



MySQL

的 安 装 与 使 用

计算机科学与技术学院

2352018 刘彦

2024 年 9 月 16 日

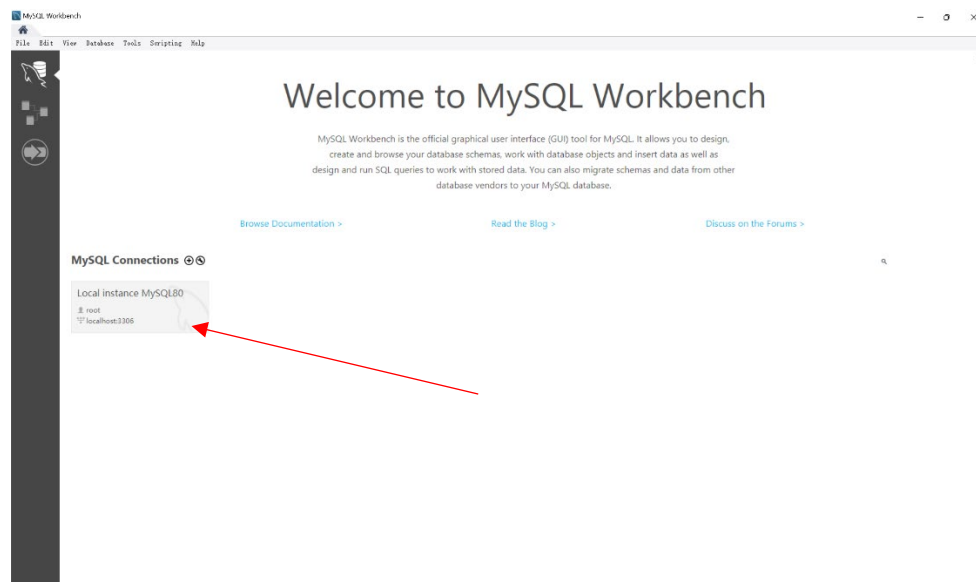
一、实验步骤

1. 初始界面

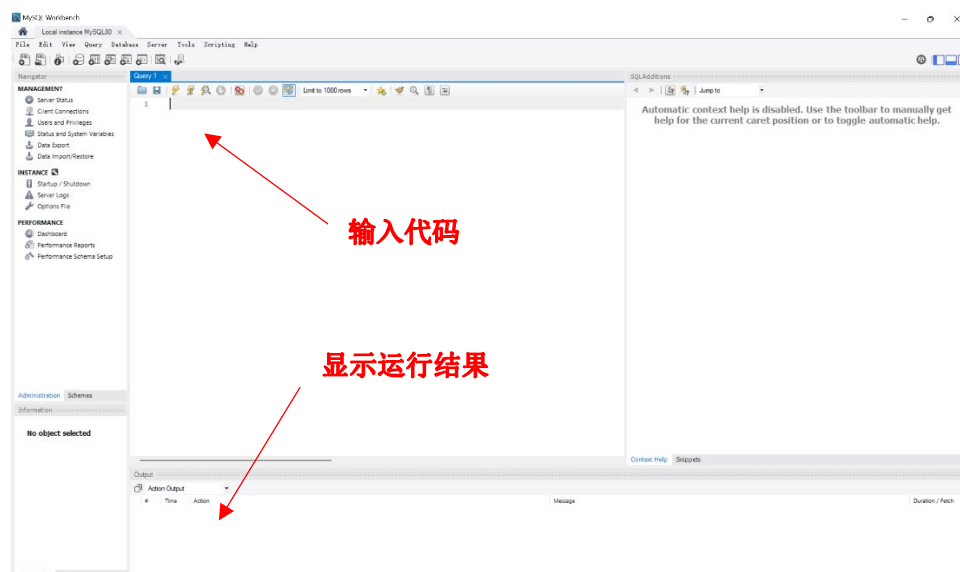
(1) 下载后，桌面出现这样的快捷方式，打开 MySQL Workbench。



(2) 进入初始界面：



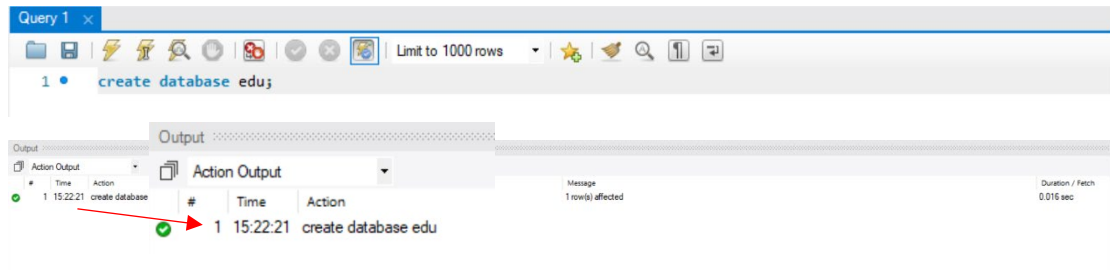
(3) 输入密码后进入



2. 数据库

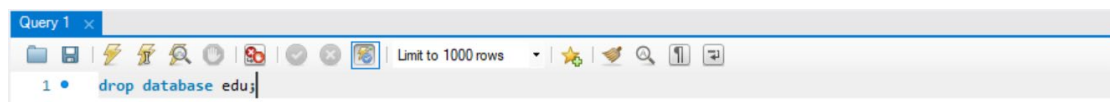
(1) 数据库建立

create database



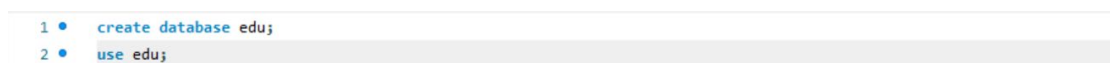
(2) 数据库删除

drop database



(3) 使用数据库

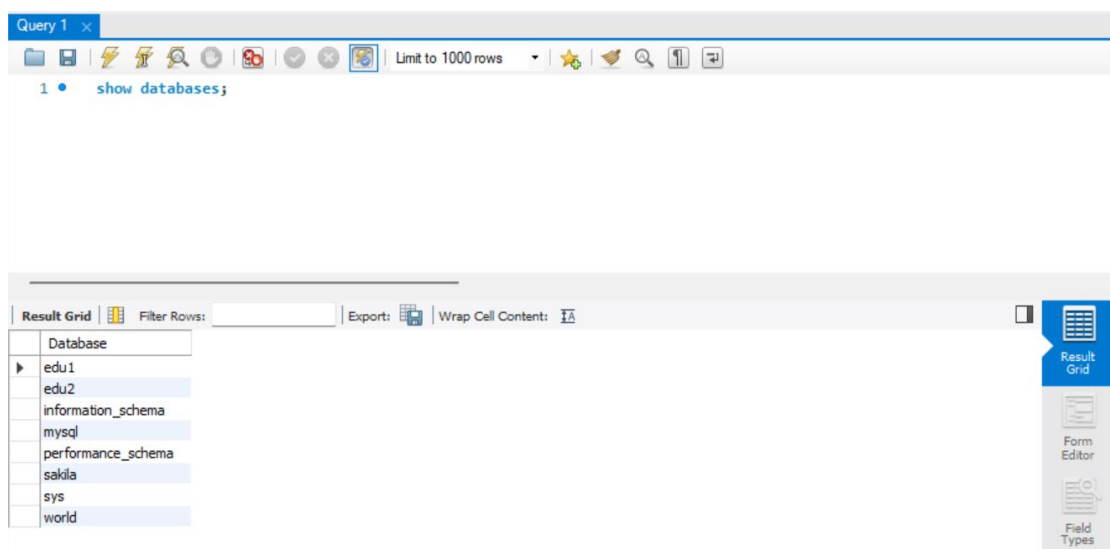
use database



必须先使用数据库后，才能在数据库中进行其他操作。

(4) 查询已有数据库

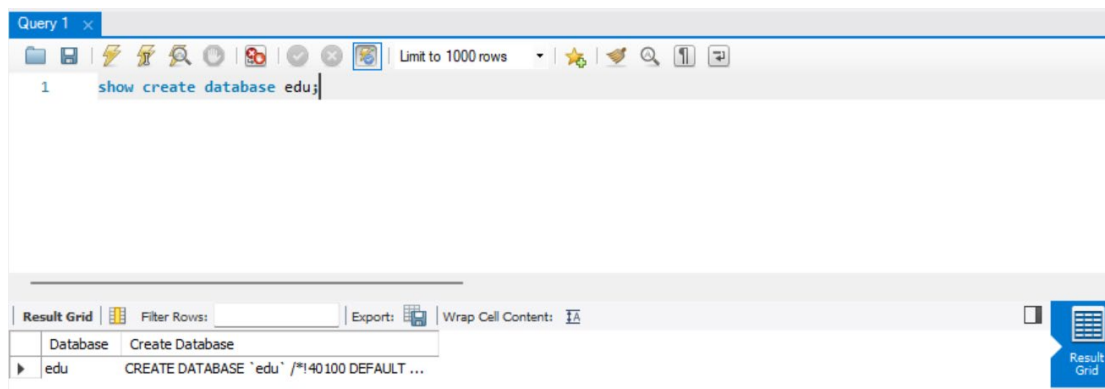
show database



显示所有存在的数据库，包括系统自带的。

(5) 查询指定数据库

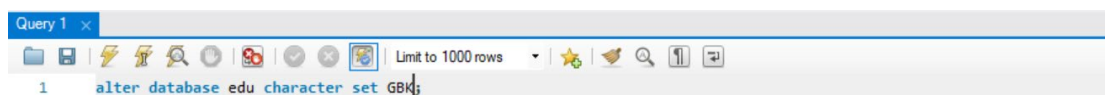
`show create database`



显示查询的数据库。

(6) 设置数据库字符类型

