### MySql数据库基础

1. 数据库的操作：

1.1.显示当前已有的数据库：show databases;

1.2创建数据库：create database if not exists 数据库名;

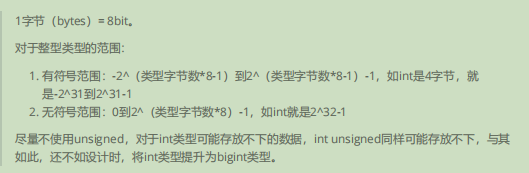
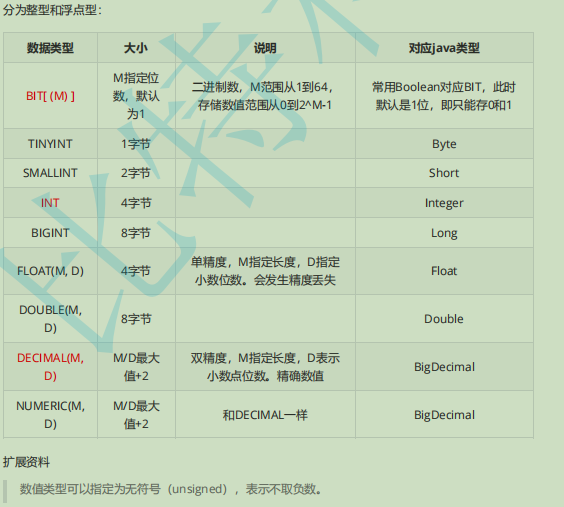
当我们创建数据库没有指定字符集和校验规则时，系统使用默认字符集，utf8,校验规则是，utf8\_general\_ci.MySQL的utf8编码不是真正的utf8，没有包含某些复杂的中文字符。MySQL真正的utf8是使用utf8mb4，建议大家都使用utf8mb4.

1.3使用数据库：use 数据库名;

1.4删除数据库：drop database if exists 数据库名;(表和数据都被删除)

1. 常用数据类型：

2.1数值类型：



2.2字符串类型：

2.3日期类型：

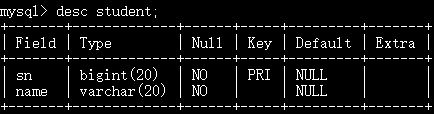


**注意：**如果使用两个timestamp，那么第二个timestamp要设定默认值，不然不能使用。

1. 表的操作：需要对数据库中的表进行操作时，首先得先连接数据库

3.1查看表结构：desc 表名;

索引类型



字段名 字段类型 是否可以 默认值 扩充

为空

3.2创建表，可以使用comment增加字段说明：

Create table student1(

Id int comment “学号”,

name varchar(20) comment “姓名”,

age int,

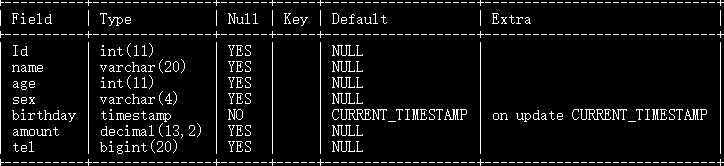
sex varchar(4),

birthday timestamp,

amount decimal(13,2),

tel bigint comment”电话”

);



3.3删除表：drop table if exists 表名;



### MySQL表的增删改查

1. CRUD：

C:create,新增数据

R:retrieve,查询数据

U:update,修改数据

D:delete,删除数据

1. 插入数据：

2.1单行数据+全列插入：

Insert into 表名 values (字段1,字段2,字段3···);



注意：1979-01-01 10:10:10不要加空格

2.2多行数据+指定列插入：

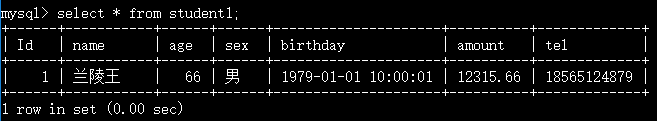
Insert into 表名 (字段1,字段2···) values

(字段1值,字段2值···),

(字段1值,字段2值···);

