Mysql 数据库 多表

今日内容介绍

# \* 课程回顾

\* MySQL数据库

\* 数据库的简介

\* 数据库是一个文件系统，获取通过SQL语句操作。

\* 访问数据库服务器，在服务器内部存在多个数据库，在数据库中是表，表中是字段，对应java类的属性。

表中一条记录对应的java一个实例对象。

\* SQL语句

\* SQL的分类

\* DDL

创建数据库，创建表、删除和修改数据库

(create drop alter)

\* DML

操作数据(insert update delete)

\* DCL

if

\* DQL

操作数据(select)

\* 操作数据库（CRUD）

\* 创建数据库

create database 数据名 character set 编码 collate 校对规则;

\* 删除数据库

drop database 数据库名;

\* 修改数据库

alter database 数据名 character set 编码 collate 校对规则;

\* 查看数据库

show databases; 查询所有的

show create database 数据库名;

\* 操作表

\* 创建表结构

create table 表名(

字段 类型(长度) 约束,

字段 类型(长度) 约束

);

\* 常见的类型

\* 字符串相关 varchar char 区别：varchar长度是可变的，char长度不变的。

\* int bigint(long) double float

\* bit 布尔类型

\* 日期类型 date time datetime timestamp

\* datetime timestamp的区别：前面需要手动录入数据，timestamp默认获取当前的系统时间

\* 约束（单表约束）

\* 主键约束（默认唯一 非空）

\* primary key

\* 自动增长 auto\_increment

\* 唯一约束

\* 代表值是唯一的

\* unique

\* 非空约束

\* not null

\* 删除表

drop table 表名;

\* 修改表结构

alter table 表名 add 新字段 添加字段

alter table 表名 modify 修改类型、约束

alter table 表名 change 修改字段名称

alter table 表名 drop 删除字段

rename 表名称

\* 查询表

desc 表名;

show tables; 所有的表

show create table 表名;

\* 操作数据

\* 插入数据

insert into 表名 (字段1...) values (值1...);

insert into 表名 values(值1...);

\* 修改数据

update 表名 set 字段=值,字段=值... [where]

\* 删除数据

delete from 表名 [where]

truncate 表名;

区别：truncate删除整个表，创建一个新的表。delete一条一条删除。

\* 查询语句

select \* from 表名 [where]

\* 聚集函数

count 获取数量

sum 求和

avg 求平均数

max 最大值

min 最小值

\* 分组

group by having 条件

\* 小的总结

s...f...w...g...h...o

=======================================================================================================

# \* 多表操作

## \* 外键约束

\* 有一个部门的表，还有一个员工表，

create database day12;

use day12;

create table dept(

id int primary key auto\_increment,

dname varchar(30)

);

create table emp(

id int primary key auto\_increment,

ename varchar(20),

salaly double,

);

insert into dept values(null,'研发部');

insert into dept values(null,'销售部');

insert into dept values(null,'人事部');

insert into dept values(null,'扯淡部');

insert into dept values(null,'牛宝宝部');

insert into emp values(null,'班长',10000,1);

insert into emp values(null,'美美',10000,2);

insert into emp values(null,'小凤',10000,3);

insert into emp values(null,'如花',10000,2);

insert into emp values(null,'芙蓉',10000,1);

insert into emp values(null,'东东',800,null);

insert into emp values(null,'波波',1000,null);

update emp set salaly=2500 where eid = 5;

\* 把研发部删除？

\* 研发部下有人员？该操作不合理。

\* 引入外键约束？

\* 作用：保证数据的完整性。

\* 添加外键

语法：alter table emp add foreign key 当前表名(dno) references 关联的表(did);

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

## \* 数据库的设计

\* 一对多 生活中一个部门下有多个员工，一个员工属于一个部门。

\* 在多方需要添加一个字段，并且和一放主键的类型必须是相同的。

\* 把该字段作为外键指向一方的主键。

\* 一方部门

\* 多方员工

\* 多对多

\* 学生可以选择多门课程，课程又可以被多名学生选择。

\* 建表原则：

\* 拆开两个一对多的关系，中间创建一个中间表，至少有两个字段。作为外键指向两个多对多关系表的主键。

\* 一对一（了解）

\* 公司，地址，一个公司对应的是一个地址。 一张表包含公司名称、公司地址。

\* 根据公司的业务需求，会把公司这张表拆开，形成一对一。

\* 建表原则

\* 主键对应

\* 唯一外键对应

## \* 假如有一个（简单）购物的网站(以后课程）

\* 包含哪些实体？ 用户 订单 订单详情表 商品 商品分类

用户表： 订单表： user\_id

订单表： user\_id, 订单详情表： order\_id

订单详情表： order\_id， good\_id,

## 多表的查询简介

\* 笛卡尔积的概念：（了解）

表A 表B

aid aname bid bname

a1 aa1 b1 bb1

a2 aa2 b2 bb2

b3 bb3

\* 查询的语法

select \* from 表A,表B; 返回的结果就是笛卡尔积。

结果：

a1 aa1 b1 bb1

a1 aa1 b2 bb2

a1 aa1 b3 bb3

a2 aa2 b1 bb1

a2 aa2 b2 bb2

a2 aa2 b3 bb3

select \* from dept,emp;