`alter database ... character set ...`



设置字符类型为 GBK。

3. 数据表

(1) 数据表建立

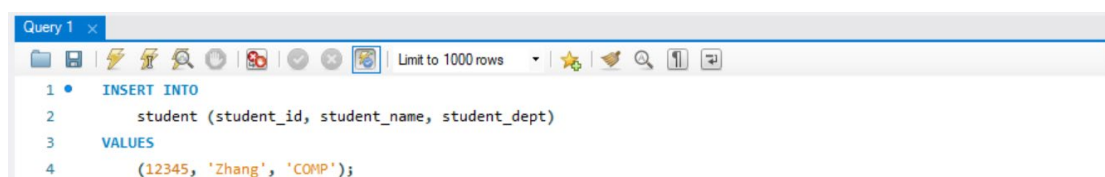
`create table`

以一个名为 student 的 table 为例，含有 3 个字段：



(2) 插入记录

`insert into... values...`



插入时要写好 table 和 table 中的字段，在 values 中标清它们的值。

(3) 替换记录

update ... set ... where ...

```
Query 1
1 • update
2   student
3   set
4     student_name = 'Zhang San'
5   where
6     student_id = 12345
```

执行完此语句后：

Result Grid

student_id	student_name
12345	Zhang
NULL	NULL

Result Grid

student_id	student_name
12345	Zhang San
NULL	NULL

(4) 删除记录

delete from ... where ...

会删除所选数据，前面加上 SET SQL_SAFE_UPDATES = 0; 用于关闭 MySQL 数据库中的安全更新模式

```
Query 1
1 • SET SQL_SAFE_UPDATES = 0;
2 • DELETE FROM
3   student
4   WHERE
5     student_name = 'Li';
```

