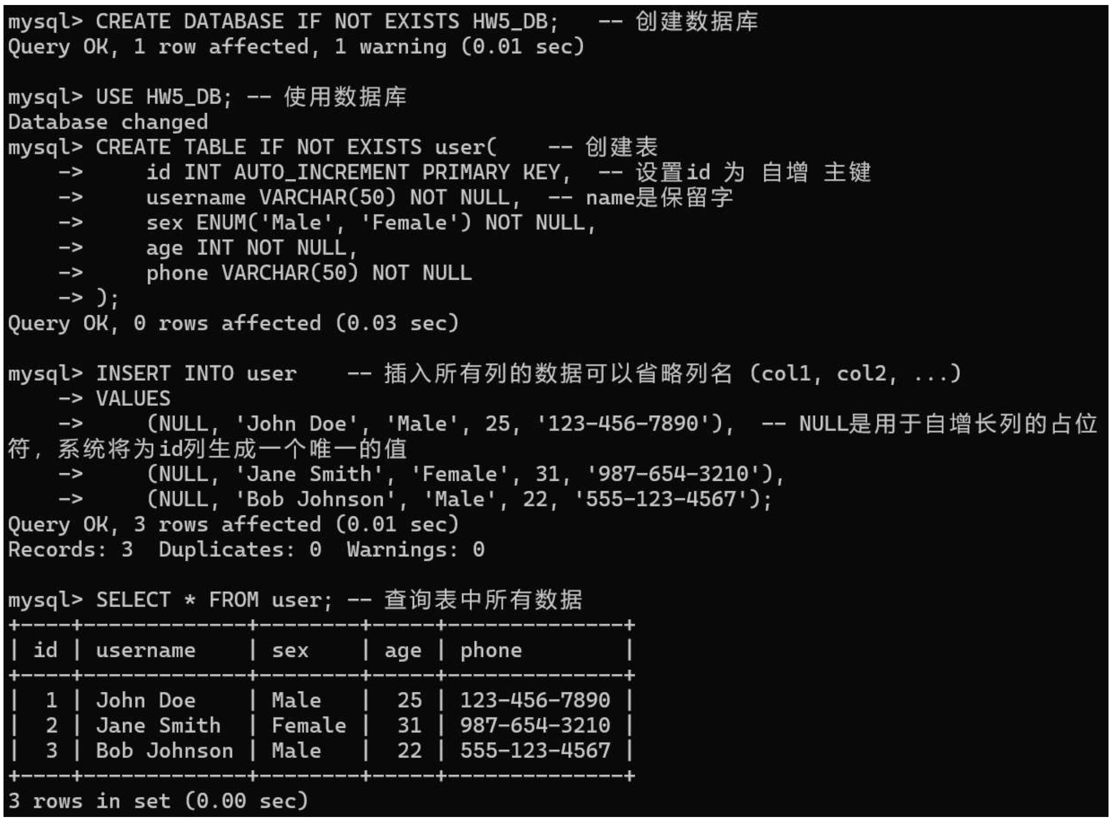
1. 在新数据库中新建一张 user 表,插入几条数据,属性包含:唯一标识(id),姓名(name)性别(sex).年龄(age).联系方式(phone)，数据如下:

   ('John Doe', 'Male', 25, '123-456-7890')

   ('Jane Smith', 'Female', 31, '987-654-3210')

   ('Bob Johnson', 'Male', 22, '555-123-4567')

