```
- u 参数用于指定用户名。
- u 参数用于指定用户名。
- p 参数表示需要输入密码。
- p 参数表示需要输入密码。
                                      ② CREATE DATABASE 数据库名:
                                                            选择数据库: 〇 USE d
                                                            选择数据库: O USE database_name;
宣看表的结构: O DESC table_name;
                                        元数据 

    查看表的索引: 

    SHOW INDEX FROM table_name
                                                         查看表的创建进句: ○ SHOW CREATE TABLE table_name;
查看表的行数: ○ SELECT COUNT(*) FROM table_name;
                                      DROP DATABASE «database name»: ○ 直接影鈴戲鄉,不能最是百存在

IF DXSTS II—「可信息子母,表示您更見編存在才持行

图除数据库

DROP DATABASE IF DXSTS] «database name»: ○ 重接提行,提高因为截率并存在而过效回说。
                                      选择数据库 O____USE database_name;
                                      DROP TABLE table_name; ② 直接删除表,不检查是否存在
                                      删除数据表 O DROP TABLE [IF EXISTS] table_name;
                                     FROM table_name
                                                                                          查询数据 (WHERE condition) (
                                                                                        NOT条件 © SELECT * FROM products WHERE NOT category = 'Conting';
BETWEEN 集件 © SELECT * FROM orders WHERE rode_date BETWEEN 2023-01-01' AND 2023-12-31;
IS NULL 条件 © SELECT * FROM employees WHERE department IS NULL;
                                                           IS NOT NULL 条件 ② SELECT * RROM Customers WHERE email IS NOT NULL:

[ORDER BY column, name (ASC | DESCI) ② ORDER BY column, name (ASC | DESCI) 是一个可能的子句,用于彻底结果像的很常稳体,就认是升序(ASC),

[UMIT number]: ③ UMIT number 第一个可能的子句,用于影响高级的行政。
                                                                    需要修改或更新 MySQL 中的数据,可以使用 UPDATE 命令来操作
                                      UPDATE 更新 UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;
                                      | WHISE Condition | 大部隊 MySQL 鞋配表中的记录 | WHISE Condition 是一个可急的子句。用于指定影響的方,如果指電 WHISE E on Month table pame WHISE Condition 是一个可急的子句。用于指定影響的方,如果指電 WHISE E on Month table pame 基金管影像数据多数合名。
                                                             LIKE 子句中使用百分号 %字符来表示任意字符
                                                             LIKE 予例中で表現であった。

SEECT Column-1, Columning ...

FROM table pame

WHERE column name LIKE pattern

pattern 果用子高点的模式、可以核合複配符。

・ 残趣所表示一个字符。 パル 密配架 ニケテ号か Y 粉任何字符を、
・ 機能が表示一个字符。 パル 密配架 ニケテ号か Y 粉任何字符を、
・ 機能が表示一个字符。 パル 密配架 エル
                                       LIKE 子句
                                                                      UNION 操作符合 日科美国介以上的 SELECT 语句的描写自创一个结果会。 并去都重要的话。 ② 比UNION 操作符合 日科美国的人政治重要行。 河 NIONO ALL 不合法检查服务。 可从来 UNION ALL 的性能可能更好。 但以来的 ALL D的性能可能更好。 但以来的确实有重去非重要好。 可以使用 UNION, UNION 接种等必须指摘产业产 ASELECT 语句的的操作对话之面的现在分类的原则。
                                                                      SELECT column1, column2, ...
FROM table1
WHERE condition1
UNION
SELECT column1, column2, ...
