



# 面向对象与多线程综合实验

结合华为云平台的基于JAVASE数  
据挖掘系统设计与实现



# 实验目标

---

1. 熟悉并理解系统的数据库设计和结构
2. 完成系统数据记录的控制台功能实现，借此更深入掌握JDBC的相关开发能力，以及系统的调试能力



# 实验要求与交付

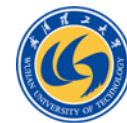
---

## 要求实现：

1. 实现并测试匹配的日志记录的数据库保存和查询功能
2. 嵌入到系统主界面，实现系统控制台的数据库数据记录和显示功能。
3. 系统优化（保证数据库中的记录不能有重复）

## ★交付：

1. 代码与运行结果
2. 开发过程中遇到问题截图和解决办法
3. 回答问题：你认为该系统的数据库表设计与结构有哪些优缺点？



# 实验内容

## 1. 做什么？

### ◆ 控制台数据记录功能

```
*****
* 1、数据采集      2、数据匹配 *
* 3、数据记录      4、数据显示 *
* 5、数据发送      0、退出应用 *
*****
```

请输入菜单项（0~5）：

分别将日志和物流记录  
保存到文件和MySQL数据库  
中并从文件和数据库中  
显示到控制台

## 2. 怎么做？

### ◆ 1. 搭建数据库基础环境

### ◆ 2. JDBC及系统框架

### ◆ 3. 系统功能实现

- 3.1 匹配日志数据保存到数据库
- 3.2 从数据库中读取匹配的日志数据
- 3.3 测试数据库存取功能
- 3.4 嵌入到控制台主界面

## 3. 对应项

- ◆ MySQL与SQLyog安装配置
- ◆ MySQL数据库操作

- db-》DBUtil.java
- util-》Config.java
- config/mysql.properties
- libs/mysql-connector

- saveMatchLogToDB.java
- readMatchedLogFromDB.java
- DBDemo.java
- MenuDriver.java



# 实验内容

## 实现数据记录和保存功能

记录保存到文本文件



文件与输入输出流



串行化实现数据文件存取  
嵌入到系统框架中实现

记录保存到数据库



JDBC数据库编程

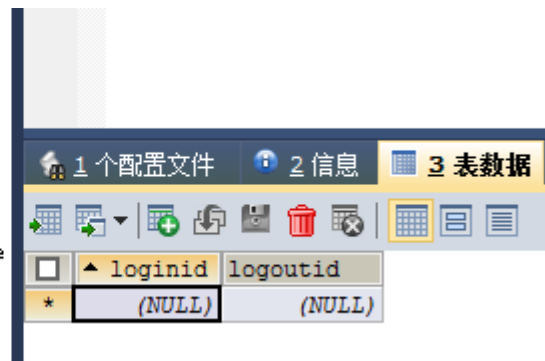
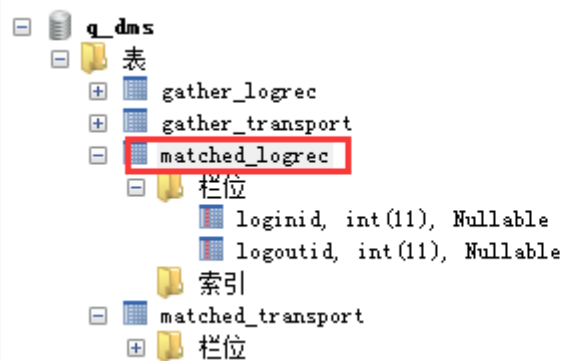
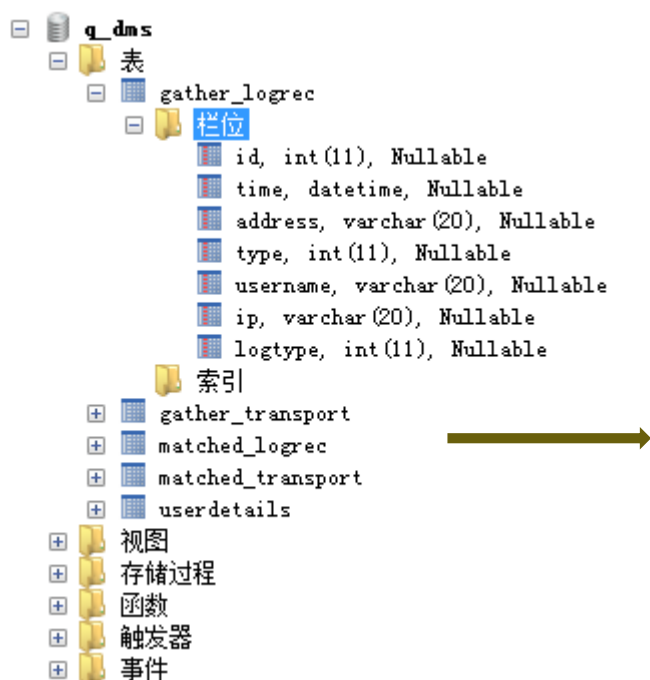


