# JAVA基础学习日记-Mysql

# --19级学生沈章 记

# Day01:

## 课1：课程回顾

\* 完成注册和登陆的功能。

\* 准备的工作

\* 技术、开源jar包

\* 开发的功能使用MVC模式

\* C：控制层（接收请求和从客户端发送过来的参数）

\* 接收参数(request对象)

\* 为了操作方便（封装数据，内省，BeanUtils开源的工具）

\* 自己new User(); user.setXXX数据

\* 完成业务逻辑的代码（不推荐写在Servlet） new UserService();

\* 接收结果，通过结果把显示的数据发送给JSP。（把结果存入域对象）

\* M：模型层（JavaBean封装数据，其他JavaBean处理业务）

\* 处理业务逻辑根据注册的功能。（用户名不能重名，邮箱不能重名）

\* 需要把结果返回去。

\* V：视图层（完成数据的显示）

\* 到JSP的页面，通过EL表达式取域的值。

## 课2：数据库的简介

\* MySQL数据库

\* 数据库

\* 数据库就是一个文件系统，访问数据的时候需要通过标准的SQL语言来完成。

\* 关系型的数据

\* 保存的实体与实体之间的关系。（用户、商品、订单）

\* 常见的数据库

\* Oracle 公司Oracle（甲骨文）数据产品，收费的大型的数据库。

\* MySQL 开源的，被Oracle收购了，小型的数据库。5.x版本免费，6.x收费了。

\* SQLServer 微软的，收费的中型的数据库。

\* DB2 IBM公司收费的大型的数据库。

\* SyBASE PowerDigener软件（非常牛）

## 课3：MySQL数据库的安装和卸载

MySQL的简介

\* 卸载干净

\* 找到MySQL的安装路径，找到my.ini的配置文件。

\* basedir="C:/Program Files (x86)/MySQL/MySQL Server 5.5/" 安装的路径

\* datadir="C:/ProgramData/MySQL/MySQL Server 5.5/Data/" MySQL存储数据的路径

\* 通过控制面板卸载MySQL

\* 找到上面的两个路径，删除就ok了。

\* 安装了

\* 安装路径不要有中文（\*\*\*\*\*）

\* MySQL默认端口是3306，不要修改。

\* 设置MySQL的编码集（采用UTF-8的编码）

\* 要把黑窗口的勾勾选上。

\* 设置用户名的密码：两行都是密码，第一行是密码，第二行是确认密码。

\* 安装完成。

\* 访问：

cmd > 输入命令 mysql -u（user） root -p（password） 回车

输入密码 回车

## 课4：MySQL重置密码

\* 密码重置

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;（information\_schema，mysql，performance\_schema前三个数据库不要动，test可以自己随意操作 ） 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

## 课5：数据库之间的关系

Mysql服务器与数据、表之间的关系：

机器安装tomcat称为web服务器

机器安装mysql数据库服务器

MySQL

DB

DB

表

表

表

Client

id=1

name=“lisi”

age=23

id=2

name=“wang”

age=24

表的一行称之为一条记录

表中一条记录对应一个java对象的数据

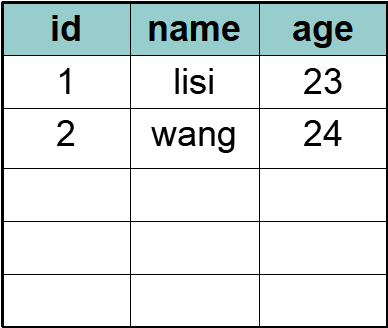
User对象

User对象

User表

列(column)

行(row)



\* MySQL之间的关系（看图）

\* 总结：一个数据库的服务器中有多个数据库，一个数据库中有多个表，

每个表有多个字段。字段和Java中类的属性是对应的。

每一条记录对应是一个Java实例对象。

## 课6：SQL语言的简介

SQL语句（\*\*\*\*\*）

\* SQL的简介

\* Structured Query Language, 结构化查询语言

\* 非过程性的语言

\* 过程性的语言：我下一条语句，需要依赖上一条或者上几条语句。

\* 非过程性的语言：写一条语句，就会执行一个结果。

\* Oracle开发PL/SQL，只能在Oracle使用。

\* SQL Server、Sybase的T-SQL

SQL语言分类：

DDL：数据定义语音（Database Definition Language）

创建数据库、创建表

DML：数据操纵语言（Database Manipulation Language）

插入数据(insert) 修改数据(update) 删除数据(delete)

