【济南中心】JavaEE就业班同步笔记第一阶段：JavaWeb之Mysql

小鲁哥哥 • 2016-12-17 16:23:33

点击查看济南黑马校区最新开班计划

【济南中心】JavaEE就业班同步笔记第一阶段：JavaWeb之数据库部分--MySQL

1. MYSQL的回顾：

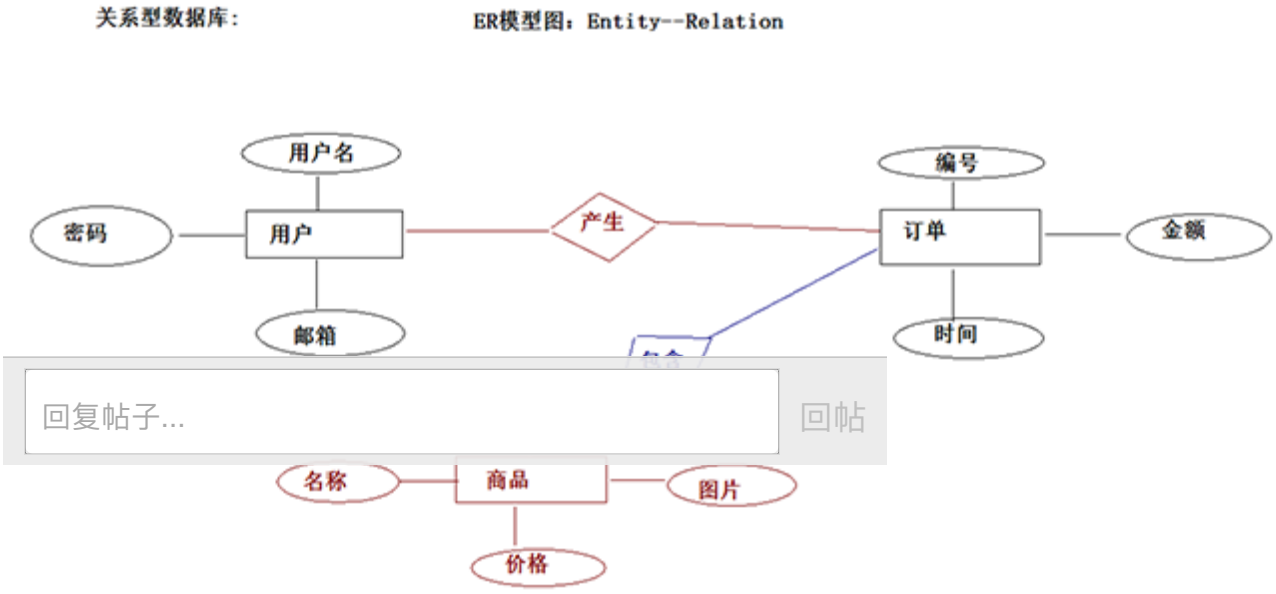
1.1 MySQL的概述：

1.1.1 什么是数据库：

数据库：就是一个文件系统,这个文件必须通过标准的SQL访问.

1.1.2什么是关系型数据库：

关系型的数据库存放的都是实体之间的关系.



1.1.3 常用的关系型数据库：
MySQL：免费的小型的数据库,现在被Oracle收购.
Oracle：Oracle公司收费的大型的数据库.
SQLServer ： 微软公司收费中型的数据库.
DB2：IBM公司收费的大型的数据库.
SyBase：SyBase公司收费的数据库.已经被淘汰.PowerDesigner数据建模的工具.
SQLite：小型的嵌入式的数据库.
***** Java程序中经常使用的数据库
MySQL
Oracle
1.1.4数据库存储的结构：
1.2. SQL的概述：
1.2.1 什么是SQL:
SQL:结构化的查询语言.
1.2.2 SQL分类:
DDL:数据定义语言
    * create,alter,drop...
DML:数据操纵语言
    * update,insert,delete
DCL:数据控制语言
    * grant,if..
DQL:数据查询语言
    * select
1.2.3 SQL的特点：
非过程性语言:一条语句就会有一个运行的结果.
1.3 使用SQL
1.3.1 使用SQL操作数据库(对数据库的CRUD的操作)
【创建数据库】
语法:
* create database 数据库名称 [character set 字符集 collate 字符集校对];
练习:
* 创建db1;