1. MySQL和前端管理工具安装
2. JDBC实现数据在数据库中存取
3. 系统功能实现



# 一、数据库设计

## ■ q\_dms数据库





# 匹配的日志记录保存到数据库

- Service包下LogRecService类中新增saveMatchLogToDB方法
- 保存到数据库q\_dms的gather\_logrec表和matched\_logrec表
- 分别将登入和登出日志保存到gather\_logrec表
- 将匹配登录登出日志的id索引存到matched\_logrec表
- 参考物流匹配记录保存

```
// 匹配日志信息保存到数据库，参数是集合
public void saveMatchLogToDB(ArrayList<MatchedLogRec> matchLogs) {
    DBUtil db = new DBUtil();
    try {
        // 获取数据库链接

        // 获取匹配的登录日志

        // 获取匹配的登出日志

        // 保存匹配记录中的登录日志

        // 保存匹配记录中的登出日志

        // 保存匹配日志的ID

        // 关闭数据库连接，释放资源

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



# 匹配的物流记录保存到数据库

```
// 匹配物流信息保存到数据库, 参数是集合
public void saveMatchTransportToDB(ArrayList<MatchedTransport> matchTrans) {
    DBUtil db = new DBUtil();
    try {
        // 获取数据库连接
        db.getConnection();
        for (MatchedTransport matchedTransport : matchTrans) {
            // 获取匹配的发送物流
            Transport send = matchedTransport.getSend();
            // 获取匹配的运输物流
            Transport trans = matchedTransport.getTrans();
            // 获取匹配的接收物流
            Transport receive = matchedTransport.getReceive();
            // 保存匹配记录中的发送状态
            String sql = "INSERT INTO gather_transport(id,time,address,type,handler,reciver,transporttype) VALUES(?,?,?, ?, ?, ?, ?)";
            Object[] param = new Object[] { send.getId(),
                new Timestamp(send.getTime().getTime()),
                send.getAddress(), send.getType(), send.getHandler(),
                send.getReciver(), send.getTransportType() };
            db.executeUpdate(sql, param);
            // 保存匹配记录中的运输状态
            param = new Object[] { trans.getId(),
                new Timestamp(trans.getTime().getTime()),
                trans.getAddress(), trans.getType(),
                trans.getHandler(), trans.getReciver(),
                trans.getTransportType() };
            db.executeUpdate(sql, param);
            // 保存匹配记录中的接收状态
            param = new Object[] { receive.getId(),
                new Timestamp(receive.getTime().getTime()),
                receive.getAddress(), receive.getType(),
                receive.getHandler(), receive.getReciver(),
                receive.getTransportType() };
            db.executeUpdate(sql, param);
            // 保存匹配日志的ID
            sql = "INSERT INTO matched_transport(sendid,transid,receiveid) VALUES(?,?,?)";
            param = new Object[] { send.getId(), trans.getId(),
                receive.getId() };
            db.executeUpdate(sql, param);
        }
        // 关闭数据库连接, 释放资源
        db.closeAll();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```





