# 实验7 综合实践

## 7.1实验目的

（1）掌握基于B/S架构的数据库管理系统设计与开发方法。  
（2）熟练应用触发器、存储过程等数据库高级技术实现业务逻辑。  
（3）培养团队协作能力及系统文档编写与项目管理实践技能。

## 7.2实验平台

（1）操作系统：Windows；

（2）编程语言：Java；

（3）脚本语言：JavaScript/Java；

（4）数据库：MongoDB；

（5）框架：SpringBoot

## 7.3 实验任务与基本要求

### 7.3.1 实验内容

设计并实现一个基于B/S或C/S架构的数据库管理系统。DBMS可以从SQL Server、Oracle或MySQL中做选择。开发的管理系统应包含以下功能：

(1) 每个模块必须实现检索、新增、编辑、删除功能；特定模块需支持批量删除操作。

(2) 系统应包含权限控制功能：权限配置需具备动态性，程序应内置用户管理及用户权限管理模块，以便灵活管理用户对各个模块、子模块或具体功能的访问权限。

(3) 至少应运用触发器或存储过程中的一种技术，以增强系统的功能性和效率。

(4) 界面设计应满足基本的人机交互标准，力求简洁易用，并具有一定的美观度。

(5) 数据库中的每张表都必须设置主键，以确保数据的唯一性和完整性。

### 7.3.2. 实验题目

**邦多利大酒店管理系统**

酒店管理系统整合了与酒店入住信息相关的各项管理功能。该系统为两类用户——客户和系统管理员提供服务。每类用户模块包含以下核心功能：

数据库应至少包含以下表格：

1. 入住登记表
2. 入住服务关联表
3. 客户信息表
4. 会员信息表
5. 酒店房间表
6. 酒店服务表
7. 酒店服务请求表
8. 管理员模块

* 负责客户信息的添加、移除、更新及查询操作；
* 负责客房信息的添加、移除、更新及查询操作；
* 负责酒店服务的添加、移除、更新及查询操作；
* 负责会员信息的添加、移除、更新及查询操作；
* 管理客户入住信息；
* 客户历史入住信息；
* 监控酒店客户入住信息；

1. 客户模块

* 酒店入住信息填写；
* 酒店服务申请；

### 7.3.3. 代码实现

**1、环境依赖**

**2、表的实现**

**2.1、定义三个类型**

**2.1.1、性别类型**

public enum Gender {  
 *MALE*("男性"),  
 *FEMALE*("女性"),  
 *OTHER*("其他");  
 private final String description;  
 Gender(String description) {  
 this.description = description;  
 }  
 public String getDescription() {  
 return description;  
 }  
}

2.1.2、会员类型

public enum MemberLevel {  
 *NONE*("非会员"),  
 *BRONZE*("铜牌会员"),  
 *SILVER*("银牌会员"),  
 *GOLD*("金牌会员"),  
 *PLATINUM*("白金会员");  
 private final String description;  
 MemberLevel(String description) {  
 this.description = description;  
 }  
 public String getDescription() {  
 return description;  
 }  
}

2.1.3、房间类型

public enum RoomType {  
 *SINGLE*("单人间"),  
 *DOUBLE*("双人间"),  
 *SUITE*("套房"),  
 *DELUXE*("豪华间");  
 private final String description;  
 RoomType(String description) {  
 this.description = description;  
 }  
 public String getDescription() {  
 return description;  
 }  
}

**2.2、定义入住表**

@Document(collection = "check\_ins") // 创建check\_ins表

public class CheckIn {

@Id // 标识此字段为表的主键

private String id; // 入住登记ID

// 客户信息 (包含冗余字段避免频繁联表查询)

private String customerId; //客户ID

private String customerName; // 客户姓名

private String customerIdCardNumber; // 客户身份证号

// 房间信息 (包含冗余字段)

private String roomId; //客房ID

private String roomNumber; // 客房号

// 入住时间信息

private LocalDate checkInDate; // 实际入住日期

private LocalDate expectedCheckOutDate; // 预计离店日期

private LocalDate actualCheckOutDate; // 实际离店日期

// 服务与费用信息

private List<CheckInServiceItem> services; // 入住期间消费的服务项列表

private double roomCost; // 房间费用

private double servicesCost; // 服务总费用

private double totalAmount; // 总费用

// 状态标识

private boolean isActive; // 是否当前有效入住

初始化触发器

public CheckIn() {

this.services = new ArrayList<>(); // 初始化空的服务列表

this.checkInDate = LocalDate.now(); // 默认入住日期为当前日期

this.isActive = true; // 默认状态为有效入住

}

//创建入住记录

public CheckIn(String customerId, String customerName, String customerIdCardNumber,