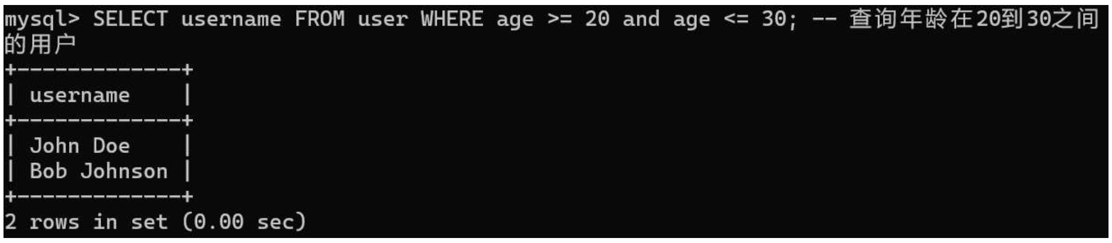
In [ ]:
```
CREATE DATABASE IF NOT EXISTS HW5_DB;   -- 创建数据库
USE HW5_DB; -- 使用数据库
CREATE TABLE IF NOT EXISTS user(    -- 创建表
    id INT AUTO_INCREMENT PRIMARY KEY,  -- 设置id 为 自增 主键
    username VARCHAR(50) NOT NULL,  -- name是保留字
    sex ENUM('Male', 'Female') NOT NULL,
    age INT NOT NULL,
    phone VARCHAR(50) NOT NULL
);
INSERT INTO user    -- 插入所有列的数据可以省略列名 (col1, col2, ...)
VALUES
    (NULL, 'John Doe', 'Male', 25, '123-456-7890'),  -- NULL是用于自增长列的占位
    (NULL, 'Jane Smith', 'Female', 31, '987-654-3210'),
    (NULL, 'Bob Johnson', 'Male', 22, '555-123-4567');
SELECT * FROM user; -- 查询表中所有数据
```



```
mysql> CREATE DATABASE IF NOT EXISTS HW5_DB;   -- 创建数据库
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> USE HW5_DB; -- 使用数据库
Database changed
mysql> CREATE TABLE IF NOT EXISTS user(    -- 创建表
    ->     id INT AUTO_INCREMENT PRIMARY KEY,  -- 设置id 为 自增 主键
    ->     username VARCHAR(50) NOT NULL,  -- name是保留字
    ->     sex ENUM('Male', 'Female') NOT NULL,
    ->     age INT NOT NULL,
    ->     phone VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO user    -- 插入所有列的数据可以省略列名 (col1, col2, ...)
    -> VALUES
    ->     (NULL, 'John Doe', 'Male', 25, '123-456-7890'),  -- NULL是用于自增长列的占位
符，系统将为id列生成一个唯一的值
    ->     (NULL, 'Jane Smith', 'Female', 31, '987-654-3210'),
    ->     (NULL, 'Bob Johnson', 'Male', 22, '555-123-4567');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM user; -- 查询表中所有数据
+----+-------------+--------+-----+--------------+
| id | username    | sex    | age | phone        |
+----+-------------+--------+-----+--------------+
|  1 | John Doe    | Male   |  25 | 123-456-7890 |
|  2 | Jane Smith  | Female |  31 | 987-654-3210 |
|  3 | Bob Johnson | Male   |  22 | 555-123-4567 |
+----+-------------+--------+-----+--------------+
3 rows in set (0.00 sec)
```

