目录

[max min sum avg count 函数 2](#_Toc24474)

[Select 查询的五种子句 2](#_Toc15123)

[group by 关键字 2](#_Toc7164)

[as 关键字 3](#_Toc23099)

[having 关键字 3](#_Toc3007)

[Limit 关键字 3](#_Toc17628)

[子查询 4](#_Toc14647)

[Union 5](#_Toc8962)

[视图：创建视图 5](#_Toc24376)

[表管理之列的增删改 6](#_Toc4930)

[Mysql 触发器 trigger 6](#_Toc31951)

[mysql 存储引擎 7](#_Toc12310)

[事务的语法 8](#_Toc939)

[数据库备份与恢复 8](#_Toc18725)

增删查改

Delete from 表名 (删除 表中所有数据)

truncate 表名 (删除 表中所有数据)

insert into 表名(stu\_name,age) values ('小明','20'),('小花','22') (插入多条数据)

desc 表名 ------列出该表的所有字段

Sql 语句的关键字先后顺序

Where group by order by having limit

一个sql语句可以出现多个关键字 但是先后顺序不能错

Sql语句的不等于符号 <>

## max min sum avg count 函数

select max（字段名称） from 表格名称 选出该字段的最大值

select min（字段名称） from 表格名称 选出该字段的最小值

select sum（字段名称） from 表格名称 返回该字段 值的总和

select avg（字段名称） from 表格名称 返回该字段 的平均值

select count（\*） from 表格名称 返回此表格的总结记录数

## Select 查询的五种子句

1.Group by 2 .having 3.order by 4.limit 5 where

group by 关键字

和上面的5个函数配合使用才有意义

// group\_id ，price 都是字段名

查询每个组的最大值 以及组号 {也可以与 min sum avg count函数搭配}

select group\_id,max(price) from 表格名称 group by group\_id；

分组查询 group\_id 组号

分组关键字 要小心 没用5个函数修饰的字段 默认取的该小组内第一个该字段的值!

## as 关键字

可以给列名或者表名 取个别名

## having 关键字

**where 关键字只能对表中存在的列进行筛选（别名就不被where识别）**

**查询结果也是一个表的结构（可以有别名 别名可以看作一个列）**

**用having 关键字 可以对查询结果进一步筛选**

**例子: name price 列名 pingjun 别名**

**select name,avg(price) as pingjun from 表格名 group by name having pingjun >50;**

**排序功能:Order by 字段名 默认升序排列 手动声明(asc)**

**降序排列 : Order by 字段名 desc**

**Order by 可以按多个字段进行排序**

**Order by 字段名1（desc||asc），字段名2（desc||asc） 优先按字段1排列，字段1 一样，再按字段2排列**

## Limit 关键字

放到 语句最后面

做分页的时候 有三个参数：

总记录数 每页记录数 当前页（当前页\*每页记录数 =跳过的记录数）

**limit n,m n:跳过的记录数 m:一页由多少条记录（其中n是可选属性）**

**n省略的话 limit m; 相当于 limit 0,m;**

**例子：select \* from 表名 limit 3,3; 相当于第二页记录**

## 子查询

**Where 型子查询：将内层查询结果(可以是一个结果，或多个结果)当作外层查询的条件**

例子： 查询每个小组中数学成绩最高的学生的姓名，math成绩，小组号

select name,math,groups from student where

math in (select max(math) from student group by groups);

**from 型子查询**

将内层查询结果当作一个临时表 必须给临时表取个别名

例子：select tem.name,max(math) from(select \* from student order by math desc) as tem group by groups;

**Exists 子查询**

exists(sql语句 找到记录) 返回真

## Union

要求：两次查询的列数一致就行

将两个表的查询结果合并成一张表

（sql01）union（sql02）

**左外链接**：先把左边的表全部列出来 再将右边的表与左边表id相同的 拼接在左表右边

select \* from student **left join** qp\_city **on** student.id=qp\_city.cityId;

**右外链接：**先把**右边**的表全部列出来 再将**左边**的表与右边表id相同的 拼接在右表的左边

select \* from student **right join** qp\_city **on** student.id=qp\_city.cityId;

