一、报表

* 报表的定义

1、以格式化的形式输出数据，对数据进行分组、汇总、计算等操作，通过报表、图表、

或嵌入图片图像等形式来丰富数据的显示。

* 无框架报表生成：

1、实现报表输出的准备工作。

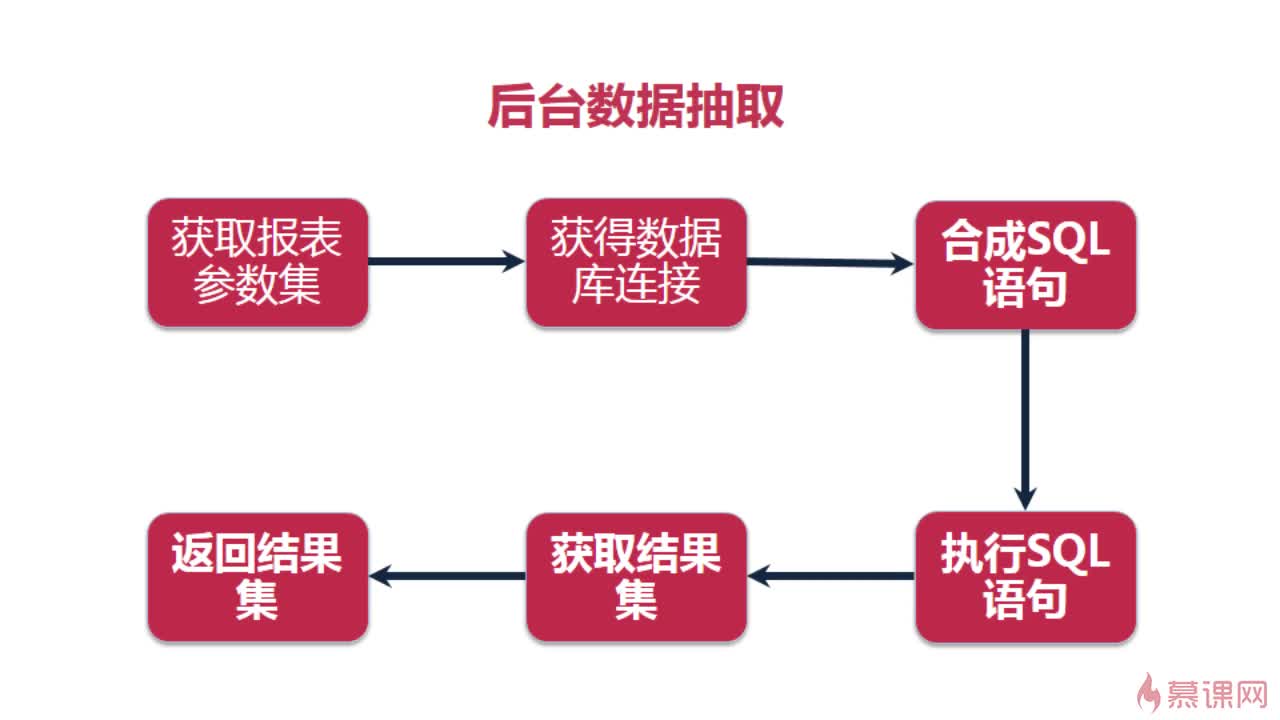
2、报表数据的组织（数据库语言层面）。

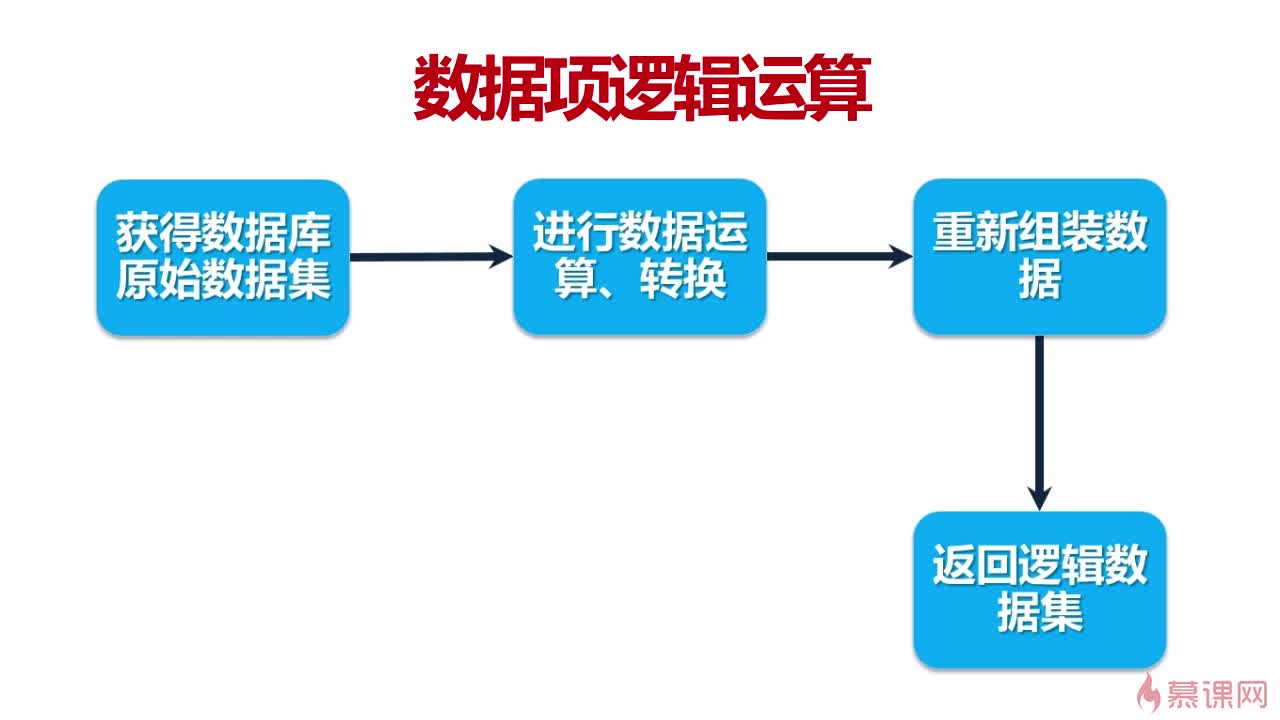
3、原始数据的组装整理。

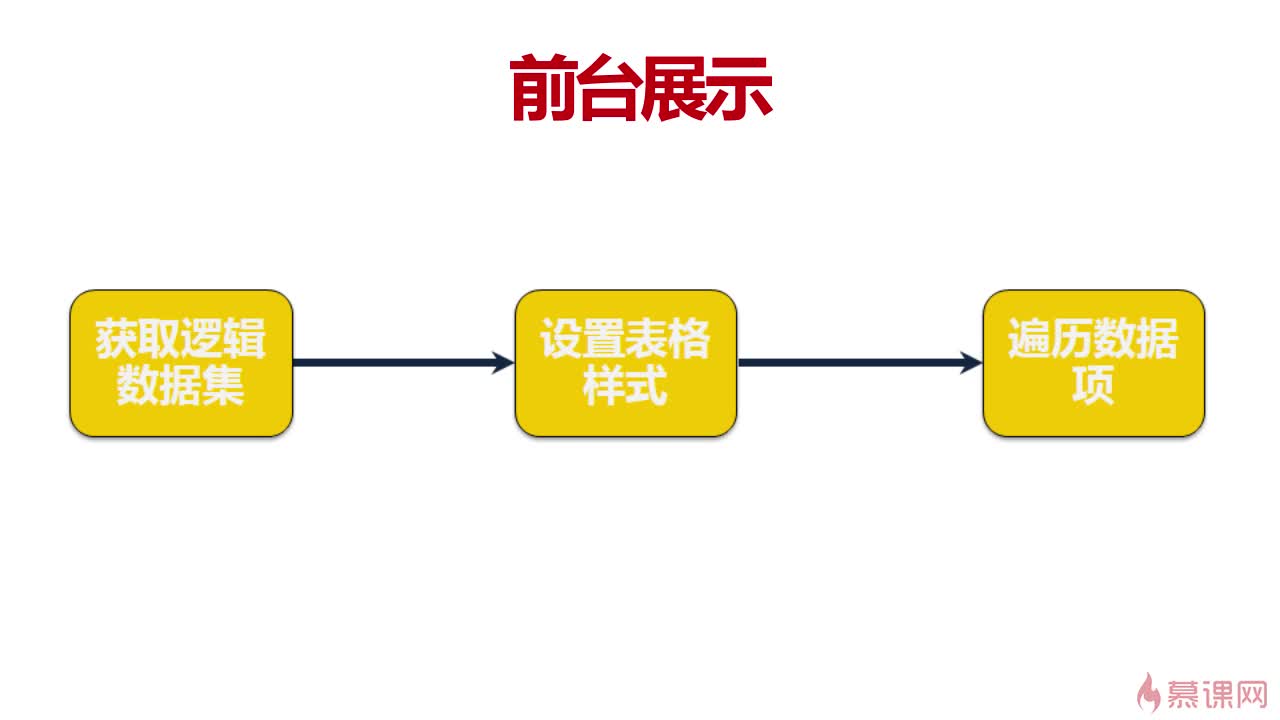
4、报表的前台表格展现。

* 报表生成的关键要素：

1、后台数据抽取 2、数据项逻辑运算 3、前台表格展现







二、案例

* 应用框架

1、Model层：

实体类entity、连接类JDBC

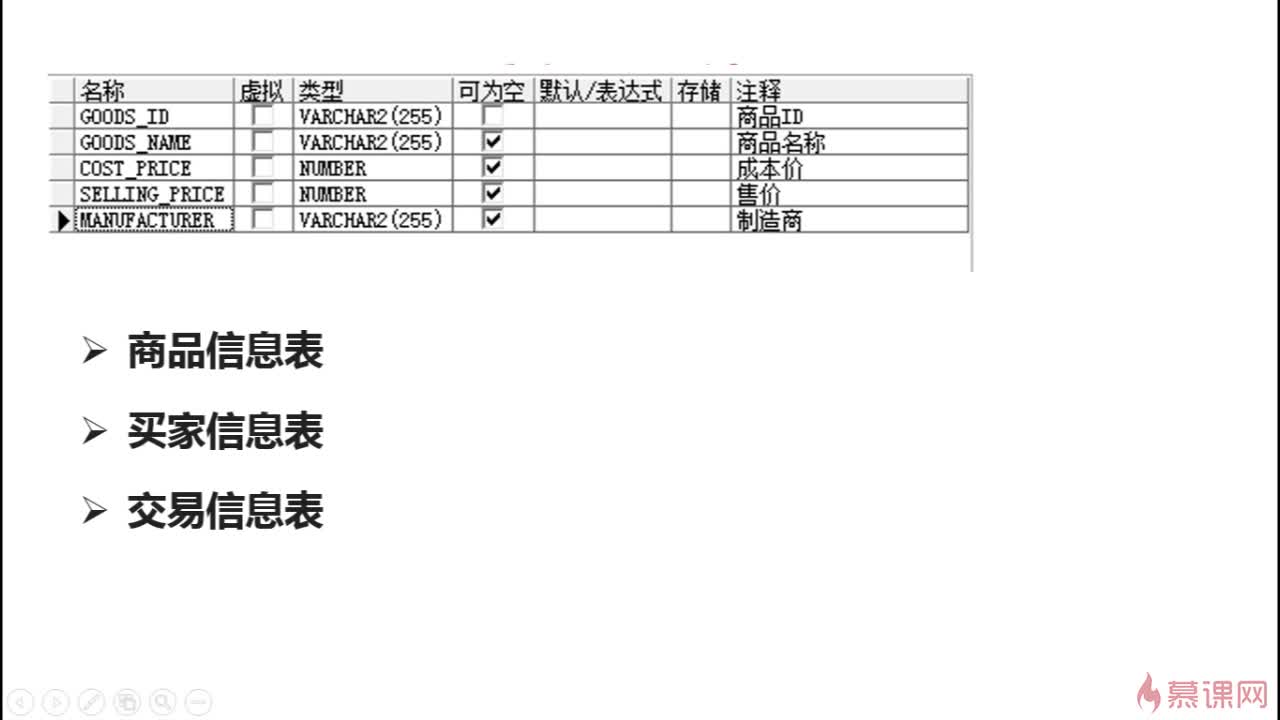
2、Control层：

逻辑类service、控制处理类servlet

3、View层：

前台JSP

* 需求分析：







* Java用JDBC连接数据库的时候部署驱动包方法：

1.选中项目，然后选择Build Path，选择Configure Build Path；

2.进入配置之后，点击Libraries，点击Add External JARs;

3.找到驱动包，点击打开，添加加入之后，点击OK即可。

* JDBC数据库链接代码

package com.util;

import java.sql.\*;

public class DBHelper {

private static final String driver="com.mysql.jdbc.Driver";