[mw_shl_code=sql,true]create database db1;[/mw_shl_code]

* 创建一个带有字符集的数据库db2;

[mw_shl_code=sql,true]create database db2 character set gbk;[/mw_shl_code]

* 创建一个带有字符集和校对规则的数据库db3;

[mw_shl_code=sql,true]create database db3 character set utf8 collate utf8_bin;[/mw_shl_code]

【查看数据库】

语法:

* 查看数据库服务器中所有的数据库:

[mw_shl_code=sql,true]show databases;[/mw_shl_code]

* 查看某个数据库的定义信息.

[mw_shl_code=sql,true] show create database 数据库名;[/mw_shl_code]

* 查看当前正在使用的数据库信息.

[mw_shl_code=sql,true]select database();[/mw_shl_code]

【删除数据库】

语法:

* 删除数据库:
  * drop database 数据库名;

【修改数据库】

语法:

* 修改数据库修改的是的数据库的字符集和校对规则.
  * alter database 数据库名 character set 新字符集 collate 校对规则;

【切换数据库】

语法:

* use 数据库名称;

1.3.2 使用SQL操作数据库中的表(对数据库的表CRUD的操作)

【创建表】

语法:

```
* create table 表名 (
    字段名 类型(长度) 约束,
    字段名 类型(长度) 约束,
    字段名 类型(长度) 约束
);
```

数据类型:

Java类型:	MySQL
byte/short/int/long	tinyint/smallint/int/bigint
String	char/varchar
float	float
double	double
boolean	bit
Date	date/time/datetime/timestamp
文本文件	Text
二级制文件	BLOB

char与varchar：

* 区别：char是固定长度的字符串,varchar可变长度的字符串.

　　如果插入一个字符串hello 插入到char 那么 插入hello　.插入到varchar中 插入hello

* datetime和timestamp都是既有日期又有时间的日期类型

　　区别：datetime需要使用外部传入的日期.如果没传这个值就是Null. timestamp会使用系统当前的时间作为这个值的默认值.

***** Oralce使用CLOB/BLOB

***** MYSQL中除了字符串类型需要设置长度其他的类型都有默认长度.

约束：

单表约束：

* 主键约束：primary key (默认就是唯一非空的)
* 唯一约束：unique
* 非空约束：not null

创建一个表：

***** 创建表之前先选择数据库:use 某个数据库;

[mw_shl_code=sql,true]create table employee(
　　eid int primary key auto_increment,
　　ename varchar(20) not null,
　　email varchar(30) unique,

```
    birthday date,
    job varchar(20),
    resume text
);[/mw_shl_code]
```

【表的查看】

查看数据库中有哪些表：

* show tables;

查看表结构:

* desc 表名;

【表的删除】

表的删除:

* drop table 表名;

【表的修改】

修改表添加列：

* alter table 表名 add 列名 类型(长度) 约束;

[mw_shl_code=sql,true]alter table employee add image varchar(50);[/mw_shl_code]

修改表删除列：

* alter table 表名 drop 列名;

[mw_shl_code=sql,true]alter table employee drop job;[/mw_shl_code]

修改表的列的类型长度及约束:

* alter table 表名 modify 列名 类型(长度) 约束;

[mw_shl_code=sql,true]alter table employee modify image varchar(80) not null;[/mw_shl_code]

修改表的列名

* alter table 表名 change 旧列名 新列名 类型(长度) 约束;

[mw_shl_code=sql,true]alter table employee change image eimage varchar(60);[/mw_shl_code]

修改表名

* rename table 旧表名 to 新表名;

[mw_shl_code=sql,true] rename table employee to user;[/mw_shl_code]

修改表的字符集：

* alter table 表名character set 字符集;

[mw_shl_code=sql,true]alter table user character set gbk;[/mw_shl_code]

1.3.3　使用SQL操作数据库中的表的记录(对表的记录的CRUD的操作)

【插入记录】

语法

* insert into 表名 (列名,列名,...) values (值1,值2,...);　　---插入指定列的值

* insert into 表名 values (值1,值2,...);　　---插入所有列的值

注意事项:

* 列名的个数与值的个数对应.