String roomId, String roomNumber, LocalDate expectedCheckOutDate,

double roomPricePerNight) {

this();

this.customerId = customerId;

this.customerName = customerName;

this.customerIdCardNumber = customerIdCardNumber;

this.roomId = roomId;

this.roomNumber = roomNumber;

this.expectedCheckOutDate = expectedCheckOutDate;

this.roomCost = roomPricePerNight; // 暂时存储房费

calculateTotalAmount(); // 计算初始总金额

}

//总房费求和

public void calculateTotalAmount() {

// 计算服务总费用：遍历所有服务项并求和

this.servicesCost = services.stream()

.mapToDouble(CheckInServiceItem::getSubtotal)

.sum();

// 注意：实际项目中房费需要根据入住天数动态计算

// 这里简化处理，假设 roomCost 已是总房费

this.totalAmount = this.roomCost + this.servicesCost;

}

//将输入内容记录入入住表中

public String getId() { return id; }

public void setId(String id) { this.id = id; }

public String getCustomerId() { return customerId; }

public void setCustomerId(String customerId) { this.customerId = customerId; }

public String getCustomerName() { return customerName; }

public void setCustomerName(String customerName) { this.customerName = customerName; }

public String getCustomerIdCardNumber() { return customerIdCardNumber; }

public void setCustomerIdCardNumber(String customerIdCardNumber) {

this.customerIdCardNumber = customerIdCardNumber;

}

public String getRoomId() { return roomId; }

public void setRoomId(String roomId) { this.roomId = roomId; }

public String getRoomNumber() { return roomNumber; }

public void setRoomNumber(String roomNumber) { this.roomNumber = roomNumber; }

public LocalDate getCheckInDate() { return checkInDate; }

public void setCheckInDate(LocalDate checkInDate) {

this.checkInDate = checkInDate;

}

public LocalDate getExpectedCheckOutDate() { return expectedCheckOutDate; }

public void setExpectedCheckOutDate(LocalDate expectedCheckOutDate) {

this.expectedCheckOutDate = expectedCheckOutDate;

}

public LocalDate getActualCheckOutDate() { return actualCheckOutDate; }

public void setActualCheckOutDate(LocalDate actualCheckOutDate) {

this.actualCheckOutDate = actualCheckOutDate;

// 实际业务中，设置离店日期时应标记为已退房

if(actualCheckOutDate != null) this.isActive = false;

}

public List<CheckInServiceItem> getServices() { return services; }

// 计算服务费用

public void setServices(List<CheckInServiceItem> services) {

this.services = services;

calculateTotalAmount(); // 更新费用

}

// 添加单个服务项

public void addServiceItem(CheckInServiceItem item) {

this.services.add(item);

calculateTotalAmount(); // 更新费用

}

public double getRoomCost() { return roomCost; }

// 设置房费时自动重新计算总金额

public void setRoomCost(double roomCost) {

this.roomCost = roomCost;

calculateTotalAmount();

}

public double getServicesCost() { return servicesCost; }

//计入服务费

public void setServicesCost(double servicesCost) {

this.servicesCost = servicesCost;

}

public double getTotalAmount() { return totalAmount; }

//计入总金额

public void setTotalAmount(double totalAmount) {

this.totalAmount = totalAmount;

}

public boolean isActive() { return isActive; }

public void setActive(boolean active) {

isActive = active;

// 设置离店日期

if(!active && actualCheckOutDate == null) {

actualCheckOutDate = LocalDate.now();

}

}

// 重写 toString 方法 (用于日志和调试)

@Override

public String toString() {

return "CheckIn{" +

"id='" + id + '\'' +

", customerId='" + customerId + '\'' +

", roomNumber='" + roomNumber + '\'' +

", checkInDate=" + checkInDate +

", totalAmount=" + totalAmount +

", isActive=" + isActive +

'}';