DCL：数据控制语言（Database Control Language）

if else

DQL：数据查询语言（Database Query Language）

查询数据 select

## 课7：SQL之创建数据库

数据库（CURD -- 增删改查）

\* 创建数据库

\* 语法：create database 数据名称; 创建一个数据了。

\* create database 数据库名称 character set 编码 collate 校对规则;

\* 校对规则：和编码是成对出现的。（参考mysql的数据手册，10.10内容。）

练习

创建一个名称为mydb1的数据库。

create database mydb1;

创建一个使用utf8字符集的mydb2数据库。

create database mydb2 character set 'utf8';

创建一个使用utf8字符集，并带校对规则的mydb3数据库。

create database mydb3 character set 'utf8' collate 'utf8\_bin';

## 课8：修改、删除和查看数据库

\* 查看数据库 show databases;

\* 查询数据库的定义 show create database 数据库;

show create database mydb2;

\* 删除数据库 drop database 数据库名称;

练习

查看当前数据库服务器中的所有数据库

show databases;

查看前面创建的mydb2数据库的定义信息

show create database mydb2;

删除前面创建的mydb1数据库

drop database mydb1;

\* 修改数据库

\* 语法：alter database 数据库 character set 编码 collate 校对规则;

\* 练习：查看服务器中的数据库，并把其中某一个库的字符集修改为gbk

alter database mydb2 character set 'gbk';

\* 其他的操作

\* 切换数据库（\*\*\*\*\*） use db\_name;

\* 查看当前使用的数据库 select database();

## 课9：创建数据库的表结构

### 表（table）（CURD -- 增删改查）

\* 语法：

create table 表名(

字段1 类型(长度) 约束,

字段2 类型(长度) 约束,

字段3 类型(长度) 约束,

字段4 类型(长度) 约束

);

注意：

\* 表名小括号，后面要有分号。

\* 每一行字段后面要有逗号，但是最后一行没有逗号。

\* 数据的类型后面有长度，如果是字符串类型，长度必须加。如果其他类型可以不加。默认长度。int 默认长度11。

public class User{

int id;

String name;

String pass;

String eamil;

String nikename;

}

\* 数据的类型

字符串型

VARCHAR、CHAR

\* varchar和char区别：

\* varchar（经常使用） 长度是可变的。 name varchar(8) 存入数据hello，但是如果存入helloworld报错了。

\* char 长度不可变的。 name char(8) 存入的数据hello，如果不够用空格补全。

\* 效率高：char效果。

大数据类型（一般不用）BLOB、TEXT

BLOB：二进制文件

TEXT：字符

数值型

TINYINT （byte）、SMALLINT（short）、INT、BIGINT（long）、FLOAT、DOUBLE

逻辑性 对应boolean

BIT

日期型

DATE、TIME、DATETIME、TIMESTAMP

\* date 只包含日期

\* time 只包含时分秒

\* datetime和timestamp包含日期和时分秒区别：

\* datetime需要手动录入时间。

\* timestamp不传入数据，默认选择当前系统时间。

\* 练习，创建表的练习

create table employee(

id int,

name varchar(20),

gender varchar(10),

birthday date,

entry\_date date,

job varchar(100),

salary double,

resume text

);

## 课10：简单复习

## 课11：创建表的约束和删除表

### 约束（单表）

\* 主键约束（\*\*\*\*\*）

\* 标识标记该条记录。 通过pramary key声明主键。（默认唯一、非空）

\* auto\_increment 数据库维护主键。自动增长。

\* 唯一约束

\* 值是唯一的。使用unique声明

\* 非空约束

\* 值不能为空 not null

\* 创建新的标签employee2，把约束加上。

create table employee2(

id int primary key auto\_increment,

name varchar(20) unique not null,

gender varchar(10) not null,

birthday date not null,

entry\_date date not null,

job varchar(100) not null,

salary double not null,

resume text not null

);

