**深 圳 大 学 实 验 报 告**

**课程名称：­ 数据库系统**

**实验项目名称： 数据库设计**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术、信息与计算专业**

**指导教师： JIANBIN QIN**

**报告组： 第0小组 成员：刘语文、刘俊楠、周琪**

**学号： 201852068、、2019191140、2017303010**

**实验报告提交时间： 2022年01月09日**

**教务处制**

|  |
| --- |
| 一、实验目的与要求  1、掌握编写网页页面以及连接云端数据库的方法  2、掌握在应用后台操纵数据库并实时交互的方法  3、掌握运行一个完整网页应用的方法。  4、掌握在完整网页上实现题目要求的方法。 |
| 二、实验环境  软件：Pycharm、腾讯云 |
| 1. 实验内容 2. 、建表过程- **Building the database (15 points)**   (1) First, determine the tables you will need by listing the name of each table and the fields it should contain. Avoid data redundancy. Do not create a field if it can be created by a “calculated field”in a query.  (2)You will need to record the transactions in a separate table. Avoid duplicating data.  (3)You must mark the appropriate primary key field(s) or foreign key field(s) for each table.  (4)Create at least five deals offered by Collective Coupons.  (5)Create more than 100 customers.  (6)Make sure that at least 100 customers sign up for some of the deals.  (7)Appropriately limit the size of the text fields; for example, a customer number does not need the default length of 255 characters.  根据1、2、3要求，我们分析得出共需要建立三张表格，分别为：Customer用户表，SignUp报名表以及Coupons优惠券表。具体ER图如下所示。  图示  描述已自动生成  图一 ER图  描述：customer和coupons是**二元的多对多关系**。一个用户可以报名多个优惠活动，一个优惠活动可以有多人参与。故，将所给字段信息拆分为三个表：customer、signup、coupons。Signup作为“关系”，连接两“实体”，连接的属性customer#和deal#皆为外键。图中，粗体倾斜且带有下划线的属性为对应表的主键。  对于用户表，我们增加了type和password属性。其中，我们将原表中的Customer Number字段，实例如“C2000”,拆分为两个属性, type和***customer#***，type标识用户类型（用户‘C’、商家‘M’or管理员‘A’），***customer#***作为整数id，用以实现序列自增。  具体建表代码如下图所示。    图二 Customer建表代码，**限制了相关属性的字段大小**，满足要求7    图三 Coupons建表代码    图四 SignUp报名表建表代码，**实现级联删除及外键引用**  具体约束创建如下：        注：sequence用以实现自增效果，Constraint用以检查字段。log\_table为后文要求实现的日志记录表（记录对er图中三张表的修改）。  导入相关数据满足要求4、5、6：  根据上述分析，将所给数据也拆分为对应表格，并另存为.csv文件，存于共享文件夹，利用copy语句导入。  图片包含 图表  描述已自动生成    可以通过前端验证：  (4)Create at least five deals offered by Collective Coupons.    (5)Create more than 100 customers.    (6)Make sure that at least 100 customers sign up for some of the deals.     1. 、题目要求-**PostgreSQL Implementation (55 points)**   注：本部分共7个要求   1. 编写一个存储过程Maximum Price，提示用户输入一个最大值，然后显示指定价格下所有交易的描述、位置、交易价格、可用日期和结束日期字段。   根据题目要求，我们在数据库内部实现maximum\_price函数。  9a2baa404702ca71a8aa032271319f7  前端页面输入一个最大值价格，后台接收到该价格数值后，在Coupons表中找出低于该价格的所有记录，并返回这些交易记录的描述、位置、交易价格、可用日期和结束日期字段并显示在前端页面。具体实现效果如下所示。    2、编写一个名为Percentage Bargain的存储过程。列出所有可用的交易，包括他们的交易编号，描述，地点，交易价格，和原始价格，然后计算百分比Bargain在交易栏。交易是交易价格和原价之间的百分比差。  为了实现题目功能，percentage\_bargain函数的具体实现语句函数如下所示。  004a1cef0734d62fd4ac395d826a3c7  图  在这个页面中，我们用一张表格展示了这些可用交易的交易编号、描述、地点、交易价格和原始价格等信息，同时也在后端计算得出优惠百分比并传至前端以显示。    3、创建一个名为“注册人数等于或超过100”的查询。在这个查询中，您需要确定哪些交易至少有100个成员注册。在输出中只显示交易编号Deal Number和描述Description的列。  为了题目要求，我们写了一个more\_than\_100函数。如下图为该函数的具体实现语句。  006af8c2d1e9f2017d85d67fac7957b  图  前端专门有一个页面用于展示“注册人数等于或超过100”的交易信息，页面表格中会显示交易对应的Deal Number和Description信息，如下图所示。    图   1. 写一个程序，叫做“最受欢迎的交易”。