* 列的类型与值的类型对应.位置也要对应.

* 列的类型如果是字符串或者日期,写值的时候使用单引号将值引起来.

* 插入的值的最大长度不能超过列的最大长度.

插入记录:

* 插入某几列的值:

[mw_shl_code=sql,true]insert into employee (eid,ename,email) values (null,'aaa','aaa@itcast.cn');[/mw_shl_code]

* 插入所有列的值:

[mw_shl_code=sql,true]insert into employee values (null,'bbb','bbb@itcast.cn','1990-09-01','HR','I am HR');[/mw_shl_code]

插入中文:

[mw_shl_code=shell,true]insert into employee (eid,ename,email) values (null,'张三','aaa@163.cn');[/mw_shl_code]

ERROR 1366 (HY000): Incorrect string value: '\xD5\xC5\xC8\xFD' for column 'ename' at row 1

***** 插入中文问题的解决:

[mw_shl_code=shell,true]show variables like '%character%';[/mw_shl_code]

```
| character_set_client     | utf8
|
| character_set_connection | utf8
|
| character_set_database   | utf8
|
| character_set_filesystem | binary
|
| character_set_results    | utf8
|
| character_set_server     | utf8
|
| character_set_system     | utf8
```

***** 找到MYSQL的安装路径/my.ini文件:

```
[client]

port=3306

[mysql]

default-character-set=gbk
```

**** 重新加载mysql的配置文件：

[mw_shl_code=shell,true]services.msc[/mw_shl_code]

* 停止mysql的服务,重新启动mysql服务.

* 执行之前的SQL语句.

【修改记录】

语法：

* update 表 set 列名=值,列名=值 [where 条件];

注意事项：

* 列名和值类型也要一致.

* 值不能超过列的最大长度.

* 值是字符串或日期,需要使用单引号.

练习：

* 修改employee表中所有记录的job为WORKER

  [mw_shl_code=sql,true]update employee set job='WORKER';[/mw_shl_code]

* 修改employee表将name为aaa的邮箱改为aaa@163.com

  [mw_shl_code=sql,true]update employee set email = 'aaa@163.com' where ename = 'aaa';[/mw_shl_code]

* 修改employee表将name为bbb的邮箱改为bbb@163.com同时修改job为HR

  [mw_shl_code=sql,true]update employee set email = 'bbb@163.com' , job='HR' where ename='bbb';[/mw_shl_code]

【删除记录】

语法：

* delete from 表 [where 条件];

注意事项：

* 删除表中的一行记录,不能删除某列值

* 如果没有条件删除表中的所有列.

练习：

* 删除id为8的记录：

[mw_shl_code=sql,true]delete from employee where eid = 8;[/mw_shl_code]

* 删除所有记录：

[mw_shl_code=sql,true]delete from employee;[/mw_shl_code]

删除表中的所有记录truncate table 表名 和 delete from 表区别?

* 区别：

* truncate table 删除表的记录：将整个表删除掉,重新创建一个新的表.truncate属于DDL.

* delete from 删除表的记录：一条一条进行删除. delete属于DML。

* 事务管理 只能作用在DML语句上.如果再一个事务中使用delete删除所有记录，可以找回.

【基本查询】

查询语句：

* select [distinct] *l列名 from 表 [where 条件];

准备：

```
[mw_shl_code=sql,true]create table exam(
    id int primary key auto_increment,
    name varchar(20),
    english int,
    chinese int,
    math    int
);[/mw_shl_code]
[mw_shl_code=sql,true]insert into exam values (null,'张三',85,74,91);
insert into exam values (null,'李四',95,90,83);
insert into exam values (null,'王五',85,84,59);
insert into exam values (null,'赵六',75,79,76);
insert into exam values (null,'田七',69,63,98);
insert into exam values (null,'李四',89,90,83);[/mw_shl_code]
```

查询所有记录：

[mw_shl_code=sql,true]select * from exam;[/mw_shl_code]

查询这个班级人的姓名和英语成绩：

[mw_shl_code=sql,true]select name,english from exam;[/mw_shl_code]

查询英语成绩,将重复英语成绩去掉：

[mw_shl_code=sql,true]select distinct english from exam;[/mw_shl_code]