\* 使用desc 表名; 查看表的信息

\* show tables ; 查看当前库内所有表名

\* show create table 表名; 查看建表语句和字符集

\* 删除表

drop table employee2;

## 课12：修改表

### 修改表

alter table 表名 add 字段 类型(长度) 约束; -- 添加字段

alter table 表名 drop 字段; -- 删除字段

alter table 表名 modify 字段 类型(长度) 约束; -- 修改类型或者约束

alter table 表名 change 旧字段 新字段 类型(长度) 约束 -- 修改字段的名称

rename table 表名 to 新表名; -- 修改表名

alter table 表名 character set utf8; -- 修改字符集

\* 练习

在上面员工表的基本上增加一个image列。

alter table employee add image varchar(20);

修改job列，使其长度为60。

alter table employee modify job varchar(60);

删除gender列。

alter table employee drop gender;

表名改为user。

rename table employee to user;

修改表的字符集为utf8

alter table user character set utf8;

列名name修改为username

alter table user change name username varchar(30);

### 数据（CURD -- 增删改查）(\*\*\*\*\*\*)

\* 添加数据

\* insert into 表名 (字段1,字段2,字段3..) values(值1,值2,值3...); 有几列就插入多少的值。

\* insert into 表名 values(值1,值2,值3...); 插入所有的列

\* 注意：

\* 数据与字段的类型相同。

\* 字段长度需要控制。

\* 字符串或者日期类型需要使用''

\* 向user表中插入数据

insert into user values (1,'xiaofeng','1994-10-10','2011-1-1','HR',19000,'aaa','abc');

insert into user values (2,'美美','1994-10-10','2011-1-1','HR',19000,'aaa','abc');

insert into user values (3,'小风','1994-10-10','2011-1-1','WORKER',21000,'aaa','abc');

insert into user values (4,'芙蓉','1994-10-10','2011-1-1','HR',1000,'aaa','abc');

insert into user values (5,'班长','1994-10-10','2011-1-1','HR',100,'aaa','abc');

## 课13：修改MySQL中文乱码的问题

### 解决中文乱码的问题（\*\*\*\*\*）

[client]

port=3306

[mysql]

default-character-set=gbk

\* 修改完需要重新启动服务。

## 课14：修改和删除数据

mysql快捷键：crtl+backspace可以一次性删除粘贴的内容。

### 修改语句

\* 语法： update 表名 set 字段=值,字段=值... [where ]

\* 如果没有where条件，默认更新所有的记录。

\* 有where提交，选择某一条记录。

将所有员工薪水修改为5000元。

update user set salary=5000;

将姓名为’班长’的员工薪水修改为3000元。

update user set salary=3000 where username='班长';

将姓名为’美美’的员工薪水修改为4000元,job改为BOSS。

update user set salary=4000,job='BOSS' where username='美美';

将班长的薪水在原有基础上增加1000元。

update user set salary = salary+1000 where username='班长';

### 删除数据 delete

语法：delete from 表名 [where ]; 删除数据

truncate 表名; 删除所有的数据

\* truncate 和 delete的区别：

\* truncate删除数据，先删除整个表。再创建一个新的空的表。（效率）

\* delete删除数据，一条一条删除的。（\*\*\*\*\*）

\* 事物（insert update delete）

删除表中名称为’班长’的记录。

\* delete from user where username='班长';

删除表中所有记录。

\* delete from user;

使用truncate删除表中记录。

\* truncate user;

## 课15：select语句一

### 查询语句

\* 语法： select \* from 表名; 查询所有（字段）

select 字段名1,字段名2,字段名3 from 表名; 显示查询字段名

select DISTINCT 字段名 from 表名; 去除重复的数据。

查询表中所有学生的信息。

select \* from stu;

查询表中所有学生的姓名和对应的英语成绩。

select name,english from stu;

过滤表中重复数据。（面试题）

select distinct english from stu;

