**2操作数据库**

1.创建数据库

中括号为可选项

Create Database [ IF NO EXISTS ] dbName

2.删除数据库

DROP Database [ IF EXISTS ] dbName

3.使用数据库

Use ‘dbName’

4．查看数据库

Show DATABASES

**2.2数据库的列类型**

数值

Tinyint 十分小的数据 1个字节

Smallint 较小的数据 2个字节

Mediumint 中等大小的数据 3个字节

Int 标准的整数 4个字节

Bigint 较大的数据 8个字节

Float 浮点数 4个字节

Double 浮点数 8个字节（精度问题）

Decimal 字符串形式的浮点数 金融计算一般用此

字符串

Char 字符串固定大小的 0~255

Varchar 可变字符串 0~65535

Tinytext 微型文本 2^8 -1

Text 文本串 2^16 -1

日期类型

Java.util.Date

Date YYYY-MM-DD 日期格式

Time HH:mm:ss 时间格式

Datetime YYYY-MM-DD HH:mm:ss 最常用的时间格式

Timestamp 时间戳，1970.1.1到现在的毫秒数！也是常用

Year 年份表示

null

没有值，未知

注意，不要使用NULL进行运算，结果为NULL

**2.3数据库的字段属性**

Unsigned:

无符号的整数

声明了该列不能为复数

Zerofill

O填充

不足的位数用0来填充 int(3) 5--005

自增

通常理解为自增，自动在上一条记录的基础上+1（默认）

通常用来设计唯一的逐渐~index，必须是整数类型

可以自定义设计主键的增值起始值和步增长

非空 NULL not Null

假设设置为not null，如果不给他赋值，就会报错

Null，如果不填写值，默认就是null!

默认

设置默认

Sex,默认值为男，若果不指定该列的值，则会有默认值！

拓展

每一个表，都必须存在以下五个字段！

Id 主键

`version` 乐观锁

Is\_delete 伪删除

Gmt\_create 创建时间

Gmt\_update 修改时间

**2.4创建数据库表**

Create table [ if not exists ]‘student’(

id int(4) [ not null ] comment '学号',

name varchar(30) [ not null ]comment '姓名',

primary key(id)

)

**2.5数据库表的类型**

MYISAM INNODB

事务支持 NO YES

数据行锁定 NO YES

外键约束 NO YES

全文索引 YES NO

表空间大小 较小 较大约为2倍

常规使用操作

MYISAM 节约空间，速度较快

INNODB 安全性高，事务的处理，多表多用户操作

在物理空间存在的位置

在数据库data文件夹下

InnoDB 在数据库表中只有一个\*frm文件，以及上级目录下的ibdata1文件

MYISAM 对应文件

\*frm 表结构的定义文件

\*MYD 数据文件(data)

\*.MYI 索引文件(index)

设置数据库表的字符集编码

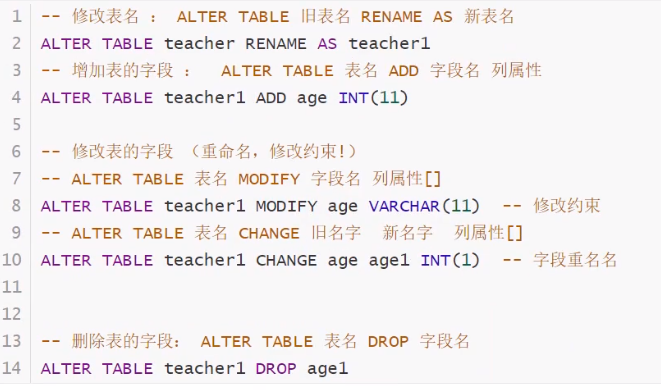
Charset = utf8

不设置mysql默认编码Latin1,不支持中文

在my.ini中配置默认的编码

Character-set-server = utf8

2.6修改删除表



删除

Drop Table IF EXISTS teacher



**3 MYSQL数据库管理**

3,1外键(了解)

