**详细设计方案**

1. **数据库模块详细设计**
   1. 数据库和数据表建立和初始化
      1. 数据库:售票管理TicketMaster的建立与初始化
      2. 数据表:优惠选项表

折扣(选项,金额)

*Discount(dChoose char(9) PRIMARY KEY, dMoney int);*

|  |  |
| --- | --- |
| 可选项*DChoose* | 金额 |
| 老人 | 75 |
| 小孩 | 50 |
| 团体 | 80 |
| 成人 | 100 |

* + 1. 数据表:用户表

用户表(票号, 游览时间, 优惠选项, 票状态)

UserTicket(uTicketId int, uUseDate date, dChoose char(9), uStatus char(9))

|  |
| --- |
| 票状态 |
| 已付款 |
| 已退款 |

* + 1. 数据表:交易记录表

交易记录表(序号, 票号, 用户名, 交易属性, 交易时间 ,优惠选项)

TradeRecord(tTradeId int PRIMARY KEY, uTicketId int,tUserId char(32),tReature char(9), tTradeDate DATE,dChoose char(9)

|  |
| --- |
| 交易属性 |
| 购票 |
| 退票 |

* 1. 存储过程设计
     1. 购票存储过程
        1. procBuyTicket(useDate DATE, choose CHAR(9),userName CHAR(32))
        2. 作用:由输入的游览日期,优惠选项,当前用户名进行购票
        3. 用户表:插入一条记录
           1. 票号取刚刚使用的票号
           2. 游览时间取参数时间
           3. 优惠选项取参数选项
           4. 票状态为已付款
        4. 交易记录表:插入一条记录
           1. 序号取最大序号加一
           2. 票号取最大票号加一
           3. 用户名取参数用户名
           4. 交易属性为购票
           5. 交易时间为当前日期
     2. 退票存储过程
        1. procRefund(ticketId int)
        2. 作用:由输入的票号进行退票
        3. 交易记录表:插入一条记录
           1. 序号取最大序号加一
           2. 票号取输入票号
           3. 用户名根据票号在交易记录表中查询
           4. 交易属性为退票
           5. 交易日期为当前日期
           6. 优惠选项根据票号从交易记录表中查询
        4. 用户表:修改一条记录(修改票状态为已退款)
           1. 根据输入票号修改票状态为”已退款”
     3. 统计指定日期各种价格的门票销售
        1. procCntDaySold(cntDate DATE,choose CHAR(9))

A. 当choose=’所有’时,统计条件为日期

对交易记录表和优惠选项表使用优惠选项进行连接

根据输入的日期和交易属性是’购票’为条件进行查询

使用SUM(*dMoney*),COUNT(\*)函数对查询结果进行计算

以同样的方法计算退票,两者相减得到指定日期的销售额和销售量

B. 当choose为其他优惠选项时,统计条件为日期和优惠选项

1. 对交易记录表和优惠选项表使用优惠选项进行连接
2. 根据输入的日期和所输入的优惠选项和交易属性是’购票’为条件进行查询
3. 使用SUM(*dMoney*),COUNT(\*)函数对查询结果进行计算
4. 以同样的方法计算退票,两者相减得到指定日期的销售额和销售量

V 统计指定月份的各种门票销售

* + - 1. procCntMonthSold(cntMonth CHAR(4),choose CHAR(9))

A. 当choose=’所有’时,统计条件为输入月份

对交易记录表和优惠选项表使用优惠选项进行连接

根据输入的月份(从数据库中截取)和交易属性是’购票’为条件进行查询

使用SUM(*dMoney*),COUNT(\*)函数对查询结果进行计算

以同样的方法计算退票,两者相减得到指定日期的销售额和销售量

B. 当choose为其他优惠选项时,统计条件为月份和优惠选项

1. 对交易记录表和优惠选项表使用优惠选项进行连接
2. 根据输入的月份(从数据库中截取)和所输入的优惠选项和交易属性是’购票’为条件进行查询
3. 使用SUM(*dMoney*),COUNT(\*)函数对查询结果进行计算
4. 以同样的方法计算退票,两者相减得到指定月份的销售额和销售量
   1. 源码文件设计:
      1. initDB.sql文件为数据库和数据表的创建和初始化
      2. proc.sql文件为所有的存储过程的创建
      3. database.h和macro.h为数据库在应用程序端的相关信息描述
      4. database.cpp为数据库相关操作的实现
5. **客户端程序详细设计**
   1. **主程序main**
      1. 源码路径:
         1. \TicketMaster\code\TicketMaster\main.cpp
      2. 作用:整个程序的入口, 调用初始化数据库接口函数, 并启动登录主窗口
      3. 行为:
         1. 初始化数据库相关信息并连接数据库
         2. 构造并show()主窗口mainWnd对象, mainWnd对象包含售票管理窗口adminWnd和用户购票窗口userWnd子对象
   2. **登录主窗口**
      1. 源码路径
         1. \TicketMaster\code\TicketMaster\mainwindow.cpp
         2. \TicketMaster\code\TicketMaster\mainwindow.h
      2. 作用:
         1. mainwindow.h为登录窗口类的声明,包括信号和槽函数的声明
         2. mainwindow.cpp为登录窗口类的槽函数和其他函数实现
      3. 行为:
         1. 登录行为:当点击登录按钮根据输入的用户名在数据库中判断是管理员或用户, 用户名在数据库中不存在则弹出错误
         2. 取消:退出应用程序
   3. **营业员(管理员)端** 
      1. 源码路径:
         1. \TicketMaster\code\TicketMaster\admindlg.cpp
         2. \TicketMaster\code\TicketMaster\admindlg.h
      2. 作用:
         1. admindlg.h为营业员窗口类的声明,包括信号和槽函数的声明
         2. admindlg.cpp为营业员窗口类的槽函数和其他函数实现
      3. 行为:
         1. 查询行为:查询结果为销售额和销售量
         2. 查询条件:两组任意组合共2\*5=10种查询条件
         3. 条件分组:时间类型{按月查询,按日查询},票类{全部,小孩,成人,老人}
         4. 可清空查询结果, 或退出登录返回到主登录界面
   4. **游客(用户)购票端**
      1. 源码路径:
         1. \TicketMaster\code\TicketMaster\userdlg.cpp
         2. \TicketMaster\code\TicketMaster\ userdlg.h
      2. 作用:
         1. userdlg.h为游客窗口类的声明,包括信号和槽函数的声明
         2. userdlg.cpp为游客窗口类的槽函数和其他函数实现
      3. 行为:
         1. 购票:选择优惠选项和游览时间点击购票即可付款,游览时间不能前于当天, 选择不同的优惠选项经数据库查询可以得到不同的金额
         2. 退票:选中右边列表中某一项, 点击退票可以实现退票, 退票时间不能后于票面游览时间, 退票后票状态为已退款
         3. 点击退出登录返回到主登录界面

**数据流图**