\* 查询的列可以运算

\* 可以使用别名：使用as 别名 并且as可以省略。

练习：

在所有学生分数上加10分特长分。

select name,math+10,english+10,chinese+10 from stu;

统计每个学生的总分。

select name,math+english+chinese from stu;

使用别名表示学生分数。

select name,(math+english+chinese) as sum from stu;

### 使用where条件过滤

查询姓名为班长的学生成绩

select \* from stu where name='班长';

查询英语成绩大于90分的同学

select name,english from stu where english < 15;

查询总分大于200分的所有同学

select name,math+english+chinese from stu where (math+english+chinese) > 200;

### 常用的符号

> < <= >= = <>（不等于）

in(范围内取内容)

like -- 模糊查询 写法：like '张\_或者%'; \_和%区别：占位符。\_只一个%可以有多个

%的写法 like '%张'; 结果XXX张

like '张%'; 结果张XXX

like '%张%'; 只要有张就行

is null -- 判断是否为null

and -- 并且

or -- 或者

not -- 不成立

create table stu(

id int primary key auto\_increment,

name varchar(20),

english int,

math int,

chinese int

);

insert into stu values (44,21,66,null,'张三');

insert into stu values (77,66,88,null,'旺财');

insert into stu values (99,55,99,null,'李四');

\* 练习

查询英语分数在 80－90之间的同学。

select \* from stu where english >80 and english <90;

select \* from stu where english between 80 and 90;

查询数学分数为18,78,46的同学。（in）

select \* from stu where math in(18,78,46);

查询所有姓班的学生成绩。

select \* from stu where name like '班%';

查询数学分>80，语文分>80的同学。

select \* from stu where math >80 or chinese > 80;

## 课16：select语句之排序

\* 排序 使用order by 升序默认的(asc)/降序(desc)

\* 出现select的语句末尾。

练习

对数学成绩排序后输出。

select name,math from stu order by math;

对总分排序按从高到低的顺序输出

select name,math+english+chinese from stu order by (math+english+chinese) desc;

对学生成绩按照英语进行降序排序，英语相同学员按照数学降序

select \* from stu order by english desc,math desc;

对姓美的学生成绩排序输出

select \* from stu where name like '美%' order by english desc;

## 课17：聚集函数

聚集函数

### count 获取数量

练习：

统计一个班级共有多少学生？

select count(\*) from stu;

统计数学成绩大于90的学生有多少个？

select count(\*) from stu where math > 90;

统计总分大于150的人数有多少？

select count(\*) from stu where (math+english+chinese) > 150;

### sum 求和（忽略null值）

可以同ifnull(xxx,0)

统计一个班级数学总成绩？

select sum(math) from stu;

统计一个班级语文、英语、数学各科的总成绩

select sum(math),sum(english),sum(chinese) from stu;

统计一个班级语文、英语、数学的成绩总和

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

select sum(math)+sum(english)+sum(chinese) from stu;

统计一个班级语文成绩平均分

select sum(chinese) / count(\*) from stu;

### avg 平均数

练习：

求一个班级数学平均分？

select avg(math) from stu;

求一个班级总分平均分

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

### max 最大值

select max(math) from stu;

### min 最小值

select min(math) from stu;

## 课18：分组语句

### group by 分组（一起使用）

条件过滤需要是having，不能使用where

练习：对订单表中商品归类后，显示每一类商品的总价.

select product,count(\*),sum(price) from orders group by product;

练习：查询购买了几类商品，并且每类总价大于100的商品

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

### 小结

select 语句 ： S-F-W-G-H-O 组合 select ... from ... where ... group by... having... order by ... ;

顺序不能改变

## 课19：课程复习

# Day02:

## 课1：课程回顾

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

## 课2：外键约束

多表操作

\* 外键约束

\* 有一个部门的表，还有一个员工表，

create database day16;

use day16;

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

);

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);

