1. PROMPT \u@\h \d> //更改图标显示
2. SHOW databases; //查看有多少数据库
3. CREATE database 数据库名称 //创建数据库
4. DROP database 数据库名称 //删除数据库
5. SHOW create database数据库名称 //查看数据库的信息
6. ALTER database数据库名称 DEFAULT CHARACTER SET 编码方式 //更改数据库的编码方式
7. CREATE TABLE 表名 (

字段名 类型 [约束添加]，

字段名 类型 [约束添加]，

。。。。。。

字段名 类型 [约束添加]

） //创建数据表

1. SHOW CREATE TABLE表名 //查看数据表的信息
2. DESC表名 或者：SHOW columns FROM表名 //查看数据表中的字段信息
3. DROP TABLE表名 //删除数据表
4. ALTER TABLE表名 旧表名 RENAME 新表名
5. ALTER TABLE表名 CHANGE 旧字段名 新字段名 新数据类型 //修改字段名
6. ALTER TABLE表名 MODIFY 字段名 新数据类型 //修改字段的数据类型
7. ALTER TABLE表名 ADD 字段名 数据类型 //添加字段
8. ALTER TABLE表名 DROP 字段名 //删除字段
9. ALTER TABLE表名 MODIFY 字段名1 数据类型 FIRST|AFTER 字段名2 //调整字段的顺序
10. 表的约束：即是一些维护数据库完整性的规则，从而保证数据表中数据的完整性和唯一性，常见的表的约束有以下：

|  |  |
| --- | --- |
| **约束条件** | **说明** |
| PRIMARY KEY | 主键约束，用于唯一标识对应的记录 |
| FOREIGN KEY | 外键约束 |
| NOT NULL | 非空约束 |
| UNIQUE | 唯一性约束 |
| DEFAULT | 默认约束，用于设置字段的默认值 |

1. CREATE TABLE 表名 (

id INT PRIMARY KEY AUTO\_INCREMENT, //设置主键约束，并且自动增加

name VARCHAR(20) UNIQUE, //唯一性约束

password VARCHAR(20) DEFAULT 123456, //默认约束

age TINYINT NOT NULL); //非空约束

1. 创建带索引的数据表：

CREATE TABLE 表名 (

字段名 类型 [约束添加]，

字段名 类型 [约束添加]，

。。。。。

字段名 类型 [约束添加]

[UNIQUE|FULLTEXT|SPATIAL] INDEX|KEY [别名] (字段名1 [ASC|DESC]) //创建索引

）

**PS:**中括号的是可选项，分别表示创建不同的索引。INDEX|KEY用来表示字段的索引，二选一即可。ASC表示升序，DESC表示降序。

1. 主键和索引不能在同一个字段。
2. CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX 索引名 ON 表名 (字段名 [ASC|DESC]); //在已经存在的表上创建索引。
3. ALTER TABLE 表名 ADD [UNIQUE|FULLTEXT|SPATIAL] INDEX 索引名 (字段名 [ASC|DESC]); 在已经存在的表上创建索引。
4. ALTER TABLE 表名 DROP INDEX 索引名 或者:DROP INDEX ON 索引名。 //删除索引名
5. INSERT INTO表名(字段名,字段名,…..,字段名) VALUES(字段值，字段值，……，字段值);

INSERT INTO 表名 VALUES(字段值，字段值，……，字段值);

INSERT INTO 表名 SET 字段名=字段值,字段名=字段值，……;

INSERT INTO表名(字段名,字段名,…..,字段名) VALUES

(字段值，字段值，……，字段值), (字段值，字段值，……，字段值)…..;

INSERT INTO 表名 VALUES

(字段值，字段值，……，字段值), (字段值，字段值，……，字段值)….;

1. UPDATE 表名 SET字段名=字段值,字段名=字段值，…… [WHERE 条件表达式]; //更新记录
2. DELETE FROM 表名 [WHERE 条件表达式];
3. SELECT [DISTINCT] \*|字段名 // DISTINCT是可选参数，用于剔除查询结果中重复的数据

FROM 表名

[WHERE 条件表达式]

[GROUP BY 字段名 [HAVING 条件表达式2]] //用于将查询结果按照指定的字段进行分组，“HAVING”用于分组后的过滤

[ORDER BY 字段名 [ASC|DESC]] //ORDER BY 后可跟多个字段

[LIMIT [OFFSET 记录数]] //限制查询结果的数量，“OFFSETT”表示偏移量

1. 条件查询

SELECT [DISTINCT] \*|字段名 FROM 表名 WHERE 字段名 IN|NOT (元素,元素….)

IN关键字用于判断某个字段是否在指定的集合中

SELECT [DISTINCT] \*|字段名 FROM 表名 WHERE 字段名 BETWEEN 值1 AND 值2

BETWEEN AND 关键字用于查询某个字段是否在范围内

SELECT [DISTINCT] \*|字段名 FROM 表名 WHERE 字段名 IS [NOT] NULL

通过空的字段查询

SELECT [DISTINCT] \*|字段名 FROM 表名 WHERE 字段名 LIKE ‘匹配字符串’

AND关键字和OR关键字查询，前者要满足全部条件，后者只要满足其中的一个条件。

1. MYSQL聚合函数

|  |  |
| --- | --- |
| **函数名** | **作用** |
| COUNT() | 返回某列的行数 |
| SUM() | 返回某列值的和 |
| AVG() | 返回某列值的平均值 |
| MAX() | 返回某列的最大值 |
| MIN() | 返回某类的最小值 |

1. 外键是指引用另一张表中的一列或多列，被引用的列应该具有主键约束或唯一性约束。如：学生表（student）中的gid是学生所在班级的id，是引入了班级表（grade）的主键，那么gid就作为学生表的外键。被引用的表，即班级表（grade）是主表；引用外键的表，即学生表（student）是从表，两者为主从关系。
2. Alter table 表名 add [constraint 别名] foreign key(外键字段名) REFERENCES 外表表名(主键名)
3. SELECT 查询字段 FROM 表1 [INNER] JOIN 表2 ON 表1.关系字段 = 表2.关系字段

//内连接查询

1. SELECT 查询字段 FROM 表1 LEFT|RIGHT [OUTER] JOIN 表2 ON 表1.关系字段 = 表2.关系字段 [WHERE 条件表达式] //左外链接查询

在LEFT|RIGHT [OUTER] JOIN 关键字左边的表为左表，右边为右表

LEFT JOIN：返回包括左表中的所有记录和右表符合连接条件的记录

RIGHT JOIN：返回包括右表中的所有记录和左表符合连接条件的记录

1. 子查询通常使用IN,EXISTS,ANY,ALL操作符，EXITS查询语句只返回TURE和FALSE
2. 视图是数据库的虚拟表,依赖与基本表,可像基本表一样进行增删改查等操作。具有简化查询语句，安全性，逻辑数据独立性等特点。视图也可看作一条查询语句
3. 视图的创建基于SELECT语句和已经存在的数据表,可创建多表视图也可创建单表视图

CREATE [OR REPLACE] [ALGORITHM={UNDEFINED|MERGE|TEMPTABLE}]

VIEW view\_name[(column\_list)]

AS SELECT\_statement //SELECT\_statement表示查询语句

[WITH[CASCADED|LOCAL] CHECK OPTION]

37. DESC 视图名 或 SHOW TABLE STATUS ‘视图名’\G //查看视图

38. 修改视图：把第36项例子中‘CREATE’改成‘ALERT’

39.UPDATE，DELETE，INSERT操作和基本表一样。