**内连接：**内链接只把符合两个id相同的记录筛选出来

select \*from student **inner** join qp\_city **on** student.id=qp\_city.cityId;

## 视图：创建视图

视图和 表的用法基本一致 视图是虚拟的

create view 视图名as select math,yuwen from student;

Mysql 字符集与校对集

1 设置服务器默认字符集 2 数据库默认字符集 3 表默认字符集 4 列默认字符集

如果某一个级别没有指定字符集，则继承上一级

## 表管理之列的增删改

**alter table 表名 add 字段1 varchar(110) after 字段2;**

**向字段2后面添加字段1**

**将字段加到表的第一列**

**alter table 表名add 字段名varchar(30) first (not null default 值);**

**alter table 表名 add 字段1 varchar(110) 向表添加一列**

**alter table student change math math int ; 修改math列为int型**

**删除列：alter table 表名 drop 列名**

Delimiter $ 告诉mysql服务器 sql语句的结束符号 是 $ 不再是；

## Mysql 触发器 trigger

特点：触发事件的操作和触发器里的SQL语句是一个事务操作，具有原子性,要么全部执行，要么都不执行；

创建触发器的四个要素：

监视地点(即table)

监视事件(增删改)

触发时间(before after)

触发事件 begin 和end 之间的sql语句

创建触发器的语法

执行增删改语句 触发 sql语句(一条或者多条sql语句 仅限增删改)

**Create trigger**  触发器名称  **after/before insert/update/delete on** 表名[件事地点]

**For each row**

**Begin**

sql语句(一条或者多条sql语句 仅限增删改) ;

**End;**

Create trigger 触发器名称 after/before insert/update/delete on 表名b1

For each row

Begin

Sql 语句 insert 语句 用new 引用 b1表的新纪录 delete语句 用 old引用 b1表删除的记录 如: old.列名

Update 语句 修改之前用old 引用 修改之后 用 new来引用

End;

After 和 before 的区别 after再执行 触发器中sql语句时可以对 增删该语句进行 限制

drop trigger 触发器名称 删除一个触发器

show triggers 查看所有的触发器

## mysql 存储引擎

事务的存储引擎 innodb ：此引擎支持事务

bdb 引擎也支持事务 此引擎使用的不多

事物的特性：

原子性：一组操作要么同时成功，要么同时失败

隔离性：在所有操作没有完成之前，其他会话不能看见中间改变的过程

某个事务的操作对其他事务是不可见的

一致性：事务发生前和发生后，数据总数匹配

持久性（时效性）：事务产生的影响不能够撤销

## 事务的语法

开启事务:start transaction

提交事务 Commit 不提交事务，数据不会同步到数据库

回滚：rollback

当 Commit 或者rollback 一个事务就结束了 下次需要重新开启

有些语句会造成上个事务隐式的提交 如：start transaction

## 数据库备份与恢复

/\*\*导出的表是建表语句和插入语句\*/

在c盘的各种软件目录下 导出hibernate 库下的user表文件

mysqldump -uroot -p159786 库名 表名**>** c:\\各种软件\\user.sql

mysqldump -uroot -p159786 hibernate 表1，表2...**>** c:\\各种软件\\XX.sql

//导出改库下所有的表 没有创建库的语句

mysqldump -uroot -p159786 库名**>**地址\\备份文件名

//以库为单位导出库文件 关键字 **-B**

mysqldump -uroot -p159786 **-B** 库名1 库名2 **>**地址\\备份文件名

//如何导出所有的库

mysqldump -uroot -p密码 -A>地址\\备份名称

**恢复备份（登陆mysql命令行情况下）**

**用sql命令将sql文件导入数据库(执行的是建表语句 和 插入语句)**

**以库为单位** source c:\\XXX.sql; 恢复库以及该库下的表

恢复表 分2步 1：use 库名 2：source c:/sorftwares/user.sql;

恢复备份（登陆mysql命令行情况下）

恢复库级：mysql -uroot -p159786 < c:/.....xxx.sql

回复表： mysql -uroot -p159786 库名 < c:/.....xxx.sql