数据库工程作业

**要求**：

1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
2. 开发模式（B/S或C/S）、开发高级语言任选，后台数据库使用大型数据库管理系统（SQL Server、Oracle、MySQL等），不要使用桌面数据库。
3. 报告中所列举的四种操作，每种操作举一个例子即可。
4. 作业成绩按照报告中的标准评分，程序只实现报告中涉及的部分即可。
5. 作业完成后，请将工程作业报告和程序打包提交给助教老师，并联系助教老师进行系统说明和演示，回答相关问题。

**工程作业报告**

1. **项目信息（10分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 2313546 | 姓名 | 蒋枘言 | 专业 | 信息安全 |
| 项目名称 | 京津城际铁路列车模拟订票系统 | | | | |
| 必备环境 | Python，MySQL | | | | |
| 系统主要功能简介（4分） | 该项目实现了一个简易版的模拟京津城际铁路列车订票系统，主要功能有注册身份信息、模拟购票和退票、输入车站名查询该车站的车次信息、输入车次查询该车次的时刻表。 | | | | |
| 系统主要页面截图（6分） | 截3-4个页面即可 | | | | |

1. **系统配置（10分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **说明** | | （2分）请说明系统配置情况（后台数据库，高级语言）；  （8分）请使用连接串连接高级语言和数据库，并分析字符串的各个部分。 | | | |
| 配置步骤  2分 | DBMS | MySQL | | | |
| 高级语言 | Python | | | |
| 连接串 分析 （6分） | | 序号 | 名称 | 功能说明 | 取值 |
| 1 | connect\_db() | 在login\_register.py中与MySQL进行连接，从而进行对应的注册、登录、修改密码等操作。 | 无 |
| 2 | connect\_db() | 在ticket.py中与MySQL进行连接，从而进行对应的购票、退票等操作。 | 无 |
| 连接串代码（截屏）  （2分） | |  | | | |
| 备注 | | 数据库主机名叫localhost，用户名叫root，数据库名叫ticket\_booking，密码这里就不展示了。 | | | |

1. **数据库设计（14分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **说明** | （10分）按照数据表的创建顺序，依次给出所涉及数据表的信息，其中参照字段以“（字段1，字段2，……，字段n）”的形式给出，被参照字段以“表名（字段1，字段2，……，字段n）”的形式给出；  （4分）一般DBMS都可以为数据库生成关系图，请将该图片截屏并粘贴到表格中。 | | | | |
| 数据表 （10） | 创建顺序 | 数据表名称 | 主键 | 参照属性 | 被参照表及属性 |
| 1 | station | station\_id | 无 | |
| 2 | train | train\_id | departure | station.station\_name |
| destination | station.station\_name |
| 3 | stopovers | (train\_id,sequence) | train\_id | train.train\_id |
| station | station.station\_name |
| 4 | seat | (train\_id,seat\_id) | train\_id | train.train\_id |
| 5 | inventory | (train\_id,seat\_id,start\_station, end\_station,ticket\_status) | train\_id | train.train\_id |
| seat\_id | seat.seat\_id |
| start\_station | station.station\_name |
| end\_station | station.station\_name |
| 6 | passenger | user\_name | 无 | |
| 7 | ticket | ticket\_id | passenger\_id | passenger.id\_card |
| train\_id | train.train\_id |
| departure\_station | station.station\_name |
| arrival\_station | station.station\_name |
| seat\_id | seat.seat\_id |
| 关系图（4） | （截屏）  Diagram 1 | | | | |
| 备注 | 其中station、train、stopovers、seat、inventory、passenger和ticket是表，train\_schedule\_view是视图。 | | | | |

