（一）SQL基础知识小练习

基础知识

- 选择所有列

> `SELECT * from xxx`

- 选择特定列

> `SELECT 特征1,特征2,... FROM 表xxxx`

- 使用 `WHERE` 子句进行条件筛选

> `SELECT * FROM titanic WHERE survived = 条件;`

- 多个条件的 `WHERE` 子句（`AND` 和 `OR`）

> `SELECT * FROM 表xxx`
>
> `WHERE survived = 1 AND pclass = 1 AND sex = 'female';`

- 使用 `BETWEEN` 关键字进行范围筛选

```
SELECT * FROM titanic

WHERE age BETWEEN 25 AND 40;

-- WHERE age BETWEEN 25 AND 40 表示选择 age 列值在 25 到 40 之间（包含 25 和 40）的行
```

- 使用 `ORDER BY` 子句进行排序

```
SELECT * FROM titanic

WHERE survived = 1

ORDER BY fare DESC, age ASC
```

```
-- ORDER BY fare DESC 表示先按 fare 列的值从大到小排序（降序）
-- 如果 fare 值相同，再按 age 列的值从小到大排序（升序）
```

- 关于空值判断（`IS NOT NULL` or `IS Null`）

```
SELECT *
FROM titanic
WHERE age IS NOT NULL;
-- WHERE age IS NOT NULL 表示只选择 age 列值不为 NULL 的行
```

背景介绍

泰坦尼克号事件背景

> 泰坦尼克号是当时世界上体积最庞大、内部设施最豪华的客运轮船，有“永不沉没”的美誉。然而，在1912 年 4 月 10 日，泰坦尼克号从英国南安普顿出发，开启了它的处女航，目的地是美国纽约。4 月 14 日 23 时 40 分左右，泰坦尼克号在北大西洋撞上冰山，两小时四十分钟后，即 4 月 15 日凌晨 2 点 20 分左右，泰坦尼克号完全沉没。这次海难是和平时期死伤人数最为惨重的一次海难之一，船上 2224 名船员及乘客中，1517 人丧生。

数据集的形成

> 泰坦尼克号事件发生后，相关人员记录了部分乘客和船员的信息，这些信息后来被整理成了泰坦尼克号数据集。数据集包含了多个与乘客相关的特征，通过对这些特征的分析，可以深入了解不同类型乘客在此次灾难中的生存情况、分布规律等。

数据集常见字段及其含义

- `survived`：表示乘客是否存活，1 代表存活，0 代表未存活。
- `pclass`：客舱等级，分为 1 等舱、2 等舱和 3 等舱，反映了乘客的社会经济地位，1 为最高等级。
- `sex`：乘客的性别，'male' 表示男性，'female' 表示女性。

- `age`：乘客的年龄，部分数据可能存在缺失值。年龄在分析生存情况时是一个重要因素，比如儿童和老人可能在逃生过程中有不同的待遇和结果。
- `sibsp`：在船上的兄弟姐妹或配偶的数量，用于衡量乘客的家庭关系中同辈的情况。
- `parch`：在船上的父母或子女的数量，体现乘客家庭关系中长辈或晚辈的情况。
- `fare`：乘客购买船票的票价，一定程度上也能反映乘客的经济实力。
- `embarked`：乘客登船的港口，主要有 'S'（南安普顿 Southampton）、'C'（瑟堡 Cherbourg）和 'Q'（皇后镇 Queenstown）。
- `class`：与 `pclass` 类似，以文本形式表示客舱等级，如 'First'（一等舱）、'Second'（二等舱）、'Third'（三等舱）。
- `who`：对乘客身份的一种描述，如 'man'（成年男性）、'woman'（成年女性）、'child'（儿童）。
- `adult_male`：布尔类型（0 或 1），表示是否为成年男性。
- `deck`：乘客所在的甲板，部分数据有缺失，不同甲板位置可能影响逃生的难易程度。
- `embark_town`：登船城镇的完整名称。
- `alive`：以文本形式表示乘客是否存活，'yes' 表示存活，'no' 表示未存活。
- `alone`：布尔类型（0 或 1），表示乘客是否独自旅行。

> 通过对泰坦尼克号数据集的分析，人们可以探究诸如性别、社会阶层、年龄等因素对生存几率的影响，这不仅在数据科学领域有重要意义，也能从历史角度让我们更深入地了解这场悲剧。

1. 查询全部数据

编写 SQL 查询语句，从 titanic 表中获取所有乘客的全部信息。

```
SELECT * FROM titanic
```

2.查询乘客的性别信息

编写 SQL 查询语句，只获取所有乘客的性别（sex）信息

```

```

```
SELECT sex FROM titanic;
```

3. 查询年龄在 20 岁及以上的男性乘客信息

编写 SQL 查询语句，获取年龄（age）大于等于 20 岁且性别为男性（sex 为 'male'）的乘客的所有信息。

```
SELECT * FROM titanic

WHERE age >= 20 AND sex = 'male';
```

4.查询在 Cherbourg 港口（embark_town 为 'Cherbourg'）登船的女性乘客的姓名

```
SELECT * FROM titanic

WHERE embark_town = 'Cherbourg' AND sex = 'female';
```

5. 查询票价（fare）在 50 到 100 之间的乘客信息

```
SELECT * FROM titanic

WHERE fare BETWEEN 50 AND 100;
```

6.按票价降序查询所有乘客信息

编写 SQL 查询语句，从 titanic 表中获取所有乘客的全部信息，并按照票价（fare）从高到低进行排序。

```
SELECT * FROM titanic

ORDER BY fare DESC;
```

7.按票价降序和年龄升序查询存活乘客信息

编写 SQL 查询语句，从 `titanic` 表中获取存活（`survived` 为 1）乘客的全部信息，先按照票价（`fare`）从高到低排序，如果票价相同则按照年龄（`age`）从小到大排序。

```
SELECT * FROM titanic

WHERE survived = 1

ORDER BY fare DESC, age ASC;
```

8. 按多种条件筛选并按复杂规则排序

编写 SQL 查询语句，获取存活（`survived` 为 1）、年龄大于 20 岁且在 Queenstown 港口（`embark_town` 为 'Queenstown'）登船的乘客信息，先按舱位等级（`pclass`）从低到高排序，舱位等级相同的按票价（`fare`）从高到低排序。

```
SELECT * FROM titanic

WHERE survived = 1 AND age > 20 AND embark_town = 'Queenstown'

ORDER BY pclass, fare DESC;
```

9.查询满足特定身份和年龄范围的乘客信息并排序

编写 SQL 查询语句，获取成年男性（`adult_male` 为 1）且年龄在 33 到 58 岁之间的乘客信息，按是否存活（`survived`）从高到低排序，若存活情况相同则按票价（`fare`）从大到小排序。

```
SELECT * FROM titanic

WHERE adult_male = 1 AND age BETWEEN 33 AND 58

ORDER BY survived DESC, fare DESC;
```