

## 4.25

- P7
  - HTML改jsp文件: 1: 26
  - 修改连接路径: 7: 20
  - 修改页面填充信息字段: 11: 04
  - 16: 45
  - 给sql语句占位符赋值: 17: 16
  - 添加成功或失败后的重定向: 25: 00

## 4.26

- P8
  - 封装数据库增删改操作：16：00  
传入要在数据库中添加的信息的参数示例：26：15
  - 数据库新增数据时时间的格式化：24：40
- P9
  - 创建用于用户查询的servlet：4：00
  - 将获取到的用户列表放到请求对象里面，然后再重定向到用户列表（7：00）：6：00
- P10
  - 遍历用户记录（需要引入标签库：2：30，3：20）：1：47
- P11
  - 查询记录分页展示中jsp代码编写：0：44 和 13：40
  - 查询记录分页展示中service层代码编写：6：11
  - 查询记录分页展示中servlet代码编写：12：31
  - 根据分页来获取每一页的数据：15：27  
MySQL分页查询语句：select \* from 表名 where 字段限制 limit 1000(查询起点),10(查询数量);
  - 首页，尾页，上一页，下一页跳转：18：40
  - 注意：数据加载时要跳转至servlet，不要跳转至jsp页面
- P12
  - 用户搜索结果分页显示：00：40
  - 将变量值赋给占位符：setString (1, "%" + 变量名 + "%")
  - 用户搜索关键字停留在搜索栏：10：00
- P13, P14：用户修改操作
  - jsp文件中“修改”标签的编写：3：45
  - servlet中“修改”的编写：5：00
  - 插入SQL语句中日期格式处理：21：14
  - 提交修改信息时把当前所在分页是第几页也一块提交；（P14）7：00
  - 根据修改结果重定向页面：（P14）6：14
  - 更新SQL语句中日期格式处理：10：30

## 4.28

- P15: 用户的删除
  - 删除后若要留在当前页面, 要把当前页码传到servlet中
  - 点击删除连接时通过get提交删除信息: 1: 00
  - jsp中判断用户身份来选择是否显示用户删除连接: 3: 30
  - jsp中弹窗显示确认深处信息: 4: 18
- P16: 用户注册界面
  - 用户注册输入注册信息: 10: 15
  - 使用calendar.js输入日期: 12: 20
- P18: 用户输入验证
  - jsp上验证用户输入不能为空: 1: 00
- P19: 使用Ajax验证用户名是否被注册
  - js中使用Ajax判断用户输入信息是否为空以及用户名是否被注册: 2: 20
  - 判断用户输入验证码是否正确: 20: 50
- P20: 验证用户信息不通过时禁止注册操作
  - 修改注册jsp: 0: 35
  - 修改js: 2: 10
  - 添加标记位使添加新信息不符合要求时禁止注册操作: 7: 10
- P23: 登录界面
  - jsp文件代码编写: 0: 30
  - servlet中根据用户登录信息判断登录结果以及接下来的动作: 6: 45
  - 登录成功后将 用户信息以及登录状态 装进session中: 11: 42
- P24: 前台信息展示
  - 将indx.jsp中重复的菜单栏部分提取到header.jsp中: 1: 30
  - 在index.jsp中添加标签: 3: 50
  - 处理登录与注册按钮, 若用户已登录, 则这两个按钮不显示: 4: 50
  - 往个人信息.jsp 文件中添加标签库: 9: 15
  - 退出登录servlet的编写: 12: 42
  - 如果访问的页面需要从数据库中获取信息, 一定要访问servlet
- P25: 后台管理员登录与退出
  - 后台登录servlet: 5: 50
  - 退出servlet: 13: 16

## 4.29

- P26: 管理后台权限验证
  - 添加过滤器: 1: 50
  - 排除对登录界面的拦截: 8: 00
  - 后台需要登录才可以操作的jsp文件在命名时注意统一格式, 方便后面对其进行过滤
- md5加密

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6     <script type="text/javascript" src="./md5-min.js"></script>
7     <script type="text/javascript" src="sha1-min.js"></script>
8     <script type="text/javascript">
9         /**
10             md5      =>  21232f297a57a5a743894a0e4a801fc3
11             sha1     =>  d033e22ae348aeb5660fc2140aec35850c4da997
12         */
13         console.log("md5      =>  "+hex_md5("admin"))
14         console.log("sha1     =>  "+hex_sha1("admin"))
```