## \* 多表查询

### \* 内连接（用的比较多）

\* 普通内连接

\* 前提条件：需要有外键的。

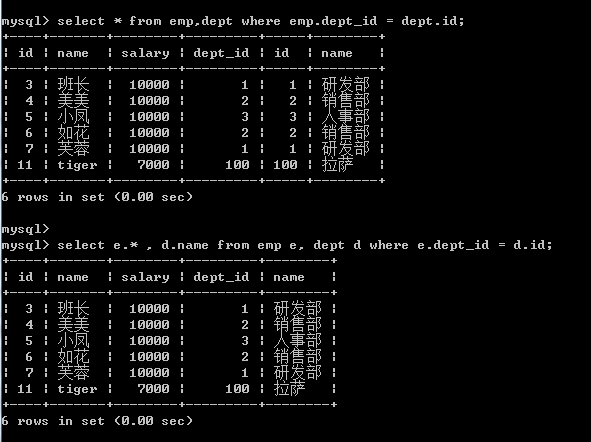
\* 提交关键字 inner join ... on

select \* from dept inner join emp on dept.did = emp.dno;

\* 隐式内连接（用的是最多的）

\* 可以不使用inner join ... on关键字

select \* from dept,emp where dept.did = emp.dno;

·

### \* 外连接

\* 左外链接（看左表，把左表所有的数据全部查询出来）

\* 前提条件：需要有外键的。

\* 语法： 使用关键字 left [outer] join ... on

select \* from dept left outer join emp on dept.did = emp.dno;

\* 右外链接（看右表，把右表所有的数据全部查询出来）

\* 前提条件：需要有外键的。

\* 语法： 使用关键字 right [outer] join ... on

select \* from dept right join emp on dept.did = emp.dno;

### \* 子查询

\* 查询的内容需要另一个查询的结果。

select \* from emp where ename > (select \* from emp where 条件);

any 任意

all 全部

>any 大于结果的最小值

>all 大于结果的最大值

=============================================================================

# 练习数据:

create table dept(

did int primary key auto\_increment,

dname varchar(30)

);

create table emp(

eid int primary key auto\_increment,

ename varchar(20),

salaly double,

dno int

);

查看所有人所属的部门名称和员工名称?

select dept.dname,emp.ename from dept,emp where dept.did = emp.dno;

select d.dname,e.ename from dept d,emp e where d.did = e.dno;

统计每个部门的人数(按照部门名称统计，分组group by count)

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 ，分组group by avg）

select d.dname,avg(salaly) 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（子查询）

\* 公司的平均工资

select avg(salaly) from emp;

\* 部门的平均工资

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

+-----+-------+--------+------+

| eid | ename | salaly | dno |

+-----+-------+--------+------+

| 1 | 班长 | 10000 | 1 |

| 2 | 美美 | 10000 | 2 |

| 3 | 小凤 | 10000 | 3 |

| 4 | 如花 | 10000 | 2 |

| 5 | 芙蓉 | 10000 | 1 |

+-----+-------+--------+------+

+-----+--------+

| did | dname |

+-----+--------+

| 1 | 研发部 |

| 2 | 销售部 |

| 3 | 人事部 |

+-----+--------+

==============================================================================================

+-----+-------+--------+------+

| eid | ename | salaly | dno |

+-----+-------+--------+------+

| 1 | 班长 | 10000 | 1 |

| 2 | 美美 | 10000 | 2 |

| 3 | 小凤 | 10000 | 3 |

| 4 | 如花 | 10000 | 2 |

| 5 | 芙蓉 | 10000 | 1 |

| 6 | 东东 | 800 | NULL |

| 7 | 波波 | 1000 | NULL |

+-----+-------+--------+------+

+-----+----------+

| did | dname |

+-----+----------+

| 1 | 研发部 |

| 2 | 销售部 |

| 3 | 人事部 |

| 4 | 扯淡部 |

| 5 | 牛宝宝部 |

+-----+----------+