# 从数据库中读取匹配的日志记录

- Service包下LogRecService类中新增readMatchedLogFromDB方法
- 利用多表联合查询得到包含登录和登出记录的ResultSet结果集
- 从结果集对象中取出登录登出记录，并添加到匹配日志集合
- 参考物流匹配记录读取

```
// 从数据库读匹配日志信息，返回匹配日志信息集合
public ArrayList<MatchedLogRec> readMatchedLogFromDB() {
    ArrayList<MatchedLogRec> matchedLogRecs = new ArrayList<MatchedLogRec>();
    DBUtil db = new DBUtil();
    try {
        // 获取数据库连接

        // 查询匹配的日志，返回结果集ResultSet

        // 获取登录记录

        // 获取登出记录

        // 添加匹配登录信息到匹配集合

        // 关闭数据库连接，释放资源
    } catch (Exception e) {
        e.printStackTrace();
    }
    // 返回匹配日志信息集合
}
```



# 从数据库中读取匹配的物流记录

```
// 从数据库中读匹配物流信息，返回匹配物流信息集合
public ArrayList<MatchedTransport> readMatchedTransportFromDB() {
    ArrayList<MatchedTransport> matchedTransports = new ArrayList<MatchedTransport>();
    DBUtil db = new DBUtil();
    try {
        // 获取数据库链接
        db.getConnection();
        // 查询匹配的物流
        String sql = "SELECT s.ID,s.TIME,s.ADDRESS,s.TYPE,s.HANDLER,s.RECIVER,s.TRANSPORTTYPE,"
            + "t.ID,t.TIME,t.ADDRESS,t.TYPE,t.HANDLER,t.RECIVER,t.TRANSPORTTYPE,"
            + "r.ID,r.TIME,r.ADDRESS,r.TYPE,r.HANDLER,r.RECIVER,r.TRANSPORTTYPE "
            + "FROM MATCHED_TRANSPORT m,GATHER_TRANSPORT s,GATHER_TRANSPORT t,GATHER_TRANSPORT r "
            + "WHERE m.SENDID=s.ID AND m.TRANSID=t.ID AND m.RECEIVEID=r.ID";
        ResultSet rs = db.executeQuery(sql, null);
        while (rs.next()) {
            // 获取发送记录
            Transport send = new Transport(rs.getInt(1), rs.getDate(2),
                rs.getString(3), rs.getInt(4), rs.getString(5),
                rs.getString(6), rs.getInt(7));
            // 获取运输记录
            Transport trans = new Transport(rs.getInt(8), rs.getDate(9),
                rs.getString(10), rs.getInt(11), rs.getString(12),
                rs.getString(13), rs.getInt(14));
            // 获取接收记录
            Transport receive = new Transport(rs.getInt(15),
                rs.getDate(16), rs.getString(17), rs.getInt(18),
                rs.getString(19), rs.getString(20), rs.getInt(21));
            // 添加匹配登录信息到匹配集合
            MatchedTransport matchedTrans = new MatchedTransport(send,
                trans, receive);
            matchedTransports.add(matchedTrans);
        }
        // 关闭数据库连接，释放资源
        db.closeAll();
    } catch (Exception e) {
        e.printStackTrace();
    }
    // 返回匹配物流信息集合
    return matchedTransports;
}
```