1. 创建表的时候



删除的时候，必须先删除引用我的表，再删除我

1. 创建表的时候没有外键关系



以上的操作都是物理外键，数据库级别的外键，不建议使用！（避免数据库过多造成困扰）

最佳实践

数据库就是单纯的表，只用来存行（数据）和列（字段）

想舒勇多张表的数据，想使用外键用程序实现

每次做DELETE或者UPDATE操作的时候必须考虑外键约束很痛苦

3.2 DML语言(重要)

数据库一样：数据存储，数据管理

DML数据操作语言

Insert

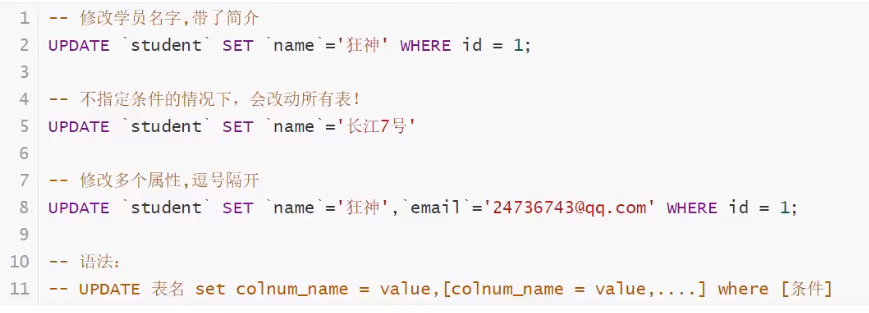
Update

Delete

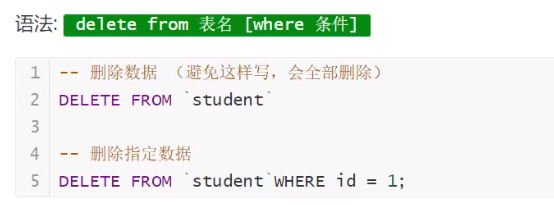
3.3添加

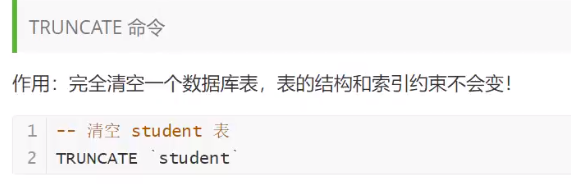


3.4修改



3.5删除





Delete 和 TRUNCATE 的区别

相同点：都能删除数据，都不会删除表结构

不同：

TRUNCATE 重新设置 自增列 计数器会归零

不会影响事务



**4 DQL 查询数据(最重点)**

4.1 DQL

（Data Query Language：数据库查询语言）

所有的查询操作都用它 Select

简单的查询。复杂的查询它都能做

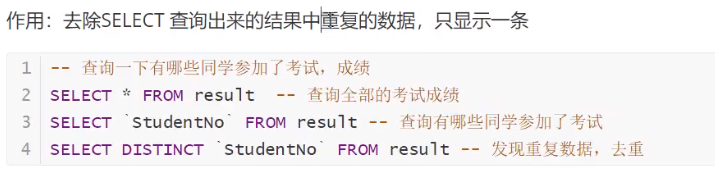
数据库中核心的语言，最重要的语句

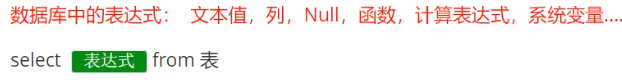
使用频率最高的语句

4.2指定查询字段



去重 distinct





4.3 Where 条件句子

作用：检索数据中符合条件的值

模糊查询：比较运算符



**4.4联表查询**



Join on 连接查询

Left

Right

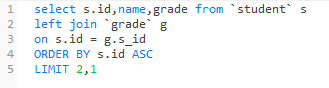
Inner

Where 等值查询

自连接

自己对自己链接，比如分级目录





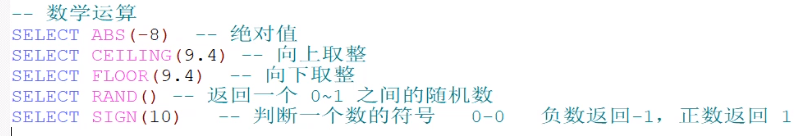