}

}

**2.3、定义入住服务关联表**

// 这个类将作为CheckIn文档的内嵌对象，不单独作为Document

public class CheckInServiceItem {  
 private String serviceInfoId; //服务ID  
 private String serviceName; // 服务名称 (冗余存储，方便显示)  
 private int quantity; // 数量  
 private double unitPrice; // 单价 (冗余存储，防止服务价格变动影响历史记录)  
 private double subtotal; // 小计 (quantity \* unitPrice)  
  
 public CheckInServiceItem() {  
 }  
  
 public CheckInServiceItem(String serviceInfoId, String serviceName, int quantity, double unitPrice) {  
 this.serviceInfoId = serviceInfoId;  
 this.serviceName = serviceName;  
 this.quantity = quantity;  
 this.unitPrice = unitPrice;  
 this.subtotal = quantity \* unitPrice;  
 }  
  
 // Getters and Setters  
 public String getServiceInfoId() {  
 return serviceInfoId;  
 }  
  
 public void setServiceInfoId(String serviceInfoId) {  
 this.serviceInfoId = serviceInfoId;  
 }  
  
 public String getServiceName() {  
 return serviceName;  
 }  
  
 public void setServiceName(String serviceName) {  
 this.serviceName = serviceName;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public void setQuantity(int quantity) {  
 this.quantity = quantity;  
 this.subtotal = quantity \* this.unitPrice; // 更新小计  
 }  
  
 public double getUnitPrice() {  
 return unitPrice;  
 }  
  
 public void setUnitPrice(double unitPrice) {  
 this.unitPrice = unitPrice;  
 this.subtotal = this.quantity \* unitPrice; // 更新小计  
 }  
 public double getSubtotal() {  
 return subtotal;  
 }  
  
 public void setSubtotal(double subtotal) {  
 this.subtotal = subtotal;  
 }  
  
 @Override（用作检查）  
 public String toString() {  
 return "CheckInServiceItem{" +  
 "serviceInfoId='" + serviceInfoId + '\'' +  
 ", serviceName='" + serviceName + '\'' +  
 ", quantity=" + quantity +  
 ", unitPrice=" + unitPrice +  
 ", subtotal=" + subtotal +  
 '}';  
 }  
}

**2.4、定义客户信息表**

@Document(collection = "customers")//创建customers表  
public class Customer {  
 @Id //ID为主键  
 private String id; // MongoDB自动生成的ID  
 private String name; // 姓名  
 private Gender gender; // 性别  
 @Indexed(unique = true) //设定唯一性  
 private String idCardNumber; // 身份证号 (唯一)  
 private String contactInfo; // 联系方式  
 private boolean isMember; // 是否为会员  
 public Customer() {  
 }  
 public Customer(String name, Gender gender, String idCardNumber, String contactInfo) {  
 this.name = name;  
 this.gender = gender;  
 this.idCardNumber = idCardNumber;  
 this.contactInfo = contactInfo;  
 this.isMember = false; // 默认为非会员  
 }  
 // Getters and Setters  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public Gender getGender() {  
 return gender;  
 }  
  
 public void setGender(Gender gender) {  
 this.gender = gender;  
 }  
  
 public String getIdCardNumber() {  
 return idCardNumber;  
 }  
  
 public void setIdCardNumber(String idCardNumber) {  
 this.idCardNumber = idCardNumber;  
 }  
  
 public String getContactInfo() {  
 return contactInfo;  
 }  
  
 public void setContactInfo(String contactInfo) {  
 this.contactInfo = contactInfo;  
 }  
  
 public boolean isMember() {  
 return isMember;  
 }  
  
 public void setMember(boolean member) {  
 isMember= member;  
 }  
  
 @Override（日志调取和检查）  
 public String toString() {  
 return "Customer{" +  
 "id='" + id + '\'' +  
 ", name='" + name + '\'' +  
 ", gender=" + gender +  
 ", idCardNumber='" + idCardNumber + '\'' +  
 ", contactInfo='" + contactInfo + '\'' +  
 ", isMember=" + isMember +  
 '}';  
 }  
}

**2.5、定义会员信息表**

@Document(collection = "member\_info")//创建“member\_info”表  
public class MemberInfo {  
 @Id //设置ID为主键  
 private String id; // 会员ID (MongoDB自动生成)  
  
