

MySQL第二天-拓薪教育

聚集函数

- 1.count 记数
- 2.sum 求和
- 3.avg
 - * 语法: select avg(列名) from 表名;
- 4.max
 - * 求最大值
- 5.min
 - * 求最小值

练习

求一个班级数学平均分?

```
select avg(ifnull(math,0)) from stu;
```

求一个班级总分平均分

```
select avg(ifnull(math,0)+english+chinese) from stu;
```

求班级英语最高分和最低分

```
select max(english) from stu;
```

```
select min(english) from stu;
```

分组 (重要)

- 1.使用group by 字段 进行分组的。

```
create database day16;
use day16;
create table orders(
    id int,
    product varchar(20),
    price float
);
```

```
insert into orders(id,product,price) values(1,'电视',900);
insert into orders(id,product,price) values(2,'洗衣机',100);
insert into orders(id,product,price) values(3,'洗衣粉',90);
insert into orders(id,product,price) values(4,'电视',900);
insert into orders(id,product,price) values(5,'洗衣粉',90);
insert into orders(id,product,price) values(6,'洗衣粉',90);
```

练习

对订单表中商品归类后,显示每一类商品的总价

```
select product,sum(price),count(*) from orders group by product; 默认是一组
```

查询购买了几类商品,并且每类总价大于100的商品

```
select product from orders group by product having sum(price) > 100;
```

总结: Having和where均可实现过滤,但在having可以使用聚集函数,where不能使用聚集函数, having通常跟在group

by后，它作用于分组

总结

```
select ... from ... where ... group by ... having ... order by ...
```

* 固定的顺序：如果没有上述的条件，把关键字去掉就ok。

修改密码

1. 停止mysql服务：

services.msc 进入到服务界面

2. 在cmd>输入一个命令：

mysqld --skip-grant-tables （开启一个mysql服务，不需要进行认证。）

3. 新打开一个cmd窗口

mysql -u root -p 不需要输入密码. 就可以进入.

4. 输入命令 show databases; 查看数据库，输入命令 use mysql; 使用mysql数据库。

5. 修改密码的语句：

```
update user set password=password('root') WHERE user='root';
```

6. 将两个窗口都关闭。

7. 任务管理器中结束 (mysqld) 进程。

8. 重启mysql服务

数据库的备份和恢复

MySQL数据库备份和恢复

1. 备份数据库表中的数据

命令：mysqldump -u 用户名 -p 数据库名 > 文件名.sql 回车后 再输入密码

2. 恢复数据库（前提创建空的数据库，并且use）

命令：mysql -u 用户名 p 数据库名 < 文件名.sql 回车后 再输入密码

* 注意1：不是在数据库的登陆状态下

* 注意2：该命令后没有分号结束

* 注意3：注意 > 符号的方向

* 注意4：恢复数据库使用的命令是mysql，而不是mysqldump

单表的约束

1. 代表记录的唯一标识。

2. 声明某一列作为主键

* 使用关键字 primary key

3. 主键值的特点

* 唯一

* 非空

* 被引用

4. 创建一张表，声明主键列

1 第一种

```
create table person(  
    id int primary key,  
    name varchar(30)
```

```
);
```

```
insert into person values (1,'聪聪');
insert into person values (2,'美美');
insert into person values (3,'小凤');
```

2 第二种

```
create table person(
    id int,
    name varchar(30),
    primary key (id)
);
```

5. 自增长

- * 可以去帮你维护主键的信息
- * 关键字: auto_increment

```
create table person(
    id int primary key auto_increment,
    name varchar(30)
);
```

```
insert into person values (null,'聪聪');
insert into person values (null,'美美');
insert into person values (null,'小凤');
```

```
delete from person where id = 1;
```

唯一和非空

- | | | | |
|-------|------------|-------|----------|
| 1. 唯一 | : 声明值是唯一的 | 使用关键字 | unique |
| 2. 非空 | : 声明值是不为空的 | | not null |

多表外键的约束

```
create table dept(
    did int primary key auto_increment,
    dname varchar(30)
);

create table emp(
    eid int primary key auto_increment,
    ename varchar(30),
    sal double,
    dno int
);

insert into dept values (1,'研发部');
insert into dept values (2,'人事部');

insert into emp values (null,'聪聪',15000,1);
insert into emp values (null,'邦邦',5000,1);
insert into emp values (null,'美美',6000,2);

insert into emp values (null,'小凤',8000,2);
```

```
insert into emp values (null,'如花',8000,null);
```

问题：

- * 直接把研发部删除掉。delete from dept where did = 1; 能删除成功。
- * 在现实生活中，不合理。如果你避免问题的发生，两个表之间设置关系。

1.引入外键的约束

2.添加外键

- * 正常的情况下（一个部门有多个员工，一个员工只能属于一个部门）
- * 设置员工标签的dno字段，作为外键，指向部门表的主键。
- * 修改员工的表，在员工添加外键。

```
alter table emp add foreign key emp (dno) references dept (did);
```

3.直接添加外键

- * 在创建表的时候，指定外键

```
create table emp(  
    eid int primary key auto_increment,  
    ename varchar(30),  
    sal double,  
    dno int,  
    foreign key emp(dno) references dept (did)  
);
```

3.直接删除部门，这回不行了。

```
delete from dept where did = 1;
```

表的设计（一对多 多对多 一对一） 一对多（重要）

- 1.看图
- 2.例子：部门和员工的例子。
- 3.建表原则：分清除一方和多方。在多方的表中添加一个字段，作为该表的外键，指向一方表的主键。

多对多（重要）

- 1.看图
- 2.学生选课。
- 3.如果是多对多的关系，创建中间表。把表的关系拆成两个一对多。在中间表至少包含两个字段，作为该表的外键指向一方表的主键。

一对一（了解）

- 1.看图

简单购物网站

- 1.用户、订单、商品、分类
- 2.实体的表中包含属性（省略）
- 3.实体与实体之间的关系？创建表的时候需要指定之间的关系。（一个用户可以产生多个订单，一个订单中包含多个商品，一个分类下有多个商品）
- 4.创建表结构