1. **含有事务应用的删除操作（13分）**

|  |  |  |
| --- | --- | --- |
| **说明** | （1分）简要说明该操作所要完成的功能；  （2分）该操作会涉及的表（必须含有两张或两张以上的关系表，同时以“表名”的形式给出）  （1分）表连接涉及字段描述（描述方式为“表1.属性=表2.属性”）  （1分）删除条件涉及的字段描述(以“表名.属性=？”形式给出)  （4分）实现该操作的关键代码（高级语言、SQL），截图即可；（其中如果删除语句中不包含任何形式的事务应用将扣除3分）  （4分）如何执行该操作，按所述方法能够正常演示程序则给分。 | |
| 功能描述（1分） | 该操作主要实现退票功能，用户可在“我的车票”里查询已购车票，点击下方“退票”按钮，选择对应的车票即可进行退票。退票后，数据库里相应的余票信息也会发生改变。 | |
| 涉及的表  （2分） | ticket,inventory | |
| 表连接涉及字段  （1分） | ticket.train\_id=inventory.train\_id  ticket.departure\_station=inventory.start\_station  ticket.arrival\_station=inventory.end\_station  ticket.seat\_id=inventory.seat\_id | |
| 删除条件字段描述 （1分） | 字段 | 规则 |
| ticket.ticket\_id | 选中订单号ticket\_id后进行退票，在表ticket按照ticket\_id属性进行索引，删除相应行。 |
| 代码 （4分） | 1. 退票操作的关键Python代码      1. 退票操作的触发器（用于更新余票信息） | |
| 程序演示（4分） | 使用乘客蒋枘言的账号进行登录，查询车票信息，发现有C2597次天津至滨海一等座，数据库表ticket里也有相应行。      表inventory里相应席位也显示“sold”。    进行退票：    再次查询车票信息，发现该车票已经消失，数据库表ticket里相应行也被删除了。      表inventory里相应席位也更新为“available”。 | |
| 备注 | 以上是一个完整的事务操作，不可分割。任何一个环节失败则整个事务进行回滚；否则提交。 | |

1. **触发器控制下的添加操作（20分）**

|  |  |  |
| --- | --- | --- |
| **说明** | （1分）简要说明该操作所要完成的功能；  （2分）简要说明该触发器所要完成的功能  （1分）该操作会涉及的表（以“表名”的形式给出）。  （2分）该操作输入数据以及输入数据应该满足的条件，如：数值范围、是否为空；  （6分）实现该操作的关键代码（高级语言、SQL），截图即可；  （8分）如何执行该操作，按所述方法能够正常演示程序则给分。 | |
| 功能描述  （1分） | 该操作主要实现购票功能，用户根据出发地和目的地进行车次查询，点击“购票”，选择车次和席别，系统将随机生成订单号和座位号，将车票的信息添加到数据库中。购票后，数据库里相应的余票信息也会发生改变。若余票不足时和同一人买相同车次、相同席别、相同出发地和目的地的车票时，无法购票成功。 | |
| 触发器描述  （2分） | 1. no\_duplicate\_ticket   该触发器主要是用于避免购相同的车票，当同一人买相同车次、相同席别、相同出发地和目的地的车票时，抛出错误，无法购票成功。   1. update\_inventory\_after\_book   该触发器主要是用于更新余票状态，当购买X车次A至B区间的座位S时，X车次的座位S在与A至B区间有交集的区间上均显示“sold”。 | |
| 涉及的表  （1分） | ticket,inventory | |
| 输入数据 （2分） | 字段 | 规则 |
| ticket\_id | 随机生成的不重复的8位大写字母和数字组成的字符串 |
| passenger\_id | 均不为空，并且都有外键约束 |
| train\_id |
| departure\_station |
| arrival\_station |
| seat\_id |
| 插入操作  源码 （3分） | 1. 判断是否还有余票      1. 生成订单号、分配座位号、添加购票信息 | |
| 触发器源码  （3分） | 1. no\_duplicate\_ticket      1. update\_inventory\_after\_book | |
| 程序演示  （4分） | 说明：不违背触发器能够执行插入操作。  还是登录乘客蒋枘言的账号，查询武清至天津的车票，假如购买C2575次商务座。      购票成功，再次查询武清至天津车票，发现余票成功扣减。在“我的车票”里也新增了该车票，数据库相应的余票信息也进行了更新。 | |
| 程序演示  （4分） | 说明：违背触发器要求，不能够执行插入操作，系统报错。  乘客蒋枘言再次尝试购买武清至天津C2575次商务座，发现购票失败。 | |
| 备注 | 以上是一个完整的事务操作，不可分割。任何一个环节失败则整个事务进行回滚；否则提交。 | |