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

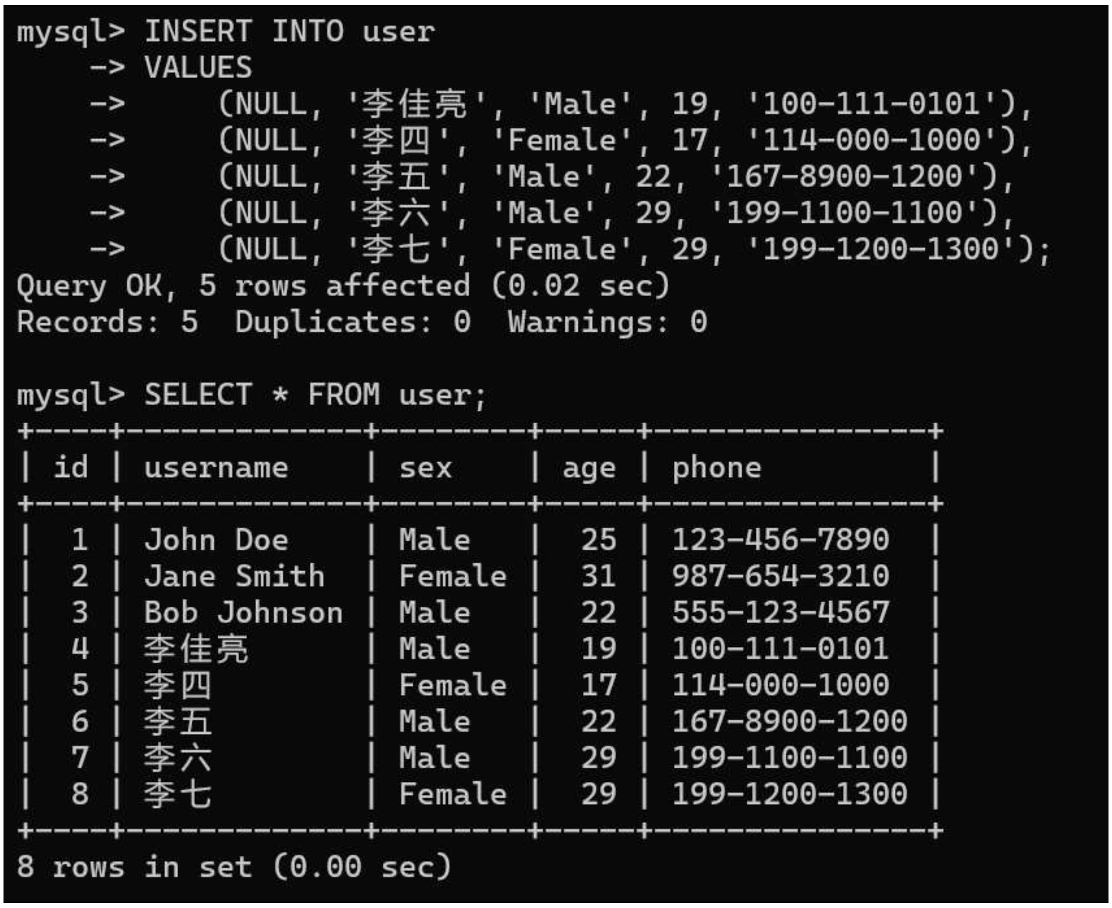
```
In [ ]: SELECT username FROM user WHERE age >= 20 and age <= 30; -- 查询年龄在20到30之间
```



```
mysql> SELECT username FROM user WHERE age >= 20 and age <= 30; -- 查询年龄在20到30之间
的用户
+-------------+
| username    |
+-------------+
| John Doe    |
| Bob Johnson |
+-------------+
2 rows in set (0.00 sec)
```

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。

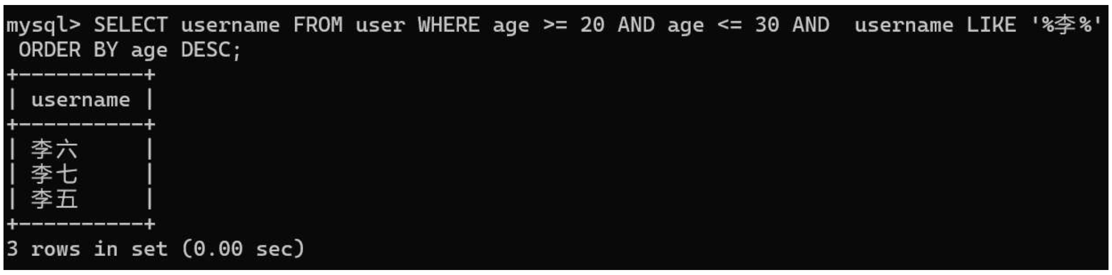
```
In [ ]: INSERT INTO user
        VALUES
            (NULL, '李佳亮', 'Male', 19, '100-111-0101'),
            (NULL, '李四', 'Female', 17, '114-000-1000'),
            (NULL, '李五', 'Male', 22, '167-8900-1200'),
            (NULL, '李六', 'Male', 29, '199-1100-1100'),
            (NULL, '李七', 'Female', 29, '199-1200-1300');
        SELECT * FROM user;
```



```
mysql> INSERT INTO user
    -> VALUES
    ->     (NULL, '李佳亮', 'Male', 19, '100-111-0101'),
    ->     (NULL, '李四', 'Female', 17, '114-000-1000'),
    ->     (NULL, '李五', 'Male', 22, '167-8900-1200'),
    ->     (NULL, '李六', 'Male', 29, '199-1100-1100'),
    ->     (NULL, '李七', 'Female', 29, '199-1200-1300');
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM user;
+----+-------------+--------+-----+---------------+
| id | username    | sex    | age | phone         |
+----+-------------+--------+-----+---------------+
|  1 | John Doe    | Male   |  25 | 123-456-7890  |
|  2 | Jane Smith  | Female |  31 | 987-654-3210  |
|  3 | Bob Johnson | Male   |  22 | 555-123-4567  |
|  4 | 李佳亮      | Male   |  19 | 100-111-0101  |
|  5 | 李四        | Female |  17 | 114-000-1000  |
|  6 | 李五        | Male   |  22 | 167-8900-1200 |
|  7 | 李六        | Male   |  29 | 199-1100-1100 |
|  8 | 李七        | Female |  29 | 199-1200-1300 |
+----+-------------+--------+-----+---------------+
8 rows in set (0.00 sec)
```