列出交易的描述以及有多少人注册了它。对输出进行排序，首先列出最受欢迎的交易。   为了实现如上功能，most\_popular\_deal函数的具体书写语句如下图所示。输出结果按照报名人数从高到低排序  b1db082bb5f6b78c6290b39ad833b46  图  同样，前端专门有一个页面用于展示“最受欢迎的交易”。我们按照注册人数降序展示出交易对应的描述，因此排列在第一个的即为最受欢迎交易，具体展示页面如下所示。    图  5、编写两个向coupon表和customers表添加元组的过程。例如，你可以使用一个过程，比如add\_coupon在coupon表中添加一个元组，其中Deal Number, Description, Location、Deal Price、Original Price、Ending Date等都是这个过程的参数。注意，Available Date应该设置为当前日期。  过程1：AddCoupon:  代码实现如下：AddCoupon从前端获取输入，调用CouponsInsert函数，其中AvailableDate无需输入，自动获取为当前日期。  fee44bb3cb70ab5c9ef45dfc99e3de6    图  因此，前端在按格式输入相应数据后点击“提交”按钮即可添加成功，且最终添加数据也可在Show Coupons页面中查找到，如下图所示。    图  过程2：AddCustomer,类似过程1，仅展示图片。        6、设置每张优惠券最多可供120名顾客使用。如果客户的数量超过限制，你的程序应该执行以下任务:  (1)打印提示问题的消息；  (2)使用触发器扩展客户的最大数量；  (3)使用日志表记录表上的更新操作(当任何表被修改时，自动向日志表添加一个元组)。  1641647745(1)  图  然而，考虑到每张优惠券最多可供120名顾客使用，因此，我们需要为向SignUp表的插入行为添加触发器。触发器语句如下图所示。  图  一旦发生向SignUp表的向SignUp表的信息插入行为，在插入信息结束后，我们会自动调用触发器函数来判断当前优惠券已报名人数是否已经超过120人（或可报名人数），一旦已报名人数已经超过可报名人数，即立刻将可报名人数额度上调10人来扩展客户的最大数量。同时，我们也使用日志表记录表上的更新操作，具体语句如下图所示。    图  图中触发器定义了三个变量e1/e2/e3,并在每次插入报名优惠券时，触发器会自动检查table\_audit有关优惠券的已拥有人数是否大于限制人数，如果大于则将其增长10，并且更新表。  7、你需要为所有的异常设计和显示适当的消息，使你的代码用户友好。  例如，如果有人想使用优惠券，但输入了不存在的交易号码，您的程序应该清楚地报告问题。  未完成   1. 界面- **Interface (20 points)**        1. 文档- **Documentation (10 points)**   按照每部分代码的功能来组织了应用，实现功能式架构    从上往下，static文件夹存放页面渲染相关js文件，template文件夹存放html文件，views文件夹存放路由装饰器。Crud文件集成对数据库的增删改查操作，database.py实现对数据库的连接及初始化，Models.py存放对表的定义。  在MyExp4文件夹之外，我们有配置文件及测试文件：    Config文件夹存放一般配置，instance/config.py文件存放特殊实例配置（密码、具体连接链接等）  下方为instance/config.py文件：     1. 加分亮点   1.实现用户登录注册界面及密码映射，使用werkzeug.security包，调用check\_password\_hash和generate\_password\_hash方法进行密码的产生和验证。具体过程为：数据库端存储密码哈希值，获取用户输入后立即产生哈希值与数据库端比对。    密码输入错误：    登录成功弹窗：    登录后左上角显示用户信息：    2.使用腾讯云数据库，利用pg\_dump工具实现了数据库迁移        3. 探索了实现相同功能的不同方法，即利用orm包或直接执行SQL语句  如：在实现MaximumPrice功能中，  直接使用包方法：  Coupons=db\_session.query(CouponsForm).all()  使用SQL语句，调用数据库内实现的函数  results=db\_session.execute(text('SELECT \* FROM maximum\_price(:max\_price)'), {'max\_price': max\_price}).all())  4.额外实现MyCoupons功能，用以查询特定用户报名了哪些优惠券    5.页面左侧集成所有表的增删改查及相关function功能: |
| 四、实验结论或体会  **实验结论：**  本次实验中我们实现了前端页面、后端代码以及本地数据库的连接，并在最后成功使用云数据库完成了指导文件中的所有要求。我们成功实现了对于不同数据表格的基础增删改查的功能，同时也实现了触发器以及对日志表的操作，并在最终完善了整体代码，使其变得具有鲁棒性，对用户更加友好。  **实验体会：**  实验中遇到了许多困难，在小组合作中也碰到了许多问题。  首先是在自己编写代码中碰上的各种各样的小问题。比如往虚拟机导入数据的时，需要用到VM共享文件夹。我们需要更新VMTools以及使用root用户设置可访问权限后，才可在postgres用户下调用csv文件。比如数据库连接问题，连接不同的网络时，虚拟机的地址也会转变。这需要我们查看VMware的产品文档，对它的网络设置有所了解。比如在设置云数据库时发现copy语句无法使用等等……  小组合作中也存在一些问题。在最初大家都不了解如何实现的情况下，我们的分工存在交叉，导致了一段时间的阻塞。其次是实现功能的思路不同，存在更熟悉orm包和更熟悉数据端的同学，导致了很大部分的重复工作。    通过本次实验，我们不仅学习到了如何应用数据库语言来实现具体功能，也逐渐在实际运用中掌握了如何前端与后端的数据传输以及触发器的使用。同时小组内的三个人也在不断沟通中渐渐明白了小组合作的重点在于能否进行合理而高效的交流以便和三人之力完成大作业。  实验分工说明：  大方向：  其中：周琪复制customer表相关需求实现，刘俊楠负责signUp表相关，刘语文负责coupons表相关。  PostgreSQL Implementation (55 points) 7个要求  刘语文：1，2，5+任务划分、文档撰写+前端+代码整合  刘俊楠：3，4，6+小组代码教师  周琪：5，6＋前端页面＋文档撰写  7共同负责，未实现 | |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。