

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')

代码:

```
CREATE DATABASE test_db;
```

```
mysql> USE test_db;
```

Database changed

```
mysql> CREATE TABLE user (
```

```
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    ->     name VARCHAR(100) NOT NULL,
```

```
    ->     sex ENUM('Male', 'Female') NOT NULL,
```

```
    ->     age INT NOT NULL,
```

```
    ->     phone VARCHAR(15) NOT NULL
```

```
    -> );
```

```
mysql> INSERT INTO user (name, sex, age, phone)
VALUES

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

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

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

```
mysql> CREATE DATABASE test_db;
Query OK, 1 row affected (0.01 sec)

mysql> USE test_db;
Database changed

mysql> CREATE TABLE user (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     name VARCHAR(100) NOT NULL,
->     sex ENUM('Male', 'Female') NOT NULL,
->     age INT NOT NULL,
->     phone VARCHAR(15) NOT NULL
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO user (name, sex, age, phone) VALUES
-> ('John Doe', 'Male', 25, '123-456-7890'),
-> ('Jane Smith', 'Female', 31, '987-654-3210'),
-> ('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;
+----+-----+-----+-----+-----+
| id | name      | 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

代码：

SELECT *

-> FROM user

-> WHERE age BETWEEN 20 AND 30;

```
mysql> 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 |
+----+-----+-----+-----+-----+
```

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。

代码：

INSERT INTO user (name, sex, age, phone) VALUES

-> ('Ren Xuening', 'Female', 20, '123-456-7890'),

-> ('Ren Xing', 'Female', 29, '678-456-7890'),

-> ('Ren Zhangkuang', 'Male', 78, '678-000-7890'),

-> ('Ren Zhang', 'Male', 68, '678-000-7878');

```
mysql> INSERT INTO user (name, sex, age, phone) VALUES
-> ('Ren Xuening', 'Female', 20, '123-456-7890'),
-> ('Ren Xing', 'Female', 29, '678-456-7890'),
-> ('Ren Zhangkuang', 'Male', 78, '678-000-7890'),
-> ('Ren Zhang', 'Male', 68, '678-000-7878');
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM user;
```

id	name	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	Ren Xuening	Female	20	123-456-7890
5	Ren Xing	Female	29	678-456-7890
6	Ren Zhangkuang	Male	78	678-000-7890
7	Ren Zhang	Male	68	678-000-7878

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

代码:

SELECT *

-> FROM user

-> WHERE age BETWEEN 20 AND 30

-> AND name LIKE '%Ren%'

-> ORDER BY age DESC;

```
mysql> SELECT *
-> FROM user
-> WHERE age BETWEEN 20 AND 30
-> AND name LIKE '%Ren%'
-> ORDER BY age DESC;
```

id	name	sex	age	phone
5	Ren Xing	Female	29	678-456-7890
4	Ren Xuening	Female	20	123-456-7890

5.写出 SQL 语句,计算 user 表中所有用户的平均年龄

代码:

```
SELECT AVG(age) AS average_age
```

```
-> FROM user;
```

```
mysql> SELECT AVG(age) AS average_age
-> FROM user;
```

average_age
39.0000

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

代码:

```
CREATE TABLE team (
```

```
-> id INT PRIMARY KEY AUTO_INCREMENT,
```

```
->     teamName VARCHAR(255) NOT NULL
```

```
-> );
```

```
mysql> CREATE TABLE score (
```

```
->     id INT PRIMARY KEY AUTO_INCREMENT,
```

```
->     teamid INT,
```

```
->     userid INT,
```

```
->     score INT NOT NULL,
```

```
->     FOREIGN KEY (teamid) REFERENCES team(id),
```

```
->     FOREIGN KEY (userid) REFERENCES user(id)
```

```
-> );
```

