课改：

学习方式：

1. 按部就班一步一步踏实脚印走（太多啰嗦地方）java面向对象，封装，继承，多态
2. 另辟蹊径（拔苗助长，我还不会走，老师就让我跑）

田忌赛马

|  |  |
| --- | --- |
| 我们学员 | 资深的技术人员：项目经理，技术经理，架构师，技术副总 |
| 弱项：java基础、javaWeb、3大框架  （面试技巧，最重要是表现自己好学和渴望进入这家公司态度） | 强项：  Java1991年17年了，市场上企业中有大把java程序员，在企业中通过非常多的项目来真实锤炼，周期3-20年。真实企业环境。 |
| 在培优课程中优势：   1. 互联网架构为核心   Docker容器化，1s启动一个服务，瞬间达到百万级别（阿里双11，2017,2018都是运行在docker之上）  Nginx+redis(x)+mycat  Rabbitmq+docker+dubbo  Jsoup 爬虫，薪资9k+  Lucene+solr/es 10K+全文检索工程师，自己百度  2）大数据项目  Hadoop，HDFS，flume，hive，storm，zookeeper，kafka  真实项目网络流量日志分析PV,UV,VV | 弱项：在工作中接触的技术的面小！越大公司越窄。跳槽！  对新技术消化能力差，（为了现有系统稳定，不可能直接把新的技术引入）但是它渴望，企业渴望！  技术变化非常快！它虽然很难实质推动企业技术变更，但是他希望有人来做这个事情，  大多企业：  大型，互联网架构+大数据  中型，dubbo互联网架构+ssm（springmvc+spring+mybatis）三大框架  小型，ssh（struts2+spring+hibernate） |

Jsp动态网站，和服务器可以交互。

传递参数，活动，和数据库一结合，这些数据就能从数据库获取。当数据库修改了，这些页面的数据重新加载页面时，就变化成新的内容。

Servlet负责数据准备，传递给jsp页面。

从数据库获取数据jdbc，数据返回值给变量

String doorName = getDoorName(去访问数据库返回店名)

Servlet是如何把信息传递给jsp（request）

request.setAttribute(key,value);

request.setAttribute(“doorName”,doorName)

//把请求转发到order.jsp页面，把参数对应值都传递给jsp页面

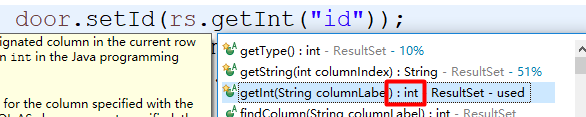
request.getRequestDispatcher(“/WEB-INF/views/Order.jsp”, ).forward(request,response);