查询李四的学生成绩:

[mw_shl_code=sql,true]select * from exam where name='李四';[/mw_shl_code]

查询名称叫李四并且英语成绩大于90的

s[mw_shl_code=sql,true]elect * from exam where name='李四' and english >90;[/mw_shl_code]

将成绩+10分进行显示:

[mw_shl_code=sql,true]select name ,english+10,chinese+10 ,math+10 from exam;[/mw_shl_code]

显示这个人的名称和对应总成绩的分数:

[mw_shl_code=sql,true]select name,english+chinese+math from exam;[/mw_shl_code]

使用as起别名,as可以省略.

[mw_shl_code=sql,true]select name , english+chinese+math as sum from exam;[/mw_shl_code]

【条件查询】

where语句后面可以加:

条件的关键字:

= , > , >= , <, <= , <>

like中可以使用占位符: _ 和 % ： 下划线匹配一个字符, %:可以匹配任意多个字符.

* like ‘张%’; like ‘张_’; like ‘%明’; like ‘%明%’;

in 后跟着一组值.

* id in (1,2,3)

and or not

【排序查询】

order by 对数据进行排序.默认升序. （asc升序,desc降序）

* 查询所有学生的信息,并且按语文成绩进行排序.

  [mw_shl_code=sql,true]select * from exam order by chinese;[/mw_shl_code]

* 查询所有学生的信息,并且按语文成绩进行降序排序.

  s[mw_shl_code=sql,true]elect * from exam order by chinese desc;[/mw_shl_code]

* 查询学生的信息,按照英语成绩降序排序，如果英语成绩相同,按照语文降序.

  s[mw_shl_code=sql,true]elect * from exam order by english desc, chinese desc;[/mw_shl_code]

* 查询姓李的学生的信息,同时按照英语升序排序.

  [mw_shl_code=sql,true]select * from exam where name like '李%' order by english asc;[/mw_shl_code]

【聚合函数】

sum()

count()

max()

min()
avg()
* 查询每个学生总成绩:
[mw_shl_code=sql,true]select name,(english+chinese+math) from exam;[/mw_shl_code]
* 统计所有学生的总分:
[mw_shl_code=sql,true] select sum(english+chinese+math) from exam; -- ifnull(english,0)
select sum(english)+sum(chinese)+sum(math) from exam;[/mw_shl_code]
* 统计学生的个数:
[mw_shl_code=sql,true]select count(*) from exam;[/mw_shl_code]
* 统计英语成绩的最高分:
[mw_shl_code=sql,true]select max(english) from exam;[/mw_shl_code]
* 统计语文成绩的最低分:
[mw_shl_code=sql,true]select min(chinese) from exam;[/mw_shl_code]
* 统计英语成绩平均分:
[mw_shl_code=sql,true] select avg(english) from exam;[/mw_shl_code]
【分组】
group by
创建一个订单详情的表:
* 统计订单中的每类商品所购买的个数:
[mw_shl_code=sql,true]SELECT product,COUNT(*) FROM orderitem GROUP BY product;[/mw_shl_code]
* 统计订单中的每类商品所花的金额:
[mw_shl_code=sql,true] SELECT product,SUM(price) FROM orderitem GROUP BY product;[/mw_shl_code]
* 统计订单中的每类商品所花总金额大于2000信息.
[mw_shl_code=sql,true]SELECT product,SUM(price) FROM orderitem GROUP BY product HAVING SUM(price) > 2000;[/mw_shl_code]
* 统计订单中名称有电子的商品并且所花金额大于1500同时按照价格降序排序:
[mw_shl_code=sql,true] SELECT product,SUM(price) FROM orderitem WHERE product LIKE '电%' GROUP BY product HAVING SUM(price) > 1500 ORDER BY SUM(price) DESC;[/mw_shl_code]
【SQL的查询语句的总结】
顺序: s...f...w...g...h...o...;

2. 案例一：将商城的案例中的表结构进行分析：

2.1 需求:

在最后的综合案例中,会创建数据库,为数据库中创建很多表.表与表之间是有关系存在,分析表之间关系并且完成表的创建.