执行完此语句后：

student_id student_name

10000	Li
12340	Wang
12341	Zhang
12345	Zhang
NULL	NULL

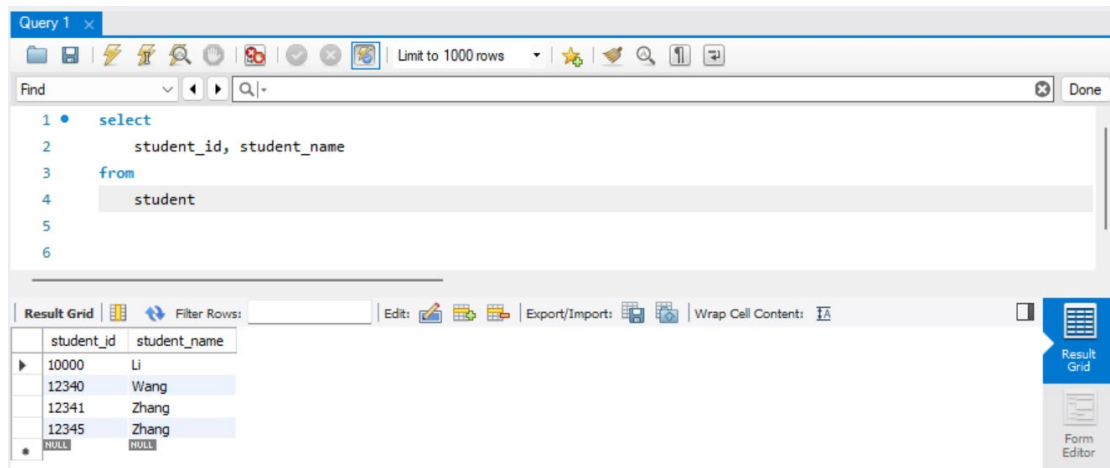
student_id student_name

12340	Wang
12341	Zhang
12345	Zhang
NULL	NULL

(5) 查询记录

① 遍历查询

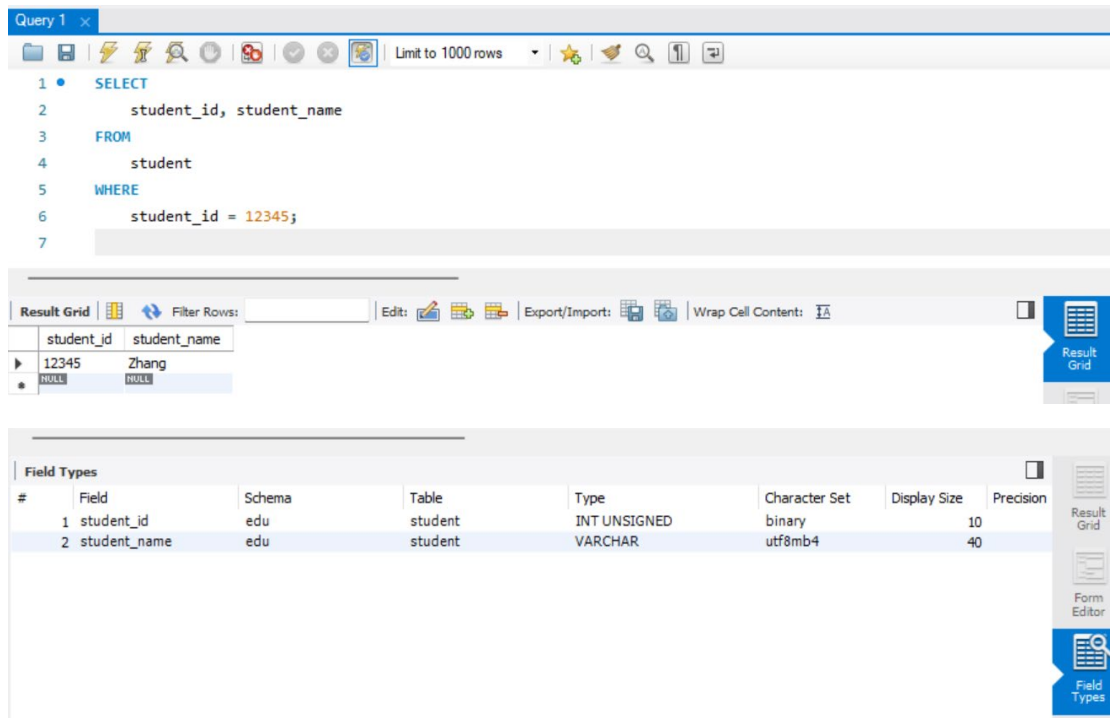