FROM table2
                                       UNION 操作符
                                                                  MySQL ORDER BY(排序) 语句可以按照一个或多个列的值进行升序(ASC)或降序(DESC)排序。
                                    MySQL 语句
                                                                         GROUP BY 语句根据一个或多个列对结果集进行分组。
                                                                        在分组的列上我们可以使用 COUNT, SUM, AVG,等函数。

SELECT column1, aggregate_function(column2)

FROM table_name
                                                                                                                                 aggregate_function(column2
table_name: 要查询的表名。
                                                                                                                                                                   mn2): 对分组后的每个组执行的聚合函数。
                                                                                                                                   condition: 可选,用于筛选结果的条件。
                                     GROUP BY 语句
                                                                                                                                                                                                            mysql > SELECT coalesce(name, '总数'), SUM(signin) as signin_count FROM employee_tbl GROUP BY name WITH ROLLUP;
                                                                                                                                                                                                           使用 WITH ROLLUP ③ WITH ROLLUP 可以实现在分组统计数据基础上两进行相同的统计(SUM,AVG,COUNT...) ④ 4 rows in set (0.01 sec)
                                                                           从多个数据表中读取数据。
                                                                                                                SELECT column1、column2、_
FROM pibled*
NNMS XON (内温度送等直接): 収収両个家中字段否配及先近公記。
NNMS XON table2 ON table3 Loolumn_name = table2.column_name = table2.column_nam
                                     MySQL 连接的使用 o
                                                                                                                                                                                                                                                                                                          imn_name = table2.column_name 是连接条件,指定了两个表中用于匹配的列。
                                                                        JOIN 接開功能大致分为三类 OLEFT JOIN (左连续): 获取左表所有记录,即使有误没有对点低低的记录。 OLEFT JOIN (左连续): 获取左表所有记录,即使有误没有对点低低的记录。 OLEFT JOIN table2 ON table1.column_name = table2.column_name;
                                                                                                                SELECT column1, column2, ...
PROM table!
RIGHT JOIN(宿產費): 与 LEFT JOIN 祖辰,用于积取在表所有记录,的使左来没有对应匹配的记录。 。 Bidt Til JOIN table2 ON table1 column_name = table2.column_name
                                      NULL 值处理 S NOT NULL 当剂的值不为 NULL 运算符返目 true.

(=>: 比较提件符(不用于 = 运算符),当比效的的两个值相等或者都为 NULL 對返回 true.
                                                                 MySQL 中使用 REGEXP 和 RLIKE操作符来进行正则表达式匹配
                                                                                                 :: 匹配任意单个字符。

^: 匹配字符串的开始。

$: 匹配字符串的结束。
                                                                                                  *: 匹配零个或多个前面的元素。
+: 匹配一个或多个前面的元素。
                                                                                                  7: 匹配零个或一个前面的元素。
                                                                正则表达式匹配的字符类
                                                                                                  [abc]: 匹配字符集中的任意一个字符。
                                                                                                 [^abc]: 匹配除了学符集中的任意一个字符以外的字符。
[a-z]: 匹配范围内的任意一个小写字母。
                                     正则表达式
                                                                                                  [0-9]: 匹配一个数字字符。
                                                                                                 \w: 匹配一个学母数学学符(包括下划线)。
\s: 匹配一个学日学符。
用于检查一个学行半是否匹配施定的正则表达式模式
                                                                                                                       使用 REGEXP 进行模式匹配 🧿
                                                                                                                                                         使用 OR 进行多个匹配条件 ③ SELECT * FROM employees WHERE last_name REGEXP 'Smith|Johnson';
                                                                 使用 RLIKE 进行模式匹配 ③ RLIKE 和 REGEXP 可以互换使用,没有区别。
                                                                  MySQL 的 ALTER 命令用于修改数据库、表和索引等对象的结构。
                                                                 # ALTER TABLE TABLE_NAME MODIFY COLUMN column_name new_datatype;
                                     - WOUTH COUMN column name new datatype:

ALTER 380 CHANGE COLUMN out, column, name new column, name datatype:

B839E 0 CHANGE COLUMN out, column, name new column, name datatype:
                                                                 ALTER TABLE table_name

¡pato PRIMARY KEY ② ADD PRIMARY KEY (column_name);
                                                                 海加 FOREIGN KEY
                                                               ALTER TABLE old_table_name
修改表名 ② RENAME TO new_table_name;
                                                           临时表只在当前连接可见,当关闭连接时,MySQL 会自动删除表并释放所有空间。
                                                                              CREATE TEMPORARY TABLE temp_table_name (
column1 datatype,
column2 datatype,
                                     临时表
                                                        其余操作和普通表类似
```