1. 查询：

3.1全列查询：select \* from 表名;



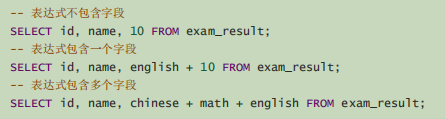
**注意：**通常情况下不建议使用全列查询，因为：

1. 查询的列越多，意味着传出的信息量越大
2. 可能会影响到索引的使用

3.2指定列查询：select 字段1,字段2··· from 表名;

**注意：**指定列查询的要查询的字段顺序可不必按照定义的顺序

3.3查询字段为表达式：

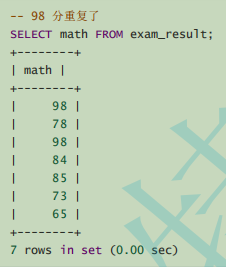
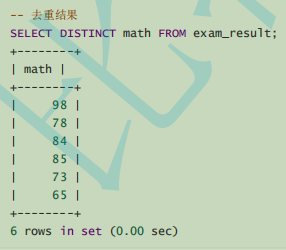


3.4别名：表示查询结果中列的别名，表示返回的结果集中，以别名作为该列的名称



3.5去重：

关键字：distinct，对某列数据进行去重

3.6排序：order by

升序排序（ASC），默认为ASC

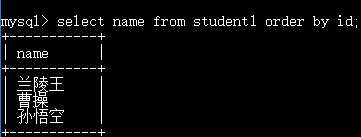
3.6.1 排序

降序排序（DESC）

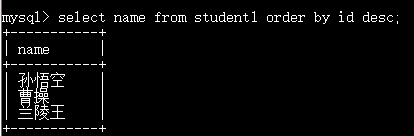
3.6.2没有order by子句的查询，返回的顺序是未定义的，不要依赖这样的查询！