Jsp负责页面的展现

展示html标签，tomcat直接翻译，静态页面

动态的信息doorName，在jsp中如何获取呢（el表达式）

${doorName}内部回去request找，key=”doorName”，如果找到把value值进行返回。如果没找，就返回空，在页面上就什么也不展示。



Rs.getInt（“id”）

getInt而不叫getInteger

在于作者的目的。返回int

在java中两类变量：

1. 基础变量int,double,long,char,boolean
2. 对象变量 Integer,Double,Long,String,Boolean

最大的差异：基础变量都有默认值，而对象变量的默认值都是null

int=0;double=0.0

需求：考试，一个学员考试，有成绩。

int =0，（int i=-1）

Integer =null

基础变量存储的空间小，对象变量存储空间大！内存

Java中尽量使用对象变量（面向对象编程）

如果是在做通信程序，推荐使用简单变量

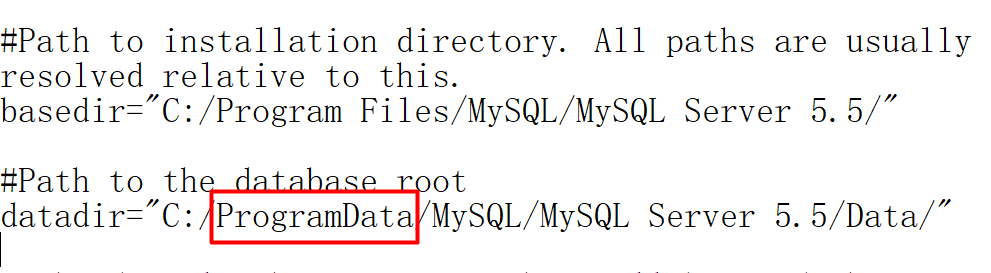
Mysql数据库实质文件类型。

早期的mysql是直接和mysql/test默认的数据库放在一起的。

现在版本换地方，从一个文件中看到，全局mysql配置文件mysql.ini（ini微软习惯配置信息的一个文件）

Mysql提供数据库所在位置

C:\Program Files\MySQL\MySQL Server 5.5\data\mysql



这是一个隐藏目录！

数据库包括表其实mysql本质文件，它把这些信息组织后，存储成二进制文件。Mysql数据库自己来维护。

String doorName = “大钟寺门店”; 单独进行数据库访问写成一个方法（函数）getDoorName（jdbc，返回值string）

String tel = “110”; 单独进行数据库访问，把上面的方法拷贝，然后修改获取值的地方

缺点：

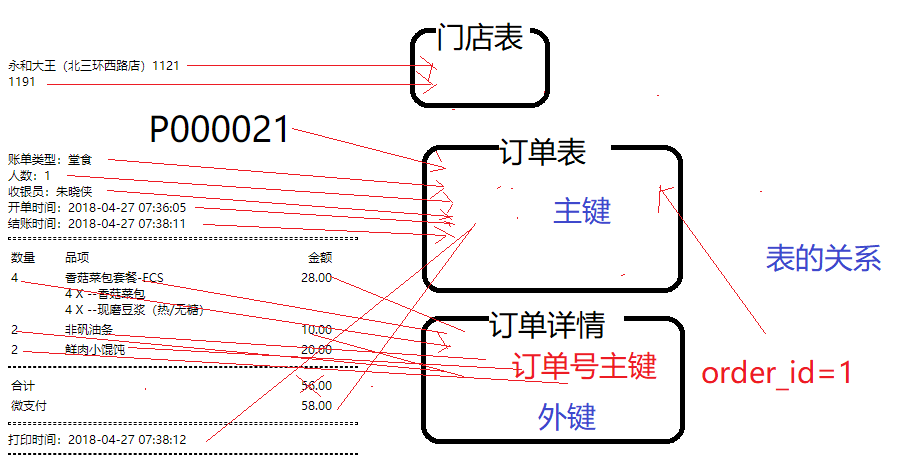
1. 大量重复代码
2. 数据库进行多次链接

解决方案：

1. pojo对象，因为这个做后，可以支持返回多个值
2. 查询数据库就合成了一个方法（一次查询），
3. 返回值就变成pojo，代码怎么设置pojo？ Door door = new Door();利用pojo对象每个属性的set方法去设置即可
4. 需要属性怎么获取，就利用pojo的get方法获取。

找出需求中的动态数据，这些数据就需要放在数据库中，方便维护！





特殊的表名，列名

SELECT \* FROM `order`;

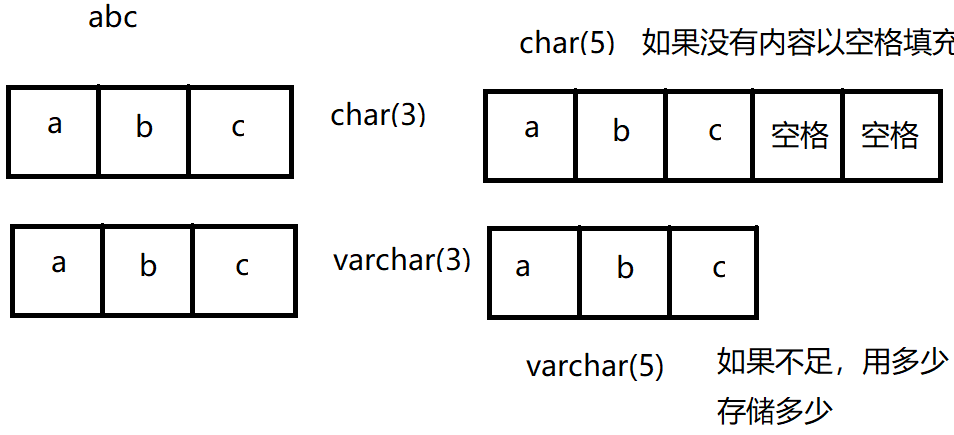
''