2.2 分析：

2.2.1 技术分析

【数据库的多表设计】

数据库都是关系型的数据库,存的是实体之间的关系.实体之间有哪些关系?

实体的关系总结起来就有三种关系：

一对多:

* 客户和订单：一个客户可以产生多个订单,一个订单只能属于是某一个客户.
* 部门和员工：一个部门下可以有多个员工,一个员工只能属于某一个部门.

多对多:

* 学生和课程：一个学生可以选择多门课程,一门课程可以被多个学生选择.
* 订单和商品：一个订单中包含多个商品,一个商品也可以出现多个订单中.

一对一：

* 公司和地址：一个公司只能有一个注册地址,一个地址也只能被一个公司注册.

【多表的设计】

一对多的关系的建表原则：

* 在多的一方创建一个字段,这个字段作为外键执向一的一方的主键.

多对多的关系的建表原则：

* 创建一个第三种表，中间表中至少需要两个字段分别作为外键执向多对多双方的各自的主键.

一对一的关系的建表原则：

* 唯一外键对应：假设一对一的双方是一对多的关系.在多的一方创建外键执向一的一方的主键.需要在外键上添加一个unique约束.
* 主键对应：将一对一的双方的主键建立映射.

【使用SQL创建一对多的关系】

创建客户表:

[mw_shl_code=sql,true]create table customer(
    cid int primary key auto_increment,
    cname varchar(20)
);[/mw_shl_code]

创建订单表

[mw_shl_code=sql,true]create table orders(
    oid int primary key auto_increment,
    addr varchar(50),
    cid int
);[/mw_shl_code]

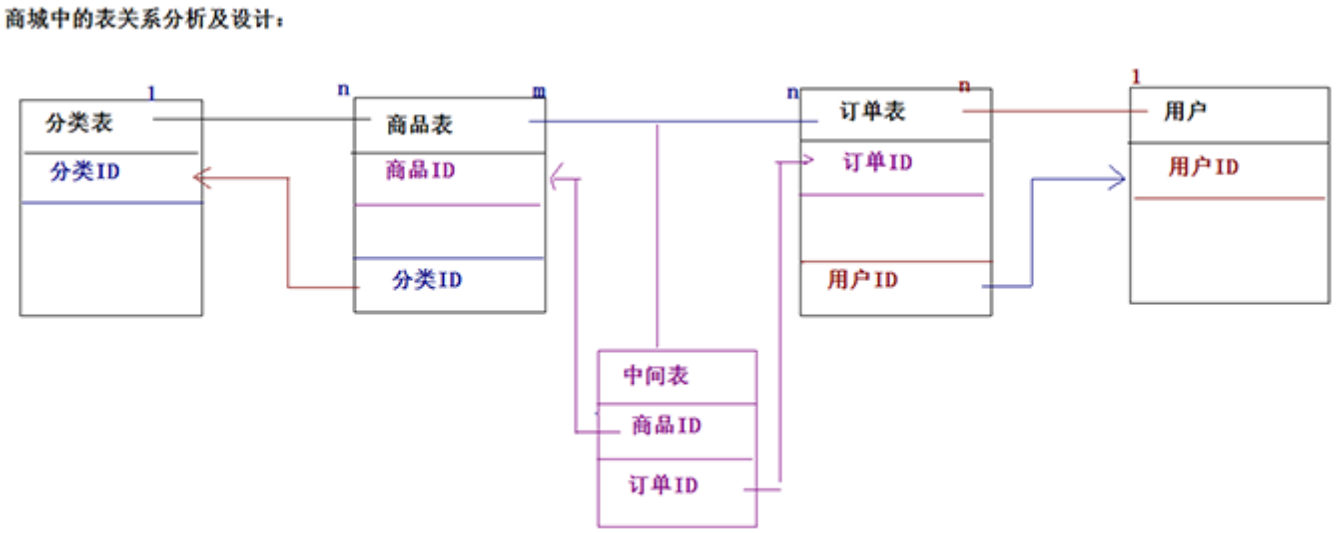
***** 约束：用来保证数据的完成型.