`select ... from...`



② 遍历查询

`select ... from ... where ...`

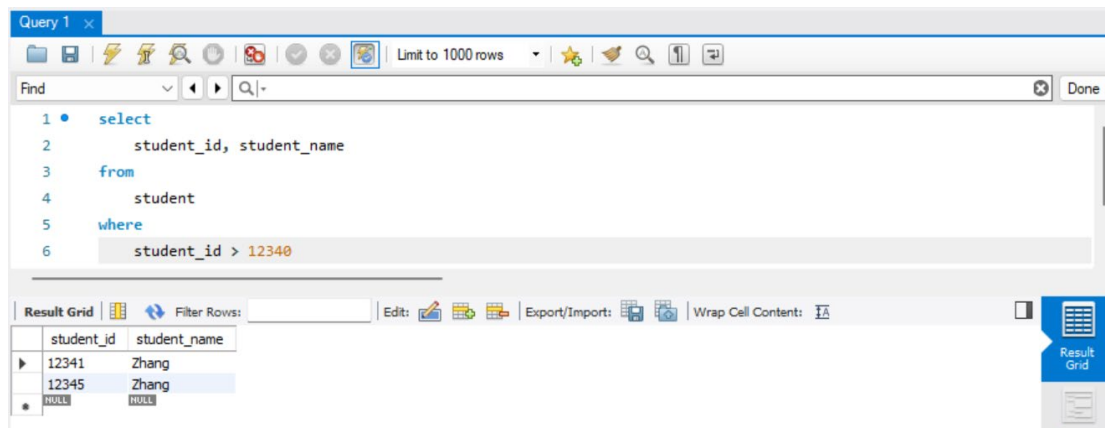
输入字段的值，找到这一条数据（可以包含这条数据中的其它内容）。



③ 设定范围查询

`select ... from ... where ...`

可以设定查询数据的范围，找出所有满足条件的数据。

