**总结**：

顺序很重要：

Select 去重（distinct） 要查询的字段 from （注意：表和字段可以取别名）

Xxx join 要连接的表 on 等值判断

Where[具体的值，子查询语句]

Group By （通过哪个字段来分组）

Having（过滤分组后的信息，条件和where是一样的，位置不同）

Order By...（通过哪个字段排序）[升序ASC/降序DESC]

Linmit strartIndex,pagesize（分页，从哪个地方开始到哪里结束）

**创建数据库**：

Create database 数据库名

**创建数据表**：

Create table if exists `数据表名`

if exists：如果不存在这个表则使用

Eg：

Create table if not exists `student`(

`id` int(4) not null[不为空]/ null[为空] auto\_increment[自增],

`name` varchar(30) not null default[设置默认值] ’匿名’,

`birthday` datetime[日期] default null,

Primary key[设置主键](`id`)

) enging=innodb default charset=utf8

**格式：**

Create table if not exists `表名`(

‘字段名’ 列类型 [属性] [索引][注释],

‘字段名’ 列类型 [属性] [索引][注释],

......

‘字段名’ 列类型 [属性] [索引][注释]

)[表类型] [字符集设置][注释]

Desc 表名——查看一张表的结构

**修改删除数据表**

——修改表名：

Alter table 旧表名1 rename as 新表名

——增加表的字段：

Alter table 表名 add 新字段名 [列属性]

——修改表的字段（重命名，修改约束！）

Alter table 表名 modify 旧字段名 [新列属性] ——modify不能修改字段名，修改约束

Alter table 表名 change 旧字段名 新字段名 [新/旧属性] ——change可以重命名也可以修改约束

——删除表的字段

Alter table 表名 drop 字段名

——删除表

Drop table if exists 表名 （加上判断if exists以免报错）

**外键**

定义外键key 给这个外键添加约束（执行引用）

Key `定义外键名` (要引用的字段),

Constraint `外键名` foreing key (`要引用的字段`) references `另一张表名`(最终引用到的字段名)

——创建表的时候没有外键关系

Alter table `表名`

Add constraint `约束名` foreing key (`作为外键的列`) references `哪个表`(哪个字段)

**DML语言**

添加Insert 修改 update 删除delete

——**插入语句（添加）**

Insert into 表名（[字段名1，字段2，字段3]）values(‘值1’，‘值2’，‘值3’，...)

Eg：

Insert into `grade`(`gradename`) values (‘大四’)

由于主键自增我们可以省略（如果不写表的字段，他就会一一匹配）

——插入多个字段

Eg：

Insert into `grade` (`gradename`)

Values (‘大一’),(‘大二’)

——**修改updata**

Update `表名` set `字段名`=’修改的内容’ where 约束条件

Eg：

Update `student` set `name` = ‘...’ where id=1;

语法：

Update 表名 set colnum\_name = value where [条件]

修改多个属性，逗号隔开

Update `表名` set `字段名`=’修改的内容’, `字段名`=’修改的内容’ where 约束条件

操作符：>, <, <>(!=), between ... and ... (在某个范围内), and,or

——删除delete

语法：delete from 表名[where 条件]——删除指定数据

Truncate命令——完全清空一个数据库表，结构和索引不会变

Truncate table 表名

DQL查询

——查询某个表

Select \* from 表名

——查询指定字段

Select `字段` from 表名

——给结果起别名

Select `字段1` as 别名, `字段2` as 别名 from 表 as 表的别名

——字段拼接

Select concat(‘字符串’，字段名) as 别名 from 表

——去重distinct（去除select出来的重复数据）

Select distinct `字段名` from 表名

**Where条件**

运算符：

And （&&），or（||），not（！）

区间也可用between and

**模糊查询**

is null （a is null）=true

Is not null （a is not null）=true

Between （a between b and c）若a在b和c之间true

Like （a like b）如果a匹配b=true

In （a in [a1,a2,a3...]）若a在a1或a2其中的某一个值=true

Like结合%（代表0到任意个字符）\_（一个字符）

Eg：

查询姓刘的同学

Select xxx from student

Where name like ‘刘%’；

查询姓刘的两个名字的同学

Select xxx from student

Where name like ‘刘\_’；

——In操作（具体的值，不能用%和\_）

查询指定学生1001、1002、1003

Select xxx from student

Where id in （1001，1002，1003）；

**联表查询（join on）**

查询参加了考试的同学（id name...）

——Inner join（如果表中至少有一个匹配，就返回行）

Select s.id, name, subject, grade

From student as s

Inner join result as r (因为两个数据表中都有id所以定义别名确定从哪个表中取id)

On s.id = r.id

【s.id】表示从哪个表中提取id

——Right join（会从左表中返回所有的值，即使右表中没有匹配）

Select s.id, name, subject, grade

From student as s

Right join result as r

On s.id = r.id

——Left join

Select s.id, name, subject, grade

From student as s

Left join result as r

On s.id = r.id