## 课3：一对多的建表原则

数据库的设计

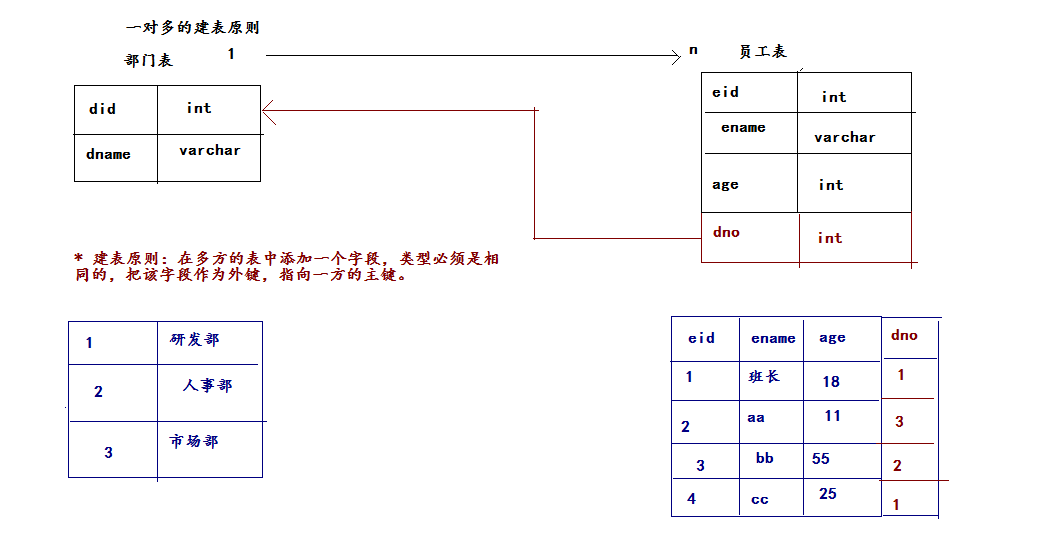
\* 一对多 生活中一个部门下有多个员工，一个员工属于一个部门。

\* 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



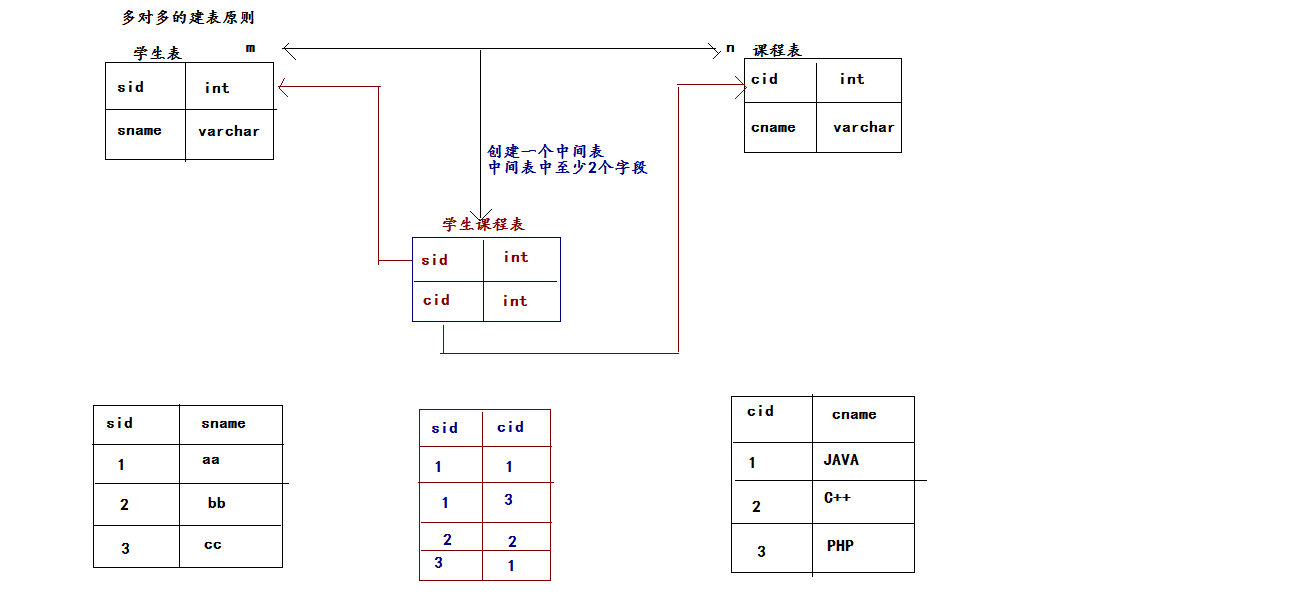
## 课4：多对多和一对一的建表原则

\* 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）

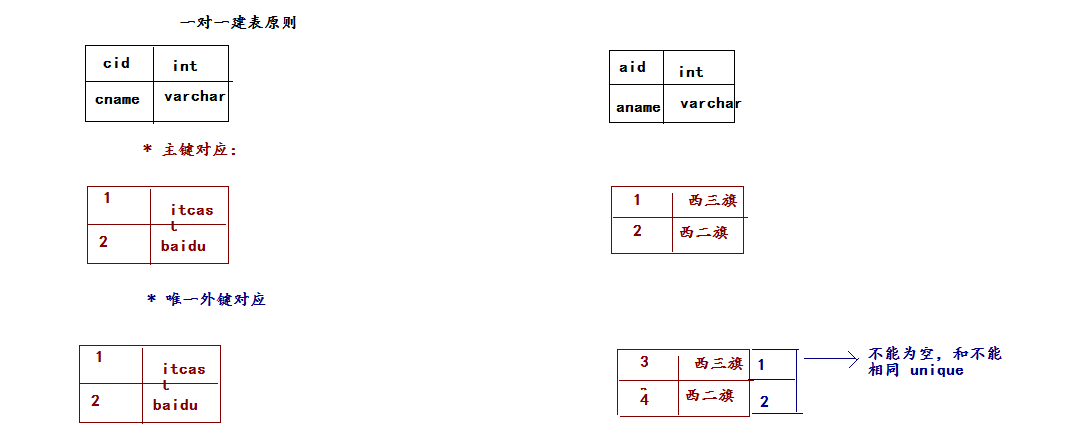