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。

代码：

```
INSERT INTO team (teamName) VALUES ('ECNU');
```

```
INSERT INTO user (name, age) VALUES ('Alice', 19), ('Bob', 22), ('Charlie', 18), ('David', 21);
```

```
INSERT INTO score (teamid, userid, score) VALUES
```

```
-> (1,1,45),
```

```
-> (1,2,78),
```

```
-> (1,3,79),
```

```
-> (1,4,100),
```

```
-> (2,5,67),
```

```
-> (2,6,47),
```

```
-> (2,7,44),
```

```
-> (2,8,99),
```

```
-> (3,9,78),
```

```
-> (3,10,68),
```

```
-> (3,11,59);
```

```
INSERT INTO user (name, age) VALUES ('Ali', 19);
```

```
mysql> INSERT INTO score (teamid, userid, score) VALUES
```

```
-> (1,12,89);
```

```
mysql> SELECT u.name, u.age
```

```
-> 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> SELECT * FROM user;
+----+-----+-----+-----+-----+
| id | name       | 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  | Ren Xuening| Female| 20   | 123-456-7890 |
| 5  | Ren Xing   | Female| 29   | 678-456-7890 |
| 6  | Ren Zhangkuang| Male | 78   | 678-000-7890 |
| 7  | Ren Zhang  | Male  | 68   | 678-000-7878 |
| 8  | Alice      | Male  | 19   | NULL        |
| 9  | Bob        | Male  | 22   | NULL        |
| 10 | Charlie    | Male  | 18   | NULL        |
| 11 | David      | Male  | 21   | NULL        |
| 12 | Ali        | Male  | 19   | NULL        |
+----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> SELECT * FROM score;
+----+-----+-----+-----+
| id | teamid | userid | score |
+----+-----+-----+-----+
| 1  | 1      | 1      | 45    |
| 2  | 1      | 2      | 78    |
| 3  | 1      | 3      | 79    |
| 4  | 1      | 4      | 100   |
| 5  | 2      | 5      | 67    |
| 6  | 2      | 6      | 47    |
| 7  | 2      | 7      | 44    |
| 8  | 2      | 8      | 99    |
| 9  | 3      | 9      | 78    |
| 10 | 3      | 10     | 68    |
| 11 | 3      | 11     | 59    |
| 12 | 1      | 12     | 89    |
+----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> SELECT u.name, u.age
-> 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;
+-----+-----+
| name | age |
+-----+-----+
| Ali  | 19  |
+-----+-----+
```



```
mysql> SELECT * FROM team;
+----+-----+
| id | teamName |
+----+-----+
| 1  | ECNU     |
| 2  | JYSN     |
| 3  | KKHN     |
+----+-----+
```

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)。

代码:

```
SELECT SUM(COALESCE(score, 0)) AS total_score
```

-> FROM score

-> JOIN team ON score.teamid = team.id

-> WHERE team.teamName = 'ECNU';

```
mysql> SELECT SUM(COALESCE(score, 0)) AS total_score
-> FROM score
-> JOIN team ON score.teamid = team.id
-> WHERE team.teamName = 'ECNU';
+-----+
| total_score |
+-----+
|          391 |
+-----+
```

9.写出 SQL 语句, 删除 user 表中个人信息的记录。

代码:

```
DELETE FROM score
```

-> WHERE userid = 4;

```
mysql> DELETE FROM user
```

```
-> WHERE id = 4;
```

```
mysql> DELETE FROM score  
-> WHERE userid = 4;  
Query OK, 1 row affected (0.01 sec)
```

```
mysql>  
mysql> DELETE FROM user  
-> WHERE id = 4;  
Query OK, 1 row affected (0.01 sec)
```

```
mysql>  
mysql> SELECT * FROM user;
```

id	name	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
5	Ren Xing	Female	29	678-456-7890
6	Ren Zhangkuang	Male	78	678-000-7890
7	Ren Zhang	Male	68	678-000-7878
8	Alice	Male	19	NULL
9	Bob	Male	22	NULL
10	Charlie	Male	18	NULL
11	David	Male	21	NULL
12	Ali	Male	19	NULL