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

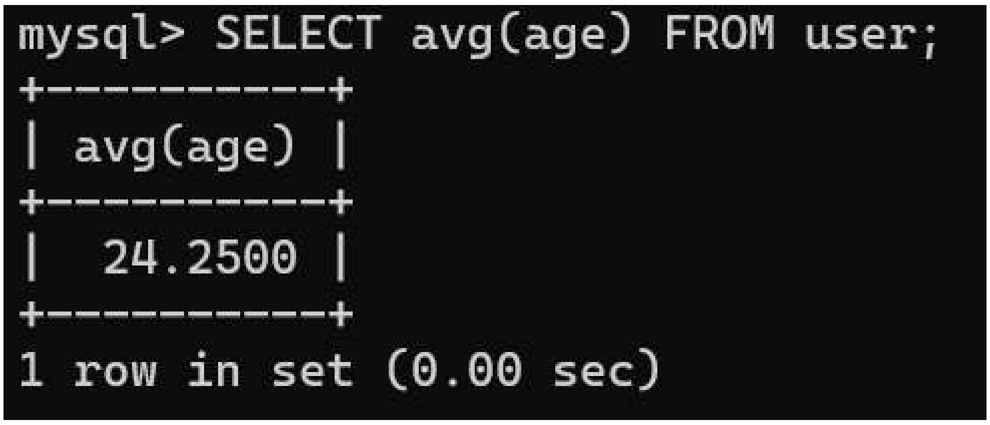
```
In [ ]: SELECT username FROM user WHERE age >= 20 AND age <= 30
        AND  username LIKE '%李%' ORDER BY age DESC;
        -- 查询年龄在20到30之间且名字含有“李”的用户
```



```
mysql> SELECT username FROM user WHERE age >= 20 AND age <= 30 AND  username LIKE '%李%'
 ORDER BY age DESC;
+----------+
| username |
+----------+
| 李六     |
| 李七     |
| 李五     |
+----------+
3 rows in set (0.00 sec)
```