3.6.3 字段值为null的比任何数据都小

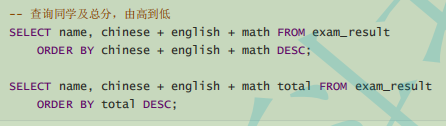
通过id升序查询：



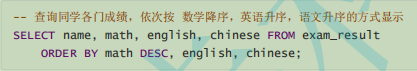
通过id降序查询：



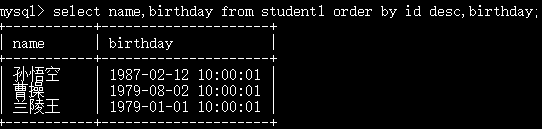
3.6.4使用表达式及别名排序



3.6.5可以对多个字段进行排序，排序优先级随书写顺序



例如：

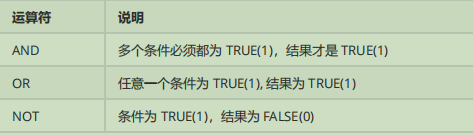


3.7 条件查询

3.7.1比较运算符：



3.7.2逻辑运算符：

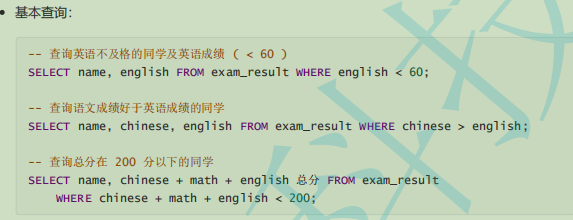


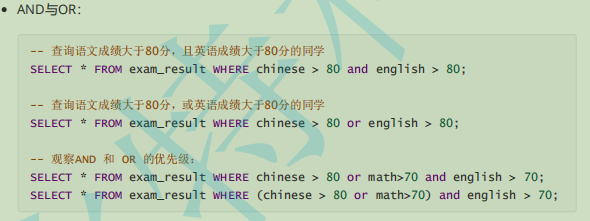
**注意：**

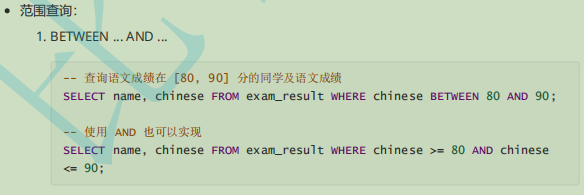
**1.**where条件查询可以使用表达式，但不能使用别名

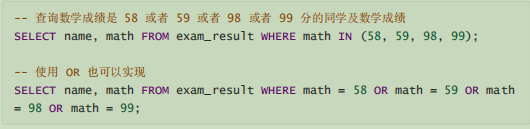
**2.**and优先级高于or，并且在使用的同时，需要用()包裹优先执行的部分

