一、SQL常见分类：

数据定义语言（DDL:Data Difined Language）:create、alter、drop;

数据查询语句（DQL:Data Quary Language）:select ；

数据操作语言（DML:Data Manipulation Language）:insert、update、delete；

数据控制语言（DCL:Data Control Language）：grant、revoke

事务处理语言（TPL: Transaction Processing Language）:start transaction、commit、rollback；

指针控制语言（CCL）:declare cursor,fetch into 、update where current。

二、使用mysql

cmd控制台打开mysql:

     mysql -uroot -proot

1.databases

显示当前所有数据库：show databases;

使用数据库：use database\_name;

创建数据库(指定字符集):create database db\_name default character set charset\_name;

修改数据库：alter schema db\_name default character set charset\_name;

查看数据库信息:show create database db\_name;

删除数据库：drop database db\_name;

2.tables

显示当前数据库的所有表：show tables;

创建student表：

create table student(

     id int(11) not null auto\_increment,

     name varchar(32),

     sex char(2),

     primary key(id)

  )default charset=utf8;

创建student\_score表：

create table student\_score(

     studentId int(11) not null,

     course varchar(10) not null,

     score int(10),

**constraint student\_id**

**foreign key (studentId)**

**references student(id)**

)default charset=utf8;

外键约束选项用于表数据修改和删除时，各个关键表中关联数据的处理。各个参数作用如下:

CASCADE：外键表中外键字段值会跟随父表更新或所在列被删除；

NO ACTION：不进行任何关联操作；

RESTRICT:相当于NO ACTION,即不进行任何操作。拒绝父表修改外键关键列，删除记录；

SET NULL：在父表的外键关联字段被修改或删除时，外键表的外键列被设置为NULL。

查看表创建信息：show create table table\_name;

查看表结构：desc table\_name;

删除表：drop table table\_name;

查询表数据：select \* from table\_name;

添加表数据：insert into user('name','salary','age','sex','tel') values('eee','32111','32','s','2345t');

删除表数据：delete from user where id=2;

更新表数据：update table\_name set 'name'='lisi' where id = 2;

修改表名称：alter table old\_table\_name rename to new\_table\_name;

添加字段：alter table old\_table\_name add  'price' varchar(10) null after sex;

3.主键：

* 唯一地标识一行记录；
* 作为一个可以被外键引用的有效对象。

外键：

* 记录表之间的联系，保证数据的完整性和一致性；
* 能够增加数据库表关系的可读性。

4.数据查询：

select,from,where,group by,having,order by ,asc|desc

连接查询：

交叉连接查询：返回被连接的两个或多个表所有数据行的笛卡尔积。

     select \* from user,student;

内连接查询：相等连接，连接后的表中的某个列与每个表中都相同。

     select \* from table1,table2 where table1.col1 = table2.col2;

     select \* from table1 inner join table2 on table1.col1 = table2.col2; (效率更高)

外连接查询：可以查询两个或两个以上的表。

左外连接：可以查询出表1中的所有记录和表2中匹配的记录。

     select \* from table1 left join table2 on table1.col1 = table2.col2;

右外连接：可以查询出表2中的所有记录和表1中匹配的记录。

     select \* from table1 right join table2 on table1.col1 = table2.col2;

全连接：可以查询出表1和表2中所有行，滤掉重复的。

     select \* from table1 full join table2 on table1.col1 = table2.col2;

自连接：把一个表与它自身进行连接。只是每次在列出这个表时便为它命名一个别名。

子查询：将一个查询语句的结果作为一个查询语句的条件。

三、视图

* 视图**是一种特殊的表**，视图**将数据以表的形式提交给用户**，却并不存储数据。
* 视图是一个虚拟的表，其**内容由查询语句来构成，将获取的数据的SQL语句赋予视图，即可创建视图**。
* 视图引用了数据表中的字段和数据，但其本身**并不存储数据**，**只是存储获取数据的SQL语句**。

对于基础表来说**，视图的作用相当于数据筛选**。视图被引用的时候才派生出数据，不占用空间，可以对视图进行查询、修改和删除。

使用视图的优点：

* 集中数据
* 简化操作
* 实现复杂的查询需求
* 简化用户权限的管理
* 便于数据共享

     创建视图：create [or replace] view view\_name as select\_statement;

     查看视图：show create view view\_name;

     修改视图：alter view vie\_name as select\_statement;

     删除视图：drop view view\_name;

     插入视图数据：insert into view\_name() values();

     修改视图数据：update view\_name set conditions;

     删除视图数据：delete from view\_name where子句；

四、索引

     索引是一种特殊的数据结构(InnoDB数据表上的索引是表空间的一个组成部分)，它们包含着对数据表里所有记录的引用指针。更通俗的说，数据库索引好比是一本书前面的目录，能加快数据库的查询速度。

MySQL索引的类型：

* 普通索引---这是最基本的索引，它没有任何限制，比如上文中为title字段创建的索引就是一个普通索引，MyIASM中默认的BTREE类型的索引，也是我们大多数情况下用到的索引。
* 唯一索引---与普通索引类似，不同的就是：索引列的值必须唯一，但允许有空值（注意和主键不同）。如果是组合索引，则列值的组合必须唯一，创建方法和普通索引类似。
* 全文索引---FULLTEXT索引仅可用于 MyISAM 表；他们可以从CHAR、VARCHAR或TEXT列中作为CREATE TABLE语句的一部分被创建，或是随后使用ALTER TABLE 或CREATE INDEX被添加。
* 单列索引、多列索引---多个单列索引与单个多列索引的查询效果不同，因为执行查询时，MySQL只能使用一个索引，会从多个索引中选择一个限制最为严格的索引。
* 组合索引---平时用的SQL查询语句一般都有比较多的限制条件，所以为了进一步榨取MySQL的效率，就要考虑建立组合索引。