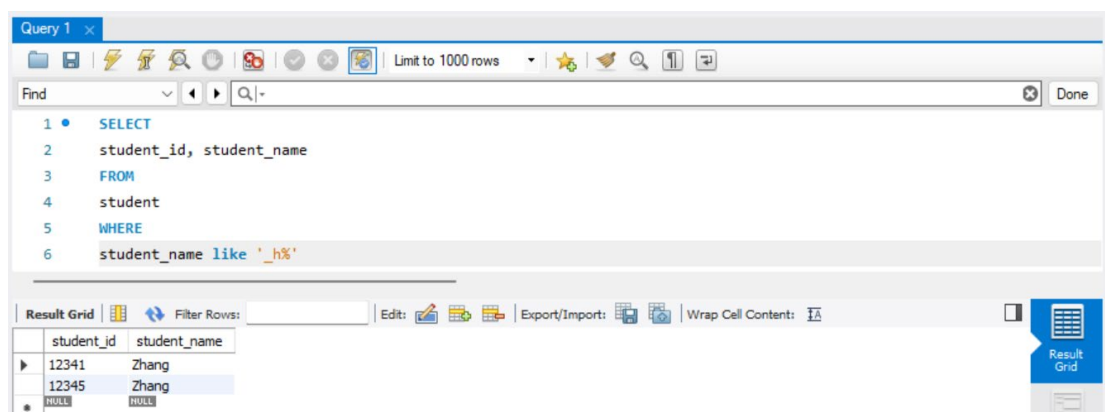
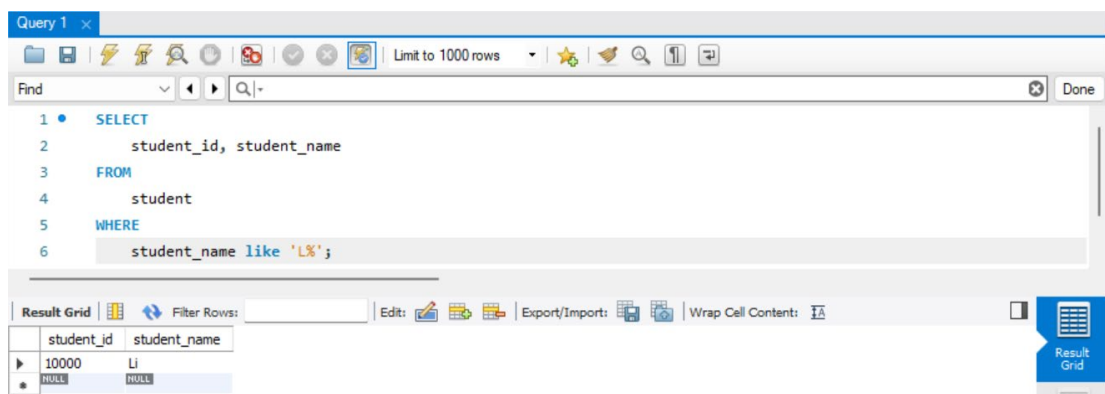


④ 模糊查询

select ... from ... where ...like...

‘X%’ 为查询 X 开头的数据。

‘_Y%’ 为查询 Y 为第二位的数据。

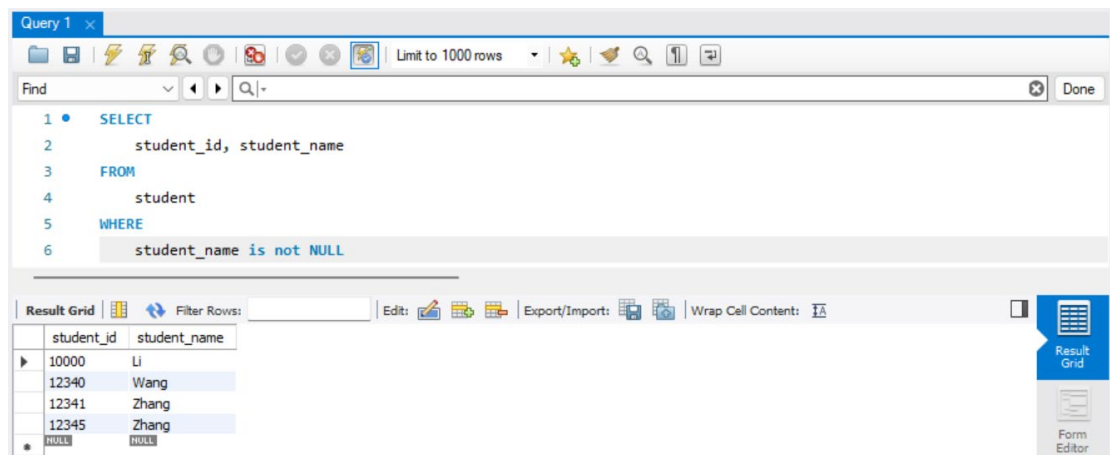
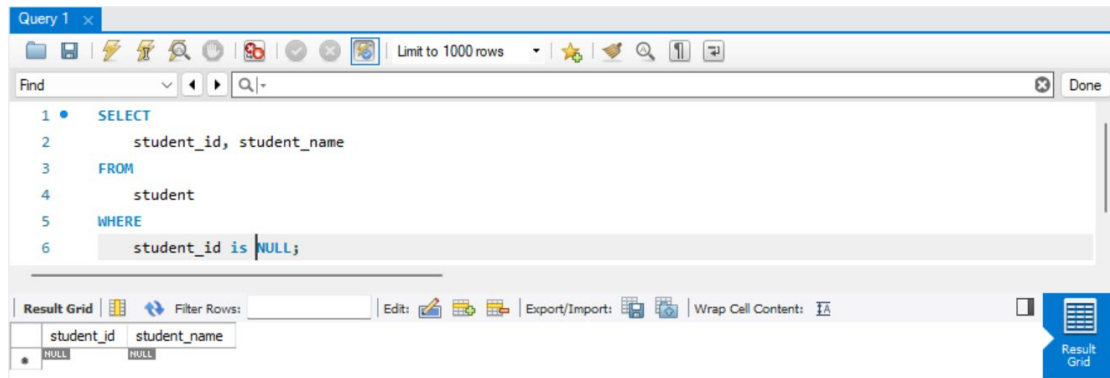