 @Indexed(unique = true) //设置唯一性  
 private String idCardNumber; // 身份证号 (关联Customer, 确保唯一性)  
  
 private LocalDate registrationDate; // 注册日期  
 private MemberLevel memberLevel; // 会员等级  
  
 public MemberInfo() {  
 }  
  
 public MemberInfo(String idCardNumber, LocalDate registrationDate, MemberLevel memberLevel) {  
 this.idCardNumber = idCardNumber;  
 this.registrationDate = registrationDate;  
 this.memberLevel = memberLevel;  
 }  
  
 // Getters and Setters  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getIdCardNumber() {  
 return idCardNumber;  
 }  
  
 public void setIdCardNumber(String idCardNumber) {  
 this.idCardNumber = idCardNumber;  
 }  
  
 public LocalDate getRegistrationDate() {  
 return registrationDate;  
 }  
  
 public void setRegistrationDate(LocalDate registrationDate) {  
 this.registrationDate = registrationDate;  
 }  
  
 public MemberLevel getMemberLevel() {  
 return memberLevel;  
 }  
  
 public void setMemberLevel(MemberLevel memberLevel) {  
 this.memberLevel = memberLevel;  
 }  
  
 @Override（日志调取和错误检查）  
 public String toString() {  
 return "MemberInfo{" +  
 "id='" + id + '\'' +  
 ", idCardNumber='" + idCardNumber + '\'' +  
 ", registrationDate=" + registrationDate +  
 ", memberLevel=" + memberLevel +  
 '}';  
 }  
}

**2.6、定义酒店房间表**

@Document(collection = "rooms") //创建“rooms”表  
public class Room {  
 @Id //设置主键  
 private String id; // MongoDB自动生成的ID  
 @Indexed(unique = true) //确定唯一性  
 private String roomNumber; // 客房号 (唯一)  
 private double price; // 单价  
 private boolean isOccupied; // 是否入住  
 private RoomType type; // 类型  
 public Room() {}  
 public Room(String roomNumber, double price, RoomType type) {  
 this.roomNumber = roomNumber;  
 this.price = price;  
 this.type = type;  
 this.isOccupied = false; // 新房间默认未入住  
 }  
  
 // Getters and Setters  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getRoomNumber() {  
 return roomNumber;  
 }  
  
 public void setRoomNumber(String roomNumber) {  
 this.roomNumber = roomNumber;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
  
 public boolean isOccupied() {  
 return isOccupied;  
 }  
  
 public void setOccupied(boolean occupied) {  
 isOccupied = occupied;  
 }  
  
 public RoomType getType() {  
 return type;  
 }  
  
 public void setType(RoomType type) {  
this.type = type;  
 }  
  
 @Override（日志调取和错误检查）  
 public String toString() {  
 return "Room{" +  
 "id='" + id + '\'' +  
 ", roomNumber='" + roomNumber + '\'' +  
 ", price=" + price +  
 ", isOccupied=" + isOccupied +  
 ", type=" + type +  
 '}';  
 }  
}

**2.7、定义酒店服务表**

@Document(collection = "service\_info") //设置“service\_info”表  
public class ServiceInfo {  
 @Id //设置主键  
 private String id; // 服务ID (MongoDB自动生成)  
 @Indexed(unique = true) //设置唯一性  
 private String serviceName; // 服务名称 (应唯一)  
 private double servicePrice; // 服务单价  
 public ServiceInfo() {}  
 public ServiceInfo(String serviceName, double servicePrice) {  
 this.serviceName = serviceName;  
 this.servicePrice = servicePrice;  
 }  
 // Getters and Setters  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getServiceName() {  
 return serviceName;  
 }  
  
 public void setServiceName(String serviceName) {  
 this.serviceName = serviceName;  
 }  
  
 public double getServicePrice() {  
 return servicePrice;  
 }  
  
 public void setServicePrice(double servicePrice) {  
 this.servicePrice = servicePrice;  
 }  
  
 @Override（日志调取和错误检查）  
 public String toString() {  
 return "ServiceInfo{" +  
 "id='" + id + '\'' +  
 ", serviceName='" + serviceName + '\'' +  
 ", servicePrice=" + servicePrice +  
 '}';  
 }  
}