\* 公司，地址，一个公司对应的是一个地址。 一张表包含公司名称、公司地址。

\* 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



## 课5：简单的购物网站分析

\* 假如有一个（简单）购物的网站

\* 包含哪些实体？ 用户 订单 商品 分类

\* 多表的查询

\* 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

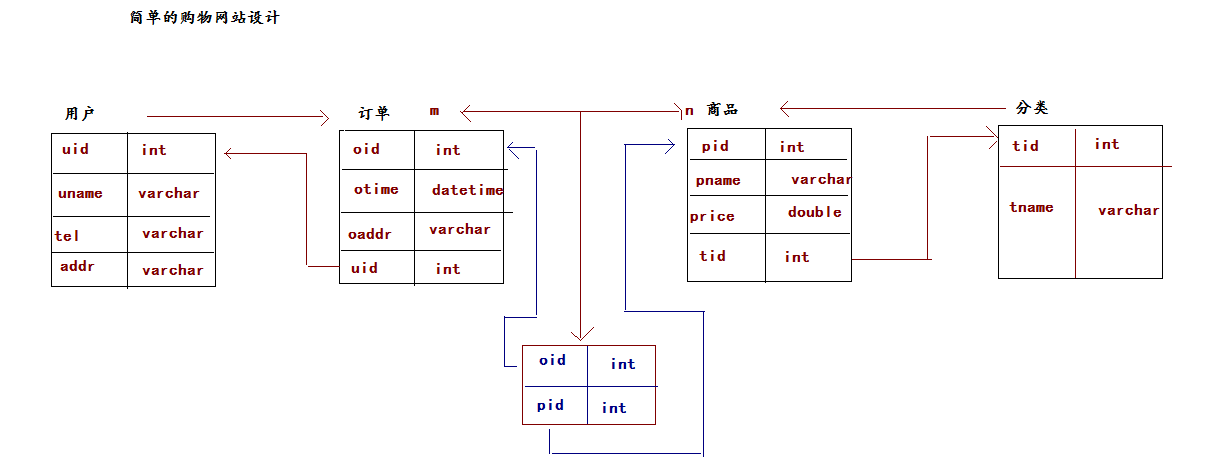
a1 aa1 b3 bb3

a2 aa2 b1 bb1

a2 aa2 b2 bb2

a2 aa2 b3 bb3

select \* from dept,emp;