private static final String url="jdbc:mysql://localhost:3306/shopping?useUnicode=true&characterEncoding=UTF-8";

private static final String username="root";

private static final String password="root";

private static Connection conn = null;

//静态代码块负责加载驱动

static{

try {

Class.forName(driver);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

//返回数据库对象

public static Connection getConnection() throws SQLException{

if(conn == null){

conn = DriverManager.getConnection(url , username ,password);

return conn;

}

return conn;

}

//测试程序

public static void main(String args[]){

try {

Connection conn = DBHelper.getConnection();

if(conn != null){

System.out.println("数据库链接成功！");

}else {

System.out.println("数据库链接失败！");

}

} catch (SQLException e) {

e.printStackTrace();

}

}}

* 执行SQL语句代码

public ArrayList<Items> getAllItems() {

Connection conn = null;

PreparedStatement stmt = null;

ResultSet rs = null;

ArrayList<Items> list = new ArrayList<Items>();

try {

conn = DBHelper.getConnection();

String sql = "SELECT \* FROM items;";

stmt = conn.prepareStatement(sql);

rs = stmt.executeQuery();

while (rs.next()) {

Items item = new Items();

item.setId(rs.getInt("id"));

…….

item.setNumber(rs.getInt("number"));

item.setPicture(rs.getString("picture"));

list.add(item);// 添加item对象到集合list

}

return list;

} catch (Exception e) {

e.printStackTrace();

return null;

} finally {

// 释放结果集

if (rs != null) {

try {

rs.close();

rs = null;

} catch (Exception e) {

e.printStackTrace();

}

}

// 释放语句

if (stmt != null) {

try {

stmt.close();

stmt = null;

} catch (Exception e) {

e.printStackTrace();

}

}

}}