**4.6子查询**

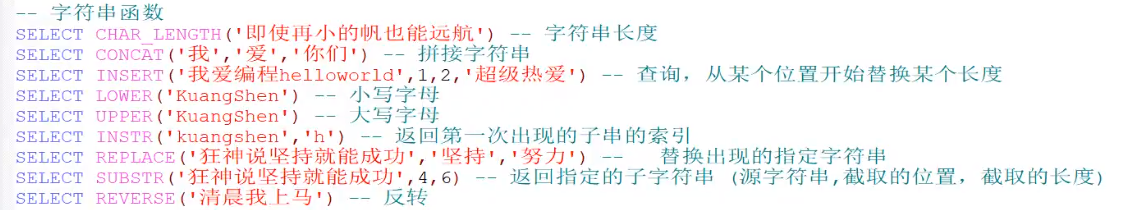
Where

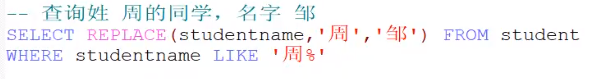
本质 在where 语句中嵌套一个子查询语句

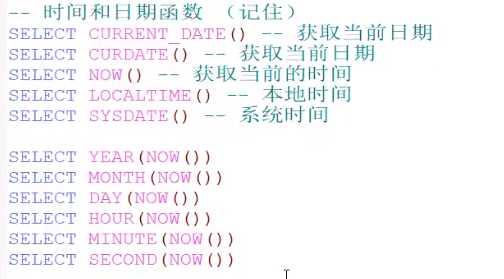
**5 MYSQL函数**

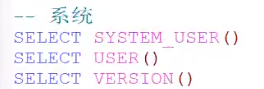
5,1 常用函数









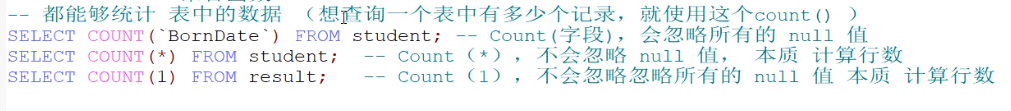


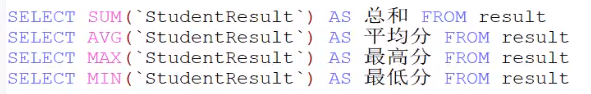
**4.7分组和过滤**



**5.2聚合函数（常用）**





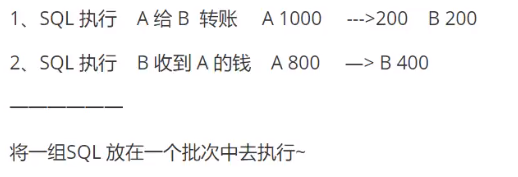


**5.3** **数据库级别的 MD5加密**

**6 事务**

6.1什么是事务

要么成功，要么失败



原子性

要么成功，要么失败

一致性

书屋前后的数据完整性要保证一致

持久性

事务一旦提交不可逆

事务没有提交，恢复原状

事务已经提交，持久化到数据库

隔离性

针对多个用户操作时，主要排除其他事务对本次事务的影响

事务的隔离级别

脏读

指一个事务读取了另外一个事务未提交的数据

不可重复读

在一个事务读取表中的某一行数据，多次读取结果不同。（这个不一定是错误，只是某些场合不对）

虚读（幻读）

是指在一个事务内读取到了别的事务插入的数据，导致前后读取的不一致（一般是行影响，多了一行）

**7 索引**

7.1 索引的分类

主键索引 ( primary key )

唯一的标识，主键不可重复，只能有一个列作为主键

唯一索引 ( unique key )

避免重复的列出现，唯一索引可以重复，多个列都可以表示为 唯一索引

常规索引 ( key/index )

默认的 index，key关键字来设置

全文索引 ( full text )