5. 写出 SQL 语句,计算 user 表中所有用户的平均年龄

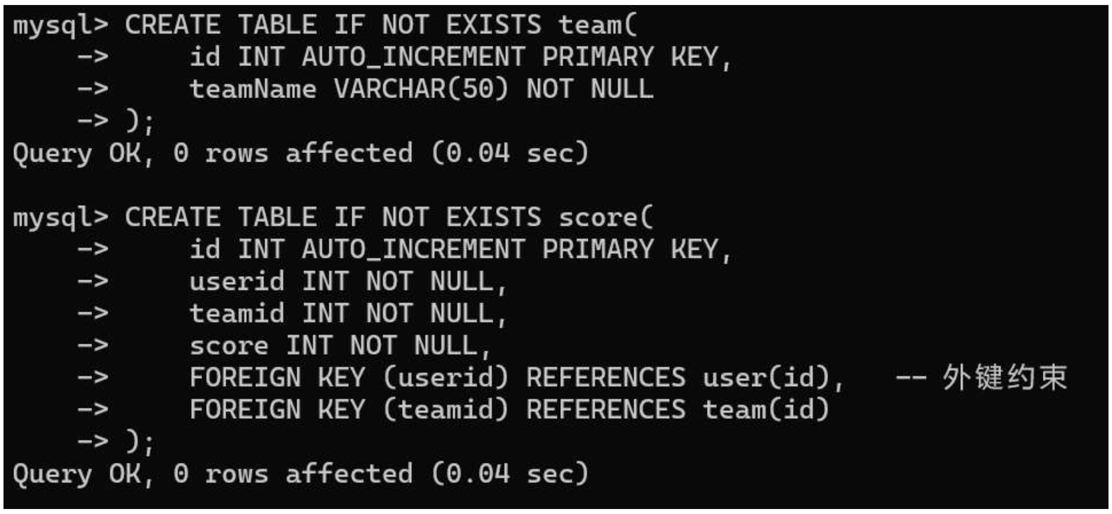
```
In [ ]: SELECT avg(age) FROM user;
```



```
mysql> SELECT avg(age) FROM user;
+----------+
| avg(age) |
+----------+
|  24.2500 |
+----------+
1 row in set (0.00 sec)
```