## 二、测试数据库存储功能

### ■ 运行Doc包下DBDemo. java来测试日志和物流记录的数据库存取功能

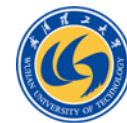
com.qst.dms.dos  
DBDemo.java

```
package com.qst.dms.dos;

import java.util.ArrayList;

public class DBDemo {
    public static void main(String[] args) {
        // 创建一个日志业务类
        LogRecService logService = new LogRecService();
        ArrayList<MatchedLogRec> matchLogs = new ArrayList<>();
        matchLogs.add(new MatchedLogRec(
            new LogRec(1001, new Date(), "青岛", DataBase.GATHER, "zhangsan", "192.168.1.1", 1),
            new LogRec(1002, new Date(), "青岛", DataBase.GATHER, "zhangsan", "192.168.1.1", 0)));
        matchLogs.add(new MatchedLogRec(
            new LogRec(1003, new Date(), "北京", DataBase.GATHER, "lisi", "192.168.1.6", 1),
            new LogRec(1004, new Date(), "北京", DataBase.GATHER, "lisi", "192.168.1.6", 0)));
        matchLogs.add(new MatchedLogRec(
            new LogRec(1005, new Date(), "济南", DataBase.GATHER, "wangwu", "192.168.1.89", 1),
            new LogRec(1006, new Date(), "济南", DataBase.GATHER, "wangwu", "192.168.1.89", 0)));
        //保存匹配的日志信息到数据库中
        logService.saveMatchLogToDB(matchLogs);
        //从数据库中读取匹配的日志信息
        ArrayList<MatchedLogRec> logList = logService.readMatchedLogFromDB();
        logService.showMatchLog(logList);

        // 创建一个物流业务类
        TransportService tranService = new TransportService();
        ArrayList<MatchedTransport> matchTrans = new ArrayList<>();
        matchTrans.add(new MatchedTransport(
            new Transport(2001, new Date(), "青岛", DataBase.GATHER, "zhangsan", "zhaokel", 1),
            new Transport(2002, new Date(), "北京", DataBase.GATHER, "lisi", "zhaokel", 2),
            new Transport(2003, new Date(), "北京", DataBase.GATHER, "wangwu", "zhaokel", 3)));
        matchTrans.add(new MatchedTransport(
            new Transport(2004, new Date(), "青岛", DataBase.GATHER, "maliu", "zhaokel", 1),
            new Transport(2005, new Date(), "北京", DataBase.GATHER, "sunqi", "zhaokel", 2),
            new Transport(2006, new Date(), "北京", DataBase.GATHER, "fengba", "zhaokel", 3)));
        //保存匹配的物流信息到数据库中
        tranService.saveMatchTransportToDB(matchTrans);
        //从数据库中读取匹配的物流信息
        ArrayList<MatchedTransport> transportList = tranService.readMatchedTransportFromDB();
        tranService.showMatchTransport(transportList);
    }
}
```



## 三、嵌入主界面调用

在控制台主界面MenuDriver.java中

在case3和case4中进行方法的调用

分别将日志和物流记录保存到文件和MySQL数据库中并从文件和数据库中显示到控制台

```
*****  
* 1、数据采集          2、数据匹配 *  
* 3、数据记录          4、数据显示 *  
* 5、数据发送          0、退出应用 *  
*****
```

请输入菜单项 (0~5):

```
case 3:  
    System.out.println("数据记录 中...");  
  
    //第二次课将匹配后的日志和物流数据存储到文件  
  
    //第三次课再加上将匹配后的日志和物流数据存储到MySQL数据库  
    //...  
  
    break;  
  
case 4: {  
    System.out.println("显示匹配的数据: ");  
  
    //第二次课显示从文件中读取的物流和日志数据  
  
    //第三次课再加上显示从数据库中读取的物流和日志数据  
    /*
```



## 三、结果展示

详见实验平台视频：第三次实验版本演示.mp4

基本信息	课程内容	教学大纲	知识点	题库	练习	作业	考试	学生名单	评论	返回	新增		
第1章 数据挖掘系统框架搭建和控制台基本功能实现										编辑	删除	隐藏小节	增加小节
第1节 数据挖掘系统框架搭建和控制台基本功能实现													
第2章 基于文件I/O流的系统控制台数据记录功能实现										编辑	删除	隐藏小节	增加小节
第1节 基于文件I/O流的系统控制台数据记录功能实现													
第3章 基于JDBC的系统控制台数据记录功能实现										编辑	删除	隐藏小节	增加小节
第1节 MySQL及管理工具的安装和数据库操作													
第2节 JDBC实现数据在MySQL数据库中存取													
第3节 将数据库操作类嵌入系统框架中并完成主界面调用													

序号	类型	标题	排序号	可下载	编辑	删除
1	文件	面向对象与多线程综合实验第三次课2020-03.pdf	1	<input checked="" type="checkbox"/>		
2	视频	第三次实验版本演示.mp4	2	<input checked="" type="checkbox"/>		

修改

取消