在特定的数据库引擎下才有，MyISAM

快速定位数据

7.2测试索引

**7.3索引原则**

索引不是越多越好

不要对经常变动数据加索引

小数据量的表不需要加索引

索引一般加载常用来查询的字段上

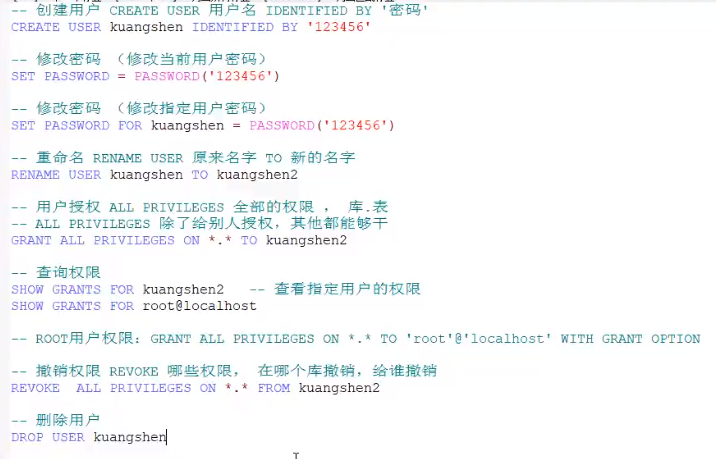
索引的数据结构

Hash类型的索引

Btree：InnoDB默认数据结构

**8．权限管理和备份**

8．1用户管理



8.2 Mysql备份

为什么要备份

保证数据的不丢失

数据转移

Mysql数据库备份的方式

直接拷贝物理文件

在Sqlyog这种可视化工具中手动导出

使用命令行导出 mysqldump命令行使用



**9 规范数据库设计**

9．1为什么需要设计

当数据库比较复杂的时候，我们就需要设计了

**糟糕的数据库设计：**

数据冗余，浪费空间

数据库插入和删除都会麻烦、异常【屏蔽使用物理外键】

程序性能差

**良好的数据库设计**

节省内存空间

保证数据库的完整性

方便我们开发系统

**软件开发中，关于数据库的设计**

分析需求：分析业务和需要处理的数据库的需求

概要设计：设计关系图E-R图

**设计数据库的步骤：（个人博客）**

收集信息，分析需求

用户表（用户登录注销，用户的个人信息，写博客，创建分类）

分类表（文章分类，谁创建的）

文章表（文章的信息）

友链表（友情链接）

自定义表（系统信息，某个关键的字）

标识实体（表需求落地到每个字段）

标识实体之间的关系

写博客：User🡪blog

创建分类：user->category

关注：user->user

友情链接：links

评论：user-user-blog

**9.2三大范式**

为什么需要数据规范化？

信息重复

更新异常

插入异常

无法正常显示信息

删除异常

丢失有效的信息

三大范式

**第一范式（1NF）**

原子性：保证每一列不可再分

**第二范式（2NF）**

前提：满足第一范式

每张表描述一件事情

**第三范式（3NF）**

前提：满足第一，二范式

需要确保数据表中的每一列数据都和逐渐直接相关，而不能间接相关

**规范性和性能的问题**

关联查询的表不得超过三张表

考虑商业化的需求和目标，（成本，用户体验！）数据库的性能更加重要

在规范性能的问题的时候，需要适当考虑一下规范性！

故意给某些表增加一些冗余的字段。（从多表查询变为单表查询）

故意增加一些计算列（从大数据量减低为小数据量的查询：索引）

**10.8 事务**

要么都成功，要么都失败

ACID原则

原子性：要么全部完成，要么都不完成

一致性：总数不变

隔离性：**多个进程互不干扰**

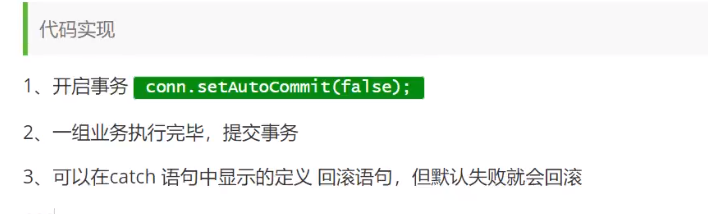
持久性：一旦提交不可逆

隔离性问题：

脏读：读取了另一个没有提交的事务

不可重复读：在同一个事务内，重复读取表中的数据，表内数据发生了变化

虚读（幻读）：在一个事务内，读到了别插入的数据，导致前后读出来的结果不一致



**10.9 数据库连接池**

数据库连接池 --- 执行完毕 ---释放

连接 – 释放 十分浪费系统资源

**池化技术：直播一些预先的资源，过来就连接预先准备好的**

---- 开门 – 业务员：等待 – 服务 ---

常用连接数：10

最大连接数：15

等待超时：100ms

编写连接池：实现一个接口 DataSource

开源数据源实现（拿来即用）

DBCP

C3P0

Druid：阿里巴巴

使用了这下数据库连接池之后，我们在项目开发中就不需要编写连接数据库的代码了