## 课6：多表查询笛卡尔积和内链接

多表查询笛卡尔积（各个表的字段数量的乘积）

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;

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

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

| did | dname | eid | ename | salaly | dno |

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

| 1 | 研发部 | 1 | 班长 | 10000 | 1 |

| 1 | 研发部 | 5 | 芙蓉 | 10000 | 1 |

| 2 | 销售部 | 2 | 美美 | 10000 | 2 |

| 2 | 销售部 | 4 | 如花 | 10000 | 2 |

| 3 | 人事部 | 3 | 小凤 | 10000 | 3 |

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

## 课7：外连接

\* 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;

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

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

| did | dname | eid | ename | salaly | dno |

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

| 1 | 研发部 | 1 | 班长 | 10000 | 1 |

| 1 | 研发部 | 5 | 芙蓉 | 10000 | 1 |

| 2 | 销售部 | 2 | 美美 | 10000 | 2 |

| 2 | 销售部 | 4 | 如花 | 10000 | 2 |

| 3 | 人事部 | 3 | 小凤 | 10000 | 3 |

| 4 | 扯淡部 | NULL | NULL | NULL | NULL |

| 5 | 牛宝宝部 | NULL | NULL | NULL | NULL |

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

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

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

| did | dname | eid | ename | salaly | dno |

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

| 1 | 研发部 | 1 | 班长 | 10000 | 1 |

| 2 | 销售部 | 2 | 美美 | 10000 | 2 |

| 3 | 人事部 | 3 | 小凤 | 10000 | 3 |

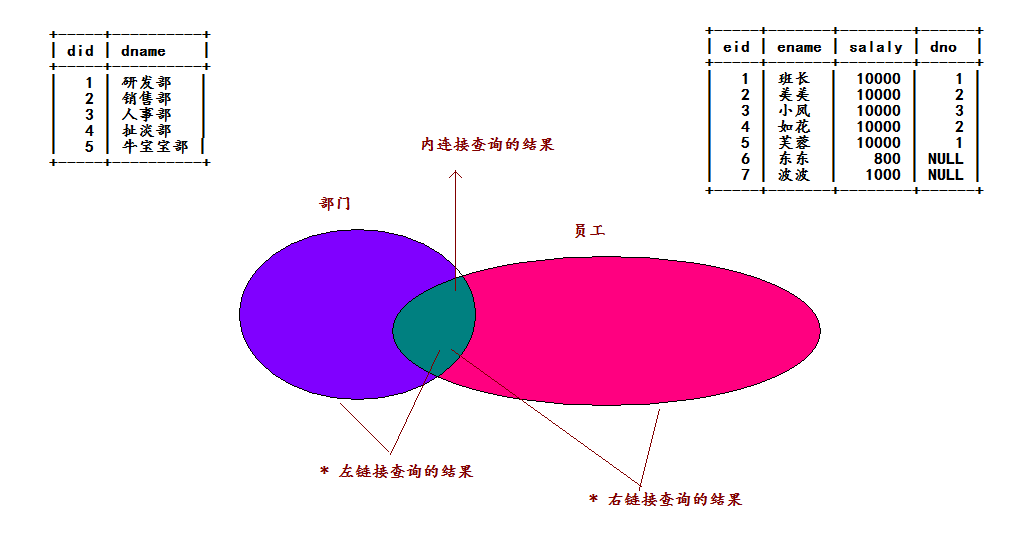
| 2 | 销售部 | 4 | 如花 | 10000 | 2 |

| 1 | 研发部 | 5 | 芙蓉 | 10000 | 1 |

| NULL | NULL | 6 | 东东 | 800 | NULL |

| NULL | NULL | 7 | 波波 | 1000 | NULL |

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



## 课8：子查询

\* 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

## 课9：简单练习

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);

## 课10：工具

为mysql中添加.sql文件： source xxx\xxx\xxx...\xxx.sql