**2.8、定义酒店服务请求表**

@Document(collection = "service\_requests") //建立"service\_requests"表  
public class ServiceRequest {  
 @Id //设置ID为主键  
 private String id;  
 private String checkInId; // 入住记录ID  
 private String customerIdCard; // 客户身份证号  
 private String roomNumber; // 房间号（便于展示）  
 private String serviceInfoId; // 服务项目ID  
 private String serviceName; // 服务名称  
 private int quantity; // 数量  
 private BigDecimal unitPrice; // 单价，新增  
 private LocalDateTime requestTime;// 申请时间  
 private String status; // APPLIED/COMPLETED/REJECTED  
 private String adminNote; // 管理员备注  
 private LocalDateTime completedTime; // 完成时间  
  
 // Getter and Setter methods  
 public String getId() { return id; }  
 public void setId(String id) { this.id = id; }  
  
 public String getCheckInId() { return checkInId; }  
 public void setCheckInId(String checkInId) { this.checkInId = checkInId; }  
  
 public String getCustomerIdCard() { return customerIdCard; }  
 public void setCustomerIdCard(String customerIdCard) { this.customerIdCard = customerIdCard; }  
  
 public String getRoomNumber() { return roomNumber; }  
 public void setRoomNumber(String roomNumber) { this.roomNumber = roomNumber; }  
  
 public String getServiceInfoId() { return serviceInfoId; }  
 public void setServiceInfoId(String serviceInfoId) { this.serviceInfoId = serviceInfoId; }  
  
 public String getServiceName() { return serviceName; }  
 public void setServiceName(String serviceName) { this.serviceName = serviceName; }  
  
 public int getQuantity() { return quantity; }  
 public void setQuantity(int quantity) { this.quantity = quantity; }  
  
 public BigDecimal getUnitPrice() { return unitPrice; }  
 public void setUnitPrice(BigDecimal unitPrice) { this.unitPrice = unitPrice; }  
  
 public LocalDateTime getRequestTime() { return requestTime; }  
 public void setRequestTime(LocalDateTime requestTime) { this.requestTime = requestTime; }  
  
 public String getStatus() { return status; }  
 public void setStatus(String status) { this.status = status; }  
  
public String getAdminNote() { return adminNote; }  
 public void setAdminNote(String adminNote) { this.adminNote = adminNote; }  
  
 public LocalDateTime getCompletedTime() { return completedTime; }  
 public void setCompletedTime(LocalDateTime completedTime) { this.completedTime = completedTime; }

public BigDecimal getSubtotal() {  
 if (unitPrice == null) return BigDecimal.ZERO;  
 return unitPrice.multiply(BigDecimal.valueOf(quantity));  
 }  
}

**3.功能实现**

## 7.4 分组与提交文档要求

以小组形式进行项目开发，每组成员限定为2至3人，组内成员将共享相同的分数。评分标准将依据实验的完成度以及题目的难度。

本实验的最终提交截止日期为\*\*\*\*，逾期提交将被自动判定为0分；同时，抄袭或直接从网络复制的实验内容也将得到0分。

所有小组需以小组为单位提交最终文档。每个小组应提交一份最终源代码和一份实验报告（实验报告中的个人总结部分要求每位成员都需独立完成），所有文件需打包并命名为“组长学号\_姓名\_实验7”的压缩文件（例如2302011220001\_李丽丽\_实验7）。提交内容应包括：

1. 程序源代码
2. 实验报告（文档格式自选，但必须包含以下内容，并以word或pdf格式提交）
   1. 所选开发环境、使用的工具、编程语言、数据库；
   2. 系统需求分析；
   3. 系统详细设计说明：包括整体流程图、总体设计图、各模块设计图等；
   4. 系统数据库设计与SQL脚本：（需包含E-R图和物理数据模型；包括创建表、视图、触发器、存储过程的SQL语句）；
   5. 系统实现细节：程序与数据库的连接说明、界面设计及其操作指南；
   6. 小组成员分工与评分：列出所有小组成员的学号、姓名及分工情况；提供组内评分细节，并解释每位成员获得相应评分的理由；
   7. 个人总结：（每位成员需提交一段100至200字的实验总结，并注明总结人）；

## 7.5参考图

物理数据模型图、系统流程图和模块设计图可参考如下。