⑤ 查询是否为 NULL

select ... from ... where ...is NULL

select ... from ... where ...is not NULL

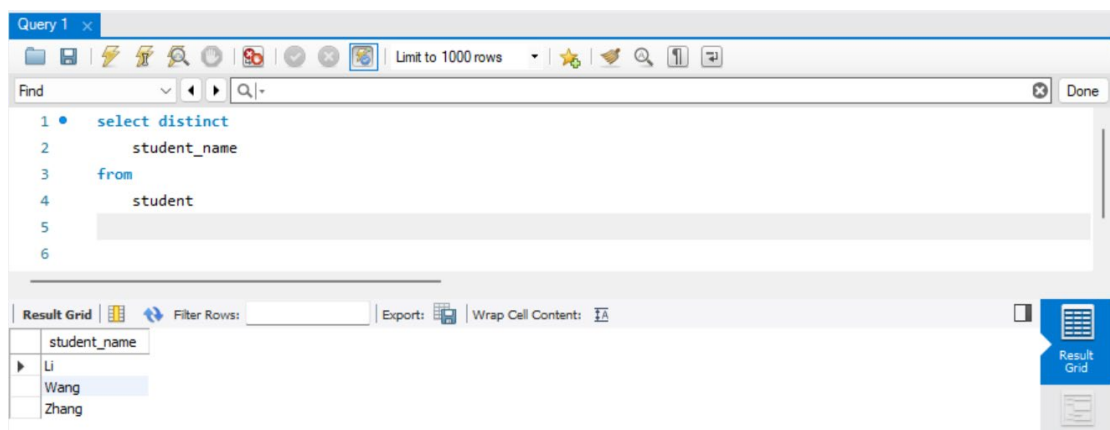
用来查找是否为空值。



⑥ 去除重复记录查询

select distinct ... from ...

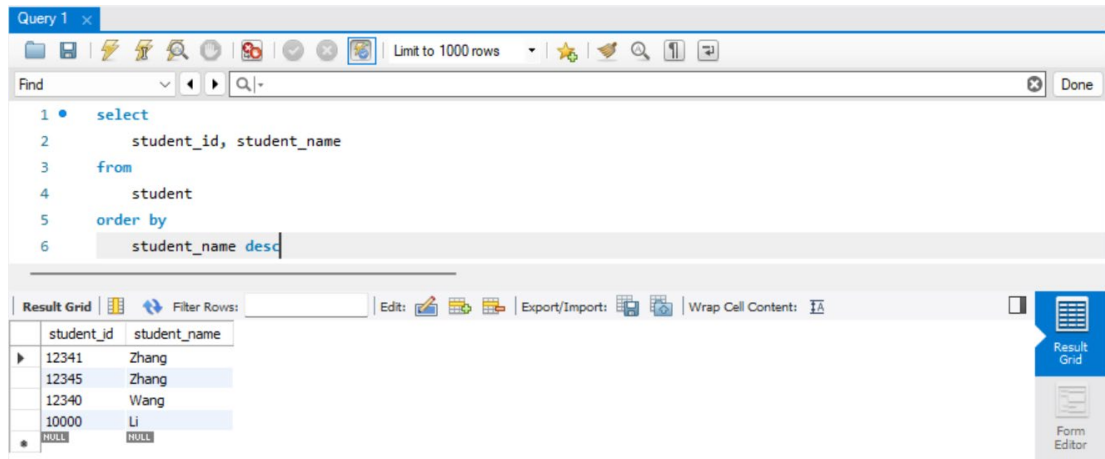
列出此字段所有不相同数据。



⑦ 排序查询

select ... from ... order by ...