1. **存储过程控制下的更新操作（18分）**

|  |  |  |
| --- | --- | --- |
| **说明** | （1分）简要说明该操作所要完成的功能；  （1分）简要说明该存储过程所要完成的功能；  （2分）说明该操作涉及操作的表（必须包含两张或两张以上的关系表，以“表名形式”描述）  （1分）表连接涉及字段描述（描述方式为“表1.属性=表2.属性”）  （2分）该操作会修改字段（以“表名.字段名”的形式给出），以及修改规则，如新数值的计算方法、在何种条件下予以修改等；  （6分）实现该操作的关键代码（高级语言、SQL），截图即可；  （5分）如何执行该操作，按所述方法能够正常演示程序则给分。 | |
| 功能描述（1分） | 此功能较为简单，就是输入用户名、原密码和新密码，来修改账号的密码。并且要求新密码不能和原密码一致。 | |
| 存储过程功能描述（1分） | 传入新密码和用户名，更新表passenger的password。 | |
| 涉及的关系表（2分） | passenger,ticket | |
| 表连接涉及字段（1） | passenger.id\_card=ticket.passenger\_id | |
| 更改字段 （2分） | 字段 | 规则 |
| password | 要求新密码不能和原密码一致即可。 |
| 更新代码 （3分） | 1. 更新操作      1. 触发器 | |
| 创建存储过程源码（3分） |  | |
| 存储过程执行源码  （1分） |  | |
| 程序演示（2分） | 说明：不违背存储过程，能够执行更新操作  查看表passenger，乘客蒋枘言的原密码是cr353888，想要将密码改为123456，尝试修改发现修改成功。      此时表passenger里的信息也成功更新。 | |
| 程序演示  （2分） | 说明：违背存储过程，系统报错  如果再次尝试将密码改为123456，发现修改失败，提示新密码必须与原密码不同。 | |
| 备注 | 修改密码首先需要确保原密码输入正确，然后保证新密码与原密码不同才可修改成功。 | |

1. **含有视图的查询操作（15分）**

|  |  |
| --- | --- |
| **说明** | （1分）简要说明该操作所要完成的功能；  （1分）简要说明建立的该视图的功能；  （2分）简要说明该操作涉及的关系数据表（以“表名”的形式给出）  （1分）简要说明表连接涉及的字段（以“表1.属性=表2.属性”）  （6分）实现该操作的关键代码（高级语言、SQL），截图即可；  （4分）如何执行该操作，按所述方法能够正常演示程序则给分。 |
| 操作功能描述（1分） | 此部分的功能就是根据用户输入的发站和到站查询车次信息和余票信息。 |
| 视图功能描述（1分） | 筛选出所有车次能够出票的合理区间，以及该区间的出发时间、到达时间和历时。 |
| 涉及的关系表（2分） | inventory,seat,stopovers |
| 表连接字段（1分） | inventory.train\_id=seat.train\_id=stopovers.train\_id  inventory.seat\_id=seat.seat\_id |
| 创建视图代码（3分） |  |
| 查询代码 （3分） |  |
| 程序演示（4分） | 输入发站和到站，点击“查询”，即可查询到车票信息。二等座、一等座、商务座下方的数字代表的是余票数量。 |
| 备注 | 视图筛选出了所有车次能够出票的合理区间，以及该区间的出发时间、到达时间和历时，在视图的基础上再进行车票查询就方便了许多。 |