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

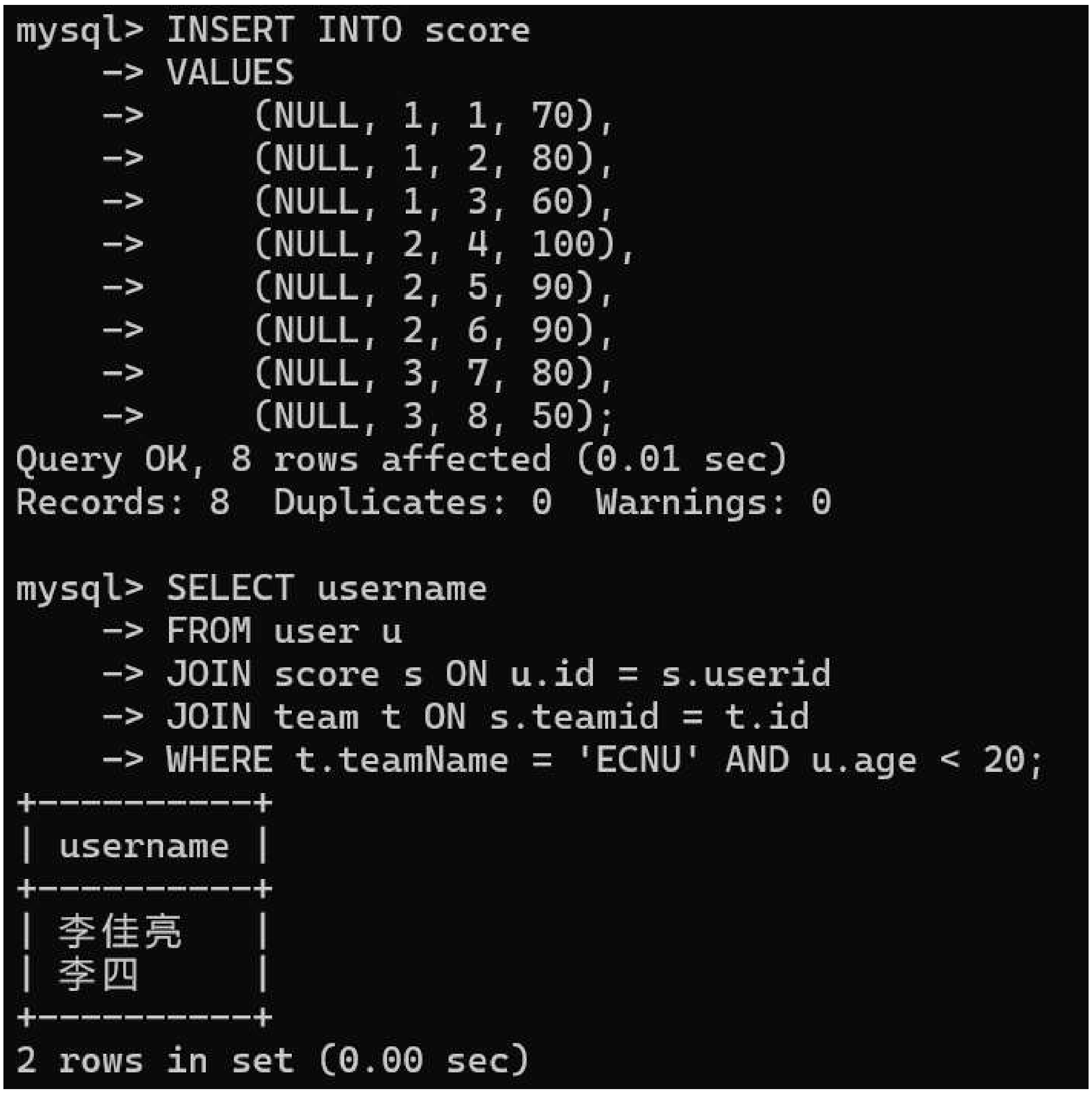
```
In [ ]: CREATE TABLE IF NOT EXISTS team(
            id INT AUTO_INCREMENT PRIMARY KEY,
            teamName VARCHAR(50) NOT NULL
        );
        CREATE TABLE IF NOT EXISTS score(
            id INT AUTO_INCREMENT PRIMARY KEY,
            teamid INT NOT NULL,
            userid INT NOT NULL,
            score INT NOT NULL,
            FOREIGN KEY (userid) REFERENCES user(id),   -- 外键
            FOREIGN KEY (teamid) REFERENCES team(id)
        );
```



```
mysql> CREATE TABLE IF NOT EXISTS team(
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     teamName VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE IF NOT EXISTS score(
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     userid INT NOT NULL,
    ->     teamid INT NOT NULL,
    ->     score INT NOT NULL,
    ->     FOREIGN KEY (userid) REFERENCES user(id),   -- 外键约束
    ->     FOREIGN KEY (teamid) REFERENCES team(id)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

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。

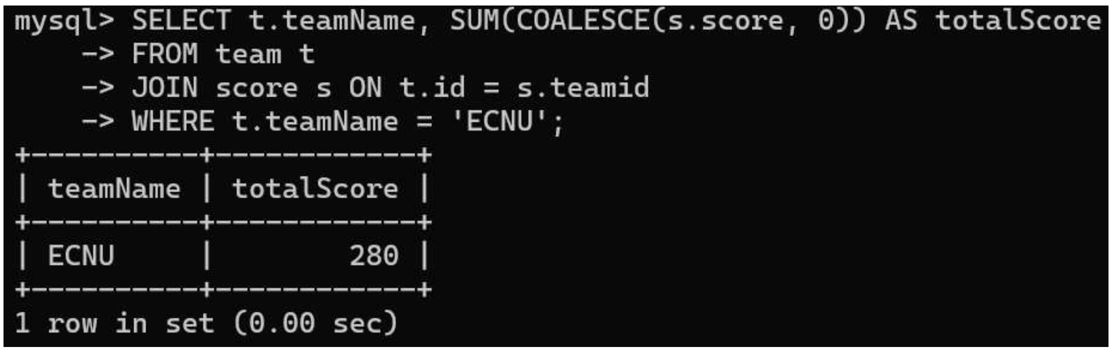
```
In [ ]:
INSERT INTO team
VALUES
    (NULL, 'THU'),
    (NULL, 'ECNU'),
    (NULL, 'SDU');
INSERT INTO score
VALUES
    (NULL, 1, 1, 70),
    (NULL, 1, 2, 80),
    (NULL, 1, 3, 60),
    (NULL, 2, 4, 100),
    (NULL, 2, 5, 90),
    (NULL, 2, 6, 90),
    (NULL, 3, 7, 80),
    (NULL, 3, 8, 50);
SELECT username
FROM user u
JOIN score s ON u.id = s.userid
JOIN team t ON s.teamid = t.id
WHERE t.teamName = 'ECNU' AND u.age < 20
```