默认为升序，加上 desc 为降序。



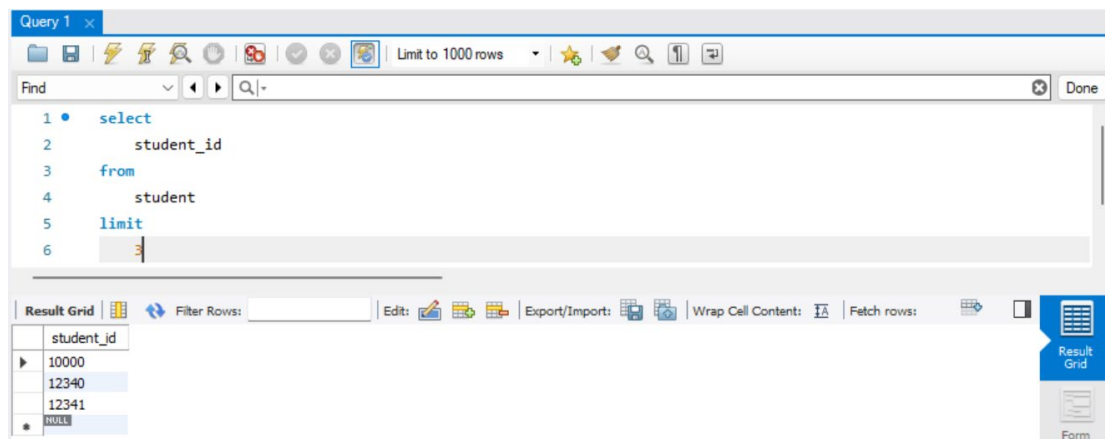
⑧ 分页查询

`select ... from ... limit x`

查找表中前 x 条数据。

`select ... from ... limit x, z`

查找表中 x+1 到 x+y-1 条数据（从第 x+1 条开始累加 y 条）。



⑨ 聚合函数

A. 计算和

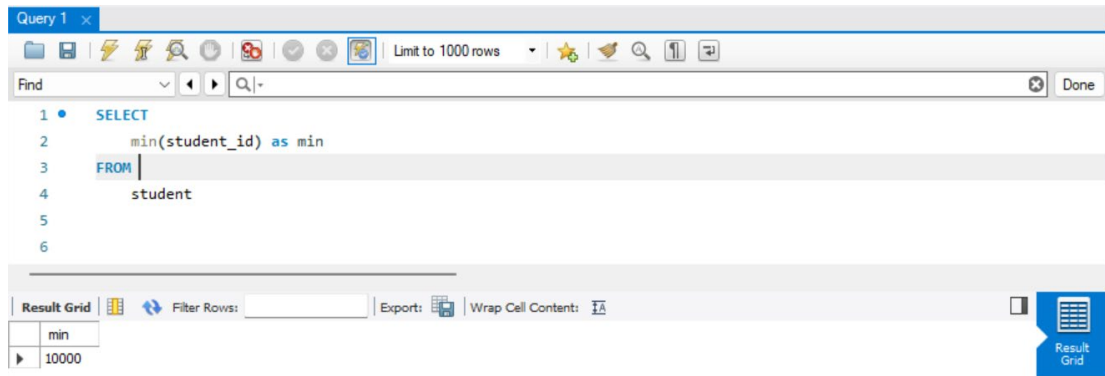
`select sum(字段) as sum from ...`

B. 计算最大值

`select max(字段) as sum from ...`

C. 计算最小值

`select min(字段) as sum from ...`

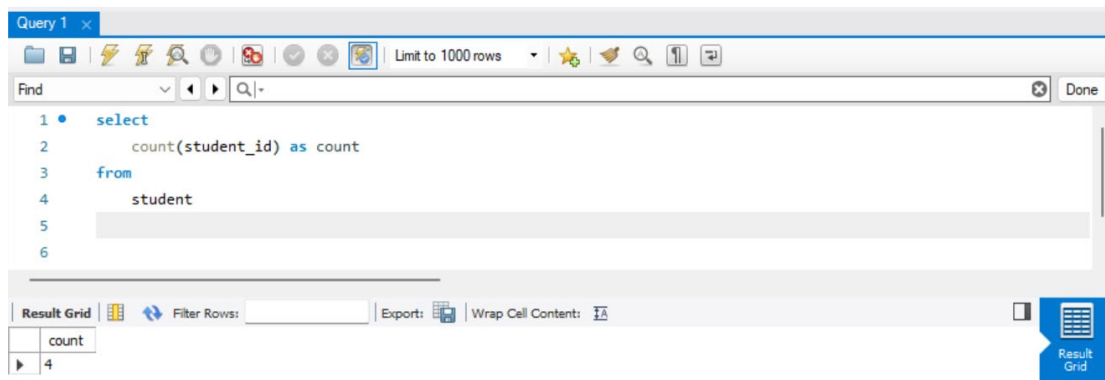


D. 计算平均值

select avg(字段) as sum from ...