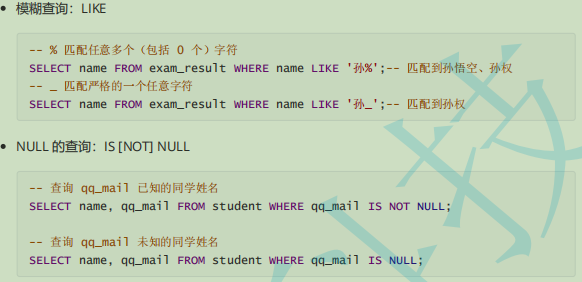
3.7.3查询案例：





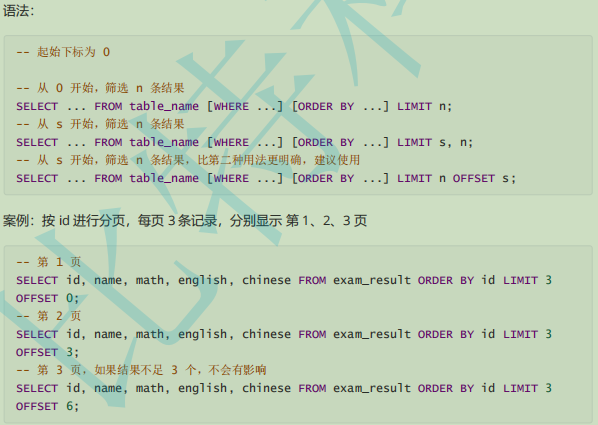




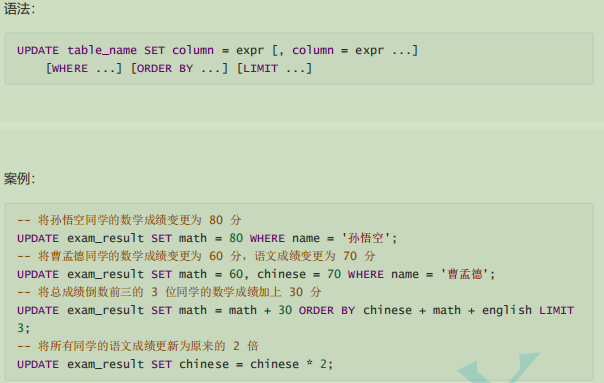


3.8分页查询：表中有多个数据，要将数据分成几部分查询，比如9个数据，分成3-3-3：1-3，

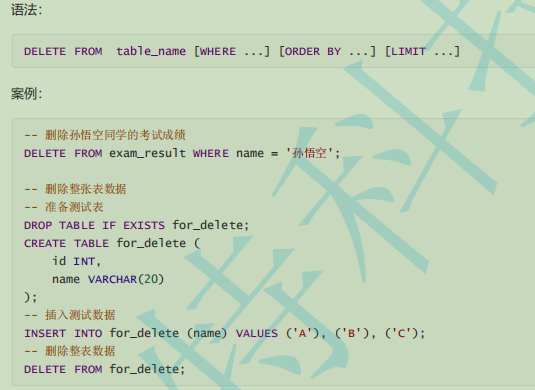
4-6,7-9进行查询，就可以使用分页查询，结果不满足3个，不影响



4.修改



1. 删除



**注意：drop是删除整张表，delete是删除表中的所有数据**

### MySQL表的增删改查进阶

1. 数据库约束

1.1约束类型：

·**not null**：指示某列不能存储null值

·**unique**：保证某列的每行有唯一的值

·**default**：规定没有给列赋值时的默认值

·**primary key:**主码，not null和unique的结合。确保某列有唯一的标识

·**foreign key:**外码，确保一个表中的数据匹配另一个表中的值的参照完整性

·**check:**保证列中的值符合指定的条件

1.2null约束：

Create table if not exists student(

Id int not null,

Sn int,

name varchar(20),

Qq\_mail varchar(20)

);

1.3unique 唯一约束：

Create table student(

Id int not null,

Sn int unique,

Name varchar(20),

Qq varchar(20)

);

1.4default 默认值约束：

Create table student(

Id int not null,

Sn int unique,

Name varchar(20) default “unkown”,

Qq varchar(20)

);

1.5primary key:主码约束

Create table student(

Id int not null primary key,//对于整数类型主码，常搭配自增长auto\_increment来使用,插入数据对应字段不给值时，使用最大值+1

Sn int unique,

Name varchar(20) default “unkown”,

Qq varchar(20)

);

1.6 foreign key:外码约束，外码用来关联其他表的主键或唯一键

Foreign key(字段名) references 主表（列）

举例：

//创建班级表,id为主码,使用MySql关键字做字段时要采用’ ’标识

Create table classes(

Id int primary key auto\_increment,

Name varchar(20)

);

//创建学生表，一个学生对应一个班级，一个班级对应多个学生。使用id作为主码，classes\_id做外码，关联班级表id

Create table student(

id INT PRIMARY KEY auto\_increment,

sn INT UNIQUE,

name VARCHAR(20) DEFAULT “unkown”,

qq\_mail VARCHAR(20),

classes\_id int,

FOREIGN KEY (classes\_id) REFERENCES classes(id)

);

1.7check约束

create table test\_user (

id int,

name varchar(20),

sex varchar(1),

check (sex ='男' or sex='女')

);

1. 表的设计：

三大范式：

2.1

一对一：

人

1 1

身份证

2.2

一对多

拥有

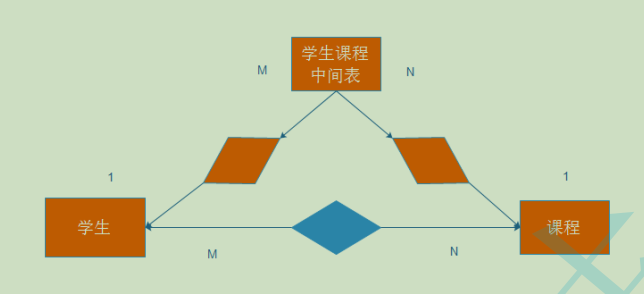
学生

班级

1 N

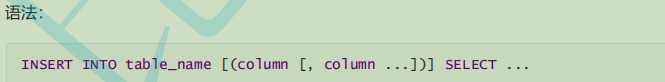
2.3

多对多



1. 新增

3.1插入查询结果：



举例：创建一张用户表，设计有name，email，sex，mobile，需要把已有的学生数据复制进来，可复制字段为name，qq\_email

Create table test\_user(

Id int auto\_increment,

Name varchar(20) comment “姓名”,

Age int comment “年龄”,

Qq\_email varchar(20) comment”电子邮箱”

);

//插入查询结果

Insert into test\_user (id,name,qq\_email) select id,name,qq\_email from student;

1. 查询

4.1聚合查询

4.1.1聚合函数