多表的查询（重要）

1. 了解笛卡尔积（两个结果的乘积）

假如说是有两个表A和表B

A		B		
aid	aname	bid	bname	aid
a1	a11	b1	b11	a1
a2	a22	b2	b22	a2
		b3	b33	a1

`select * from A,B;` 查询的结果产生的结果就是笛卡尔积

结果：

a1	a11	b1	b11
a1	a11	b2	b22
a1	a11	b3	b33
a2	a22	b1	b11
a2	a22	b2	b22
a2	a22	b3	b33

使用部门和员工两个表演示笛卡尔积

`select * from dept,emp;`

多表查询之内链接

0. 前提条件：两个表有联系，通过外键关联。

1. 普通内链接

- * 语法：关键字 ... inner join ... on 条件;
- * 注意：
 - * 在inner join关键字之前写表1
 - * 在inner join关键字之后写表2
 - * on的后面写条件：（表1是dept，表2是emp）dept.did = emp.dno
- * 语句：`select * from dept inner join emp on dept.did = emp.dno;`

2. 隐式内链接（用的最多的）

- * 语法：`select ... from 表1,表2 where 表1.字段 = 表2.字段;`
- * 语句：`select * from dept,emp where dept.did = emp.dno;`
- * 别名：`select * from dept d,emp e where d.did = e.dno;`
- * 指定字段：`select d.dname,e.ename,e.sal from dept d,emp e where d.did = e.dno;`

多表查询之外链接

1. 左外链接（左链接）

- * 语法：`... 表1 left outer join 表2 on 表1.字段 = 表2.字段`
- * 语句：`select * from dept left outer join emp on dept.did = emp.dno;`
- * 特点：看左表，默认把左表中的所有数据都查询出来，再查询出有关联的数据。

2. 右外链接

- * 语法：`... 表1 right outer join 表2 on 表1.字段 = 表2.字段`
- * 语句：`select * from dept right outer join emp on dept.did = emp.dno;`

* 特点：看右表，默认把右表中的全部数据都查询出来，再查询出有关联的数据。

向dept和emp表中插入一些数据

```
insert into dept values (null,'牛宝宝部');
```

```
insert into dept values (null,'扯淡部');
```

```
insert into emp values (null,'陈冠希',100,null);
```

```
insert into emp values (null,'张柏芝',200,null);
```

* 测试

* 内链接测试：select * from dept d,emp e where d.did = e.dno;

* 左链接测试：select * from dept left join emp on dept.did = emp.dno;

* 右链接测试：select * from dept right join emp on dept.did = emp.dno;

多表查询的总结

* 如果两张表中的数据没有多余的数据，下面这些个查询的结果都是一样的。

1. 普通内链接查询

* inner join .. on 条件;

2. 隐式内链接查询

* select * from A,B where 条件;

3. 左外链接查询

* left outer join ... on

4. 右外链接查询

* right outer join ... on

* 如果两个表中有多余的数据，使用左链接，查询出的结果先把左表所有的数据全部都查询出来，再把两个表中有关联的数据查询出。

* 如果两个表中有多余的数据，使用右链接，查询出的结果先把右表所有的数据全部都查询出来，再把两个表中有关联的数据查询出。

子查询

1. 你可能一个查询语句不能查出你想要的结果，你可以把多个查询结合到一起使用。

2. 你编写的一个语句中包含多个select关键字。

3. 问题：查询出英语的成绩大于英语的平均分的同学？

1. 查询英语的平均分 select avg(english) from stu;

2. 查询的学生的信息，条件，英语的成绩需要大于平均分。

* select * from stu where english > (select avg(english) from stu);

4. 子查询的特点

* select * from 表 where 条件 > 子查询

* select * from (子查询会先产出结果)

```
select * from (select english from stu);
```

5. 在where条件的后面使用运算符

< > <= >= <>

all 所有 >all

* >all (5,6,7)

```
any 任意      >any  
* >any (5,6,7)
```

```
dept emp
```

想要查询什么？

几张表

条件

查看聪聪所属的部门名称和员工名称？

- * 查询什么？部门的名称 员工的名称

- * 表：部门表和员工表

- * 条件：聪聪所属的部门（去除笛卡尔积），只想查聪聪 name='聪聪'

```
select d.dname,e.ename from dept d,emp e where d.did = e.dno and e.ename = '聪聪';
```

统计每个部门的人数(按照部门名称统计)

- * 查询：部门的名称和人数

- * 表：部门和员工（人数）

- * 条件：where d.did = e.dno 分组（部门分组）

```
select d.dname,count(*) from dept d,emp e where d.did = e.dno group by d.dname;
```

统计部门的平均工资（按部门名称统计）

- * 查询：平均工资

- * 表：部门 员工属性是工资

- * 条件：where d.did = e.dno 分组（部门分组）

```
select d.dname,avg(e.sal) from dept d,emp e where d.did = e.dno group by d.dname;
```

统计部门的平均工资大于公司平均工资的部门

- * 查询：部门

- * 表：部门和员工

- * 条件：部门的平均工资 > 公司的平均工资

```
select d.dname,avg(e.sal) from dept d,emp e where d.did = e.dno group by d.dname having  
avg(e.sal) > (select avg(sal) from emp);
```