E. 计算个数

select count(字段) as sum from ...

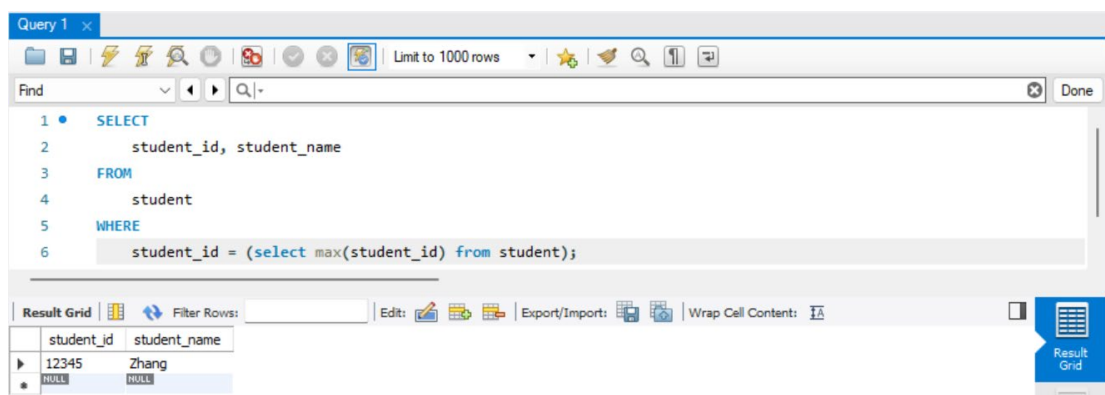


⑩ 分组查询

select ... from ... group by ...

⑪ 子查询

查询语句中嵌套查询语句。



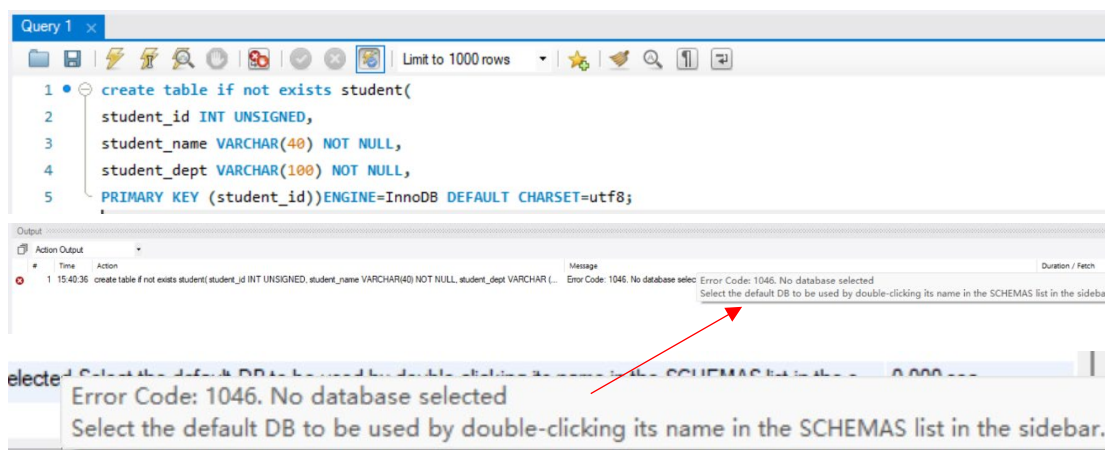
二、总结

1. 个人想法

- (1) SQL 语言大小写不敏感，用;作为一句话的结束，整体容易理解，和自然语言比较相似。
- (2) 学习 SQL 语言要多实践，才能更熟悉其使用和记住相应语法。这有利于我理解关系结构与数据字段的概念，从而掌握数据库的构建。
- (3) 表是建立关系数据库的基本结构，用来存储数据具有已定义的属性，在表的操作过程中，有查看表信息、查看表属性、修改表中的数据、删除表中的数据及修改表、删除表和查询数据等的操作。
- (4) 增删改查，这四个简单的动作构成了数据管理的基础。

2. 遇到问题

- (1) 要先选择数据库(use db_name)，才能在这个数据库中建立数据表，否则会报错。



- (2) 如果你在数据库或编码环境中使用 'utf8' 作为字符集，可能需要考虑将来的兼容性问题，系统会报 warning，

0 row(s) affected, 1 warning(s):
3719 'utf8' is currently an alias for the character set UTF8MB3, but will be an alias for UTF8MB4 in a future release. Please consider using UTF8MB4 in order to be unambiguous.

解决方法是使用 utf8mb4，支持更多的字符。