` 刀秋

创建表的主键

尽量的和业务无关，如果业务唯一性不能保证，我们的表丝毫不受影响！

Char和varchar有什么区别？



Char固定长度，查询的性能高，但是不宜太长，会浪费我们的存储的空间

Varcar变长，长度变化，查询的性能低，存储不会浪费空间

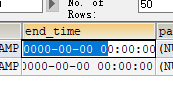
Fk 外键 forgein key

表设计

如何从用户原始需求中提炼数据库表？

1. 原始界面中逐字分析
2. 把信息分成几类：静态内容和动态的内容。把静态内容就排除在外。
3. 怎么把信息拆到不同的数据库表中
   1. 从上向下分析，如果遇到动态内容，第一遇到先创建一个表
   2. “店名”这个信息是描述什么的，一般就会以关键信息点作为表名tb\_door
   3. 遇到“电话”，首先判断是否属于当前这个表的内容，电话就是属于描述门店信息的，所以就成为这个表的字段。
   4. 遇到“账单号”，也是判断是否属于当前这个表，不属于描述门店的信息，创建新的表。创建订单表。tb\_order
   5. “账单类型”看它从属我们哪个表，明显属于订单表
   6. “订单数量”判断是否从事现有的表，它描述的和上面的表的分类都不相同，又创建新表tb\_order\_detail。如果感觉它也可以属于订单表，可以先放放试试，如果把这个内容放入到订单，其它字段内容，而且大多的字段，内容都重复。如果其中一个修改，这多条数据都有随之修改。如果发生这样的事情，表就设计的有问题，不能大量重复，就把信息拆开，形成了新表。
   7. 合计不能放入，要求：如果这个字段的内容可以由其它字段加工而成，这样的内容不能放入到数据库中。合计=数量\*单价，累加
   8. “微支付”要同时多看几张订单，就可以拆除，这个内容含有两个信息，只是表达的比较隐晦。它的值有：微支付和支付宝支付，现今支付。拆成两个字段：支付类型，和支付金额。
   9. “打印时间”可以放入数据库表（后期对打印时间做统计），也可以不放入（不去统计时，不用放入表）。





Mysql+sqlyog在处理日期时不够严谨，有默认值0000-00-00 00:00:00这样的格式值直接放在字段中，如果查询数据库时含有这样的内容，将来java中就会报错。它不是一个正确日期格式。

完成这个订单信息

--查询某个订单所对应门店信息

SELECT \* FROM tb\_door WHERE id=1;

带3个字段返回，Door pojo

--查询某个订单的信息

SELECT \* FROM tb\_order WHERE id=1;

返回n个字段，Order pojo

--查询某个订单下订单详情信息

SELECT \* FROM tb\_order\_detail

WHERE order\_id=1;

返回3个字段和3条记录，

数组

解决一条OrderDetail pojo

List<OrderDetail> 数组<泛型>，泛型是告诉我们数组里每个元素的值的类型是谁

Jsp最终获取数据不想去3个对象中获取，就像直接从一个对象中取获取

Order有两个特殊属性标识对象之间关联

对一：门店 Door door；

对多：订单详情List<OrderDetail>

在jsp页面中遍历集合，使用JSTL java支持标签库

开发步骤;

1. 引入jar

/yh/WebContent/WEB-INF/lib/jstl-1.2.jar

1. Jsp中先引入taglib

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

1. 在Jsp引入 c:forEach

<c:forEach items="${order.orderDetails}" var="o">

Items填写request集合元素，必须el写法

var起每行别名

日期处理

1. 修改pojo返回参数类型String printTime;

rs.getDate("print\_time") 只返回日期部分

rs.getTimestamp("print\_time") 返回时间戳，会多毫秒数

利用格式化函数SimpleDateFormat函数

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

order.setPrintTime(sdf.format(rs.getTimestamp("print\_time")));

合计

<c:set var="total" value="0"/> 设置变量，变量名total，设置一个初始值为0

相当于java中的while，遍历一个集合List集合，var设定一个别名，访问方便

<c:forEach items="${order.orderDetails}" var="o">

设置一个total进行累加

<c:set var="total" value="${total+o.price}"/>

<tr>

<td valign="top">${o.num}</td> 直接利用别名来获取属性值

<td>${o.item}</td>

<td valign="top" align="right">${o.price}</td>

</tr>

</c:forEach>

<td width="40" align="right">${total}</td> 合计就是el表达式

小结：

1. 和数据库进行查询，查询3次，查询出某个订单，通过这个订单获取door\_id，通过door\_id获取它门店对象信息，然后在通过订单orderId去找所属的订单详情的一个集合List集合。
2. 数据库查询后的返回值

Door door = new Door();

door.setDoorName(rs.getString(“door\_name”));

List<OrderDetail> odList = new ArrayList<OrderDetail>();

while(rs.next()){

OrderDetail orderDetail = new OrderDetail();

orderDetail.setNum (rs.getInt(“num”));

。。。。

odList.add(orderDetail);

}

return odList;

Order order = new Order();

while(rs.next()){

order.orderNo(rs.getString(“order\_no”));

}

//关联/绑定，订单和门店关系

Door door = getDoor(order.doorId); 从数据库查询出当前订单对应门店

order.setDoor(door);

//关联/绑定，订单和订单详情的关系

List<OrderDetails> odList = getOrderDetails(order.id);

order.setOrderDetails(odList);

pojo

Door

private Integer id;

private String doorName;

private String tel;

OrderDetail

private Integer id;

private Integer num;

private String item;

private Double price;

//最复杂，多一个对象关联关系

Order

//一个订单对应一个门店，对一

private Door door;

//一个订单对应多个订单详情，对多

List<OrderDetail> orderDetails;

private Integer id;

private String orderNo;

….