* 多表约束：外键约束！！！

***** 给orders表中的cid添加外键约束.

[mw_shl_code=sql,true]alter table orders add foreign key (cid) references customer(cid);[/mw_shl_code]

2.3 商城模型分析：



3 案例二：使用SQL完成多表的查询：

3.1 需求：

使用多表的查询，完成某个分类下的商品的显示.很多的情况下都需要使用多表的查询.

3.2 分析：

3.2.1 技术分析：

【多表的查询的SQL】

多表的查询的方式：

* 交叉连接：

* select * from A,B; --- 获得的是两个表的笛卡尔积.

* 内连接: inner join -- inner 可以省略

* 显式内连接：select * from A inner join B on 条件;

[mw_shl_code=sql,true] SELECT * FROM customer c INNER JOIN orders o ON c.cid = o.cid;[/mw_shl_code]

* 隐式内连接：select * from A,B where 条件;

[mw_shl_code=sql,true]SELECT * FROM customer c ,orders o WHERE c.cid = o.cid;[/mw_shl_code]

* 外连接：outer join -- outer 可以省略

* 左外连接：left outer join -- select * from A left outer join B on 条件;

[mw_shl_code=sql,true]SELECT * FROM customer c LEFT OUTER JOIN orders o ON c.cid = o.cid;[/mw_shl_code]

* 右外连接：right outer join -- select * from A right outer join B on 条件;

[mw_shl_code=sql,true]SELECT * FROM customer c RIGHT OUTER JOIN orders o ON c.cid = o.cid;[/mw_shl_code]

【多表查询的子查询】

一个SQL语句查询的过程中需要依赖另一个查询语句.

[mw_shl_code=sql,true]SELECT * FROM customer c,orders o WHERE c.cid = o.cid AND c.cid IN (SELECT cid FROM orders WHERE addr LIKE '海淀%');[/mw_shl_code]

【多表练习】

按客户名称统计订单的个数.

[mw_shl_code=sql,true]SELECT c.cname,COUNT(*) FROM customer c,orders o WHERE c.cid = o.cid GROUP BY c.cname;[/mw_shl_code]

回帖

44条回帖

黑马嗨嗨嗨

收藏了。赞赞赞赞

沙发 • 2016-12-18 19:54:59

回帖

孤独于我

收藏啊666

藤椅 • 2016-12-19 21:31:50

回帖

小虎同学

收藏了哈哈

板凳 • 2016-12-22 11:34:31

回帖

来自宇宙超级黑马专属苹果客户端

嗯~得坚持

棒棒的棒棒的棒棒的

报纸 • 2016-12-22 23:10:30

回帖

小陵不懂

学习一下,真不错

地板 • 2016-12-26 09:33:52

回帖

小虎同学

真心不错，回来赞一下

7# • 2016-12-27 23:48:13

回帖

来自宇宙超级黑马专属苹果客户端

@211

收藏了，谢谢！

8# • 2017-1-1 16:31:07

回帖

事难懂

东西不错，果断收藏了

9# • 2017-1-3 13:18:47

回帖

hittor

感谢分享！

10# • 2017-1-3 19:16:03 回帖

Yin灬Yan

好帖子 收藏下

11# • 2017-1-10 00:07:58 回帖

zmanx

原来是黑马官方的啊,怪不得啊,好厉害哦

12# • 2017-2-12 22:13:37 回帖

爱吃橘子的小泽

顶一下,祝黑马越办越好

13# • 2017-2-12 22:15:15 回帖

zmanx

[mw_shl_code=java,true]public static void function(){
    Date date = new Date();
    system.out,println(date);
}[/mw_shl_code]

14# • 2017-2-12 22:25:00 回帖

恋上零冰度

过几天就要学了，非常感激，哈哈，已收藏。

15# • 2017-2-12 23:12:50 回帖

来自宇宙超级黑马专属安卓客户端

yibuhuiba

谢谢分享啊

16# • 2017-2-18 19:14:38 回帖

a690223483

收藏收藏 是真的棒

17# • 2017-2-22 21:00:49 回帖

zhangkaitong

谢谢分享 赞赞赞！！！！！

18# • 2017-2-24 13:27:06 回帖

xulinml

谢谢老师~！~感谢感谢

19# • 2017-2-28 08:54:10 回帖

为过去的我买单

谢谢分享！

20# • 2017-3-2 10:10:38 回帖