```
mysql> INSERT INTO score
    -> VALUES
    ->     (NULL, 1, 1, 70),
    ->     (NULL, 1, 2, 80),
    ->     (NULL, 1, 3, 60),
    ->     (NULL, 2, 4, 100),
    ->     (NULL, 2, 5, 90),
    ->     (NULL, 2, 6, 90),
    ->     (NULL, 3, 7, 80),
    ->     (NULL, 3, 8, 50);
Query OK, 8 rows affected (0.01 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> SELECT username
    -> FROM user u
    -> JOIN score s ON u.id = s.userid
    -> JOIN team t ON s.teamid = t.id
    -> WHERE t.teamName = 'ECNU' AND u.age < 20;
+----------+
| username |
+----------+
| 李佳亮   |
| 李四     |
+----------+
2 rows in set (0.00 sec)
```

8. 写出 SQL 语句,计算 teamName为“ECNU”的总分(假设 score 存在 null值,nul值默认为 0 加入计算)。

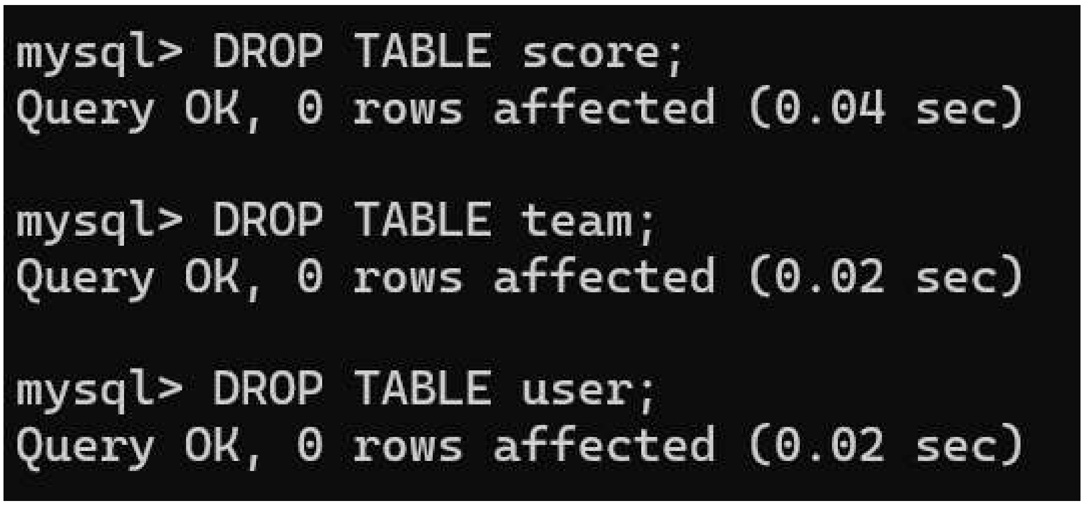
In [ ]:
```
-- COALESCE(s.score, 0): 如果s.score为NULL, 则返回0
SELECT t.teamName, SUM(COALESCE(s.score, 0)) AS totalScore
FROM team t
JOIN score s ON t.id = s.teamid
WHERE t.teamName = 'ECNU';
```



```
mysql> SELECT t.teamName, SUM(COALESCE(s.score, 0)) AS totalScore
    -> FROM team t
    -> JOIN score s ON t.id = s.teamid
    -> WHERE t.teamName = 'ECNU';
+----------+------------+
| teamName | totalScore |
+----------+------------+
| ECNU     |        280 |
+----------+------------+
1 row in set (0.00 sec)
```

9. 写出SQL语句，删除user表中个人信息的记录。

In [ ]:
```
DROP TABLE score;
DROP TABLE team;
DROP TABLE user;
```



```
mysql> DROP TABLE score;
Query OK, 0 rows affected (0.04 sec)

mysql> DROP TABLE team;
Query OK, 0 rows affected (0.02 sec)

mysql> DROP TABLE user;
Query OK, 0 rows affected (0.02 sec)
```