第九章　第九次作业

数据库案例设计综合练习

某金融数据库系统背景：

某银行为了建设金融管理系统，需要对该系统的数据库进行设计，针对该系统，主要将对象分为客户、银行卡、理财产品、保险、基金。假设该系统存在着以下关系：客户可以办理银行卡，同时客户可以购买不同的银行产品，如理财产品，基金和保险。购买后的银行产品统称为客户的资产。

数据库系统业务需求描述：

- 客户可以办理多张银行卡，不仅可以办理储蓄卡，如果符合要求还可以办理信用卡，客户办理成功后获得银行卡的卡号。
- 银行提供多种理财产品，客户根据理财产品的信息（包括产品编号，产品名称，产品描述）购买理财产品，一个客户可以购买多种理财产品，不同的理财产品有不同的购买金额和理财年限。
- 银行提供多种保险，客户根据保险的信息（包括保险编号，保险名称，适用人群，保险项目）购买理财产品，一个客户可以购买多种保险，不同的保险有不同的保险金额和保险年限。
- 银行提供多种基金，客户根据基金的信息（包括基金编号，基金名称，基金类型，风险等级和基金管理者）购买理财产品，一个客户可以购买多种基金，不同的基金有不同的基金金额。
- 客户购买银行产品（理财产品、保险、基金）成为自己的资产后，需要记录这些资产的状态（可用，冻结），每个资产的购买数量，每个资产的收益和购买时间。

项目具体要求：

- 创建 finance 数据库作为项目数据库，数据库编码为 UTF-8。
- 在 finance 模式下完成金融管理系统中所有数据库对象（数据表）的创建，并完成数据的填充（可以由 navicat 自动生成）。其中，客户数据不少于 20 条，银行卡数据不少于 10 条，其他数据不少于 5 条；
- 对表添加外键约束，在银行信息表和资产信息表中，都存在每个银行卡必须有一个持卡者、每份资产必须都有一个资产拥有者这样的对应关系。因此针对这种对应关系，创建外键约束。
- 在理财产品表、保险信息表和基金信息表中，都存在金额这个属性，在现实生活中，金额不会存在负数。因此针对表中金额的属性，增加大于 0 的约束条件。
- 可以增加其他自定义约束。（可以用触发器来实现，不强制要求）

本次作业要求：

1. 分析系统功能，提交相应的 E-R 图（可以用画图软件（如 visio）进行 E-R 图设计，或者手绘后拍照上传超星平台）；(30 分)

2. 根据设计完成的 E-R 图，转换对应的关系模式，包括对象的具体属性描述和对象之间的关系描述，需至少满足 3NF 要求；(30 分)

3. 在 MySQL 中创建数据库，并加入适量的测试数据（某个关系截图，不用所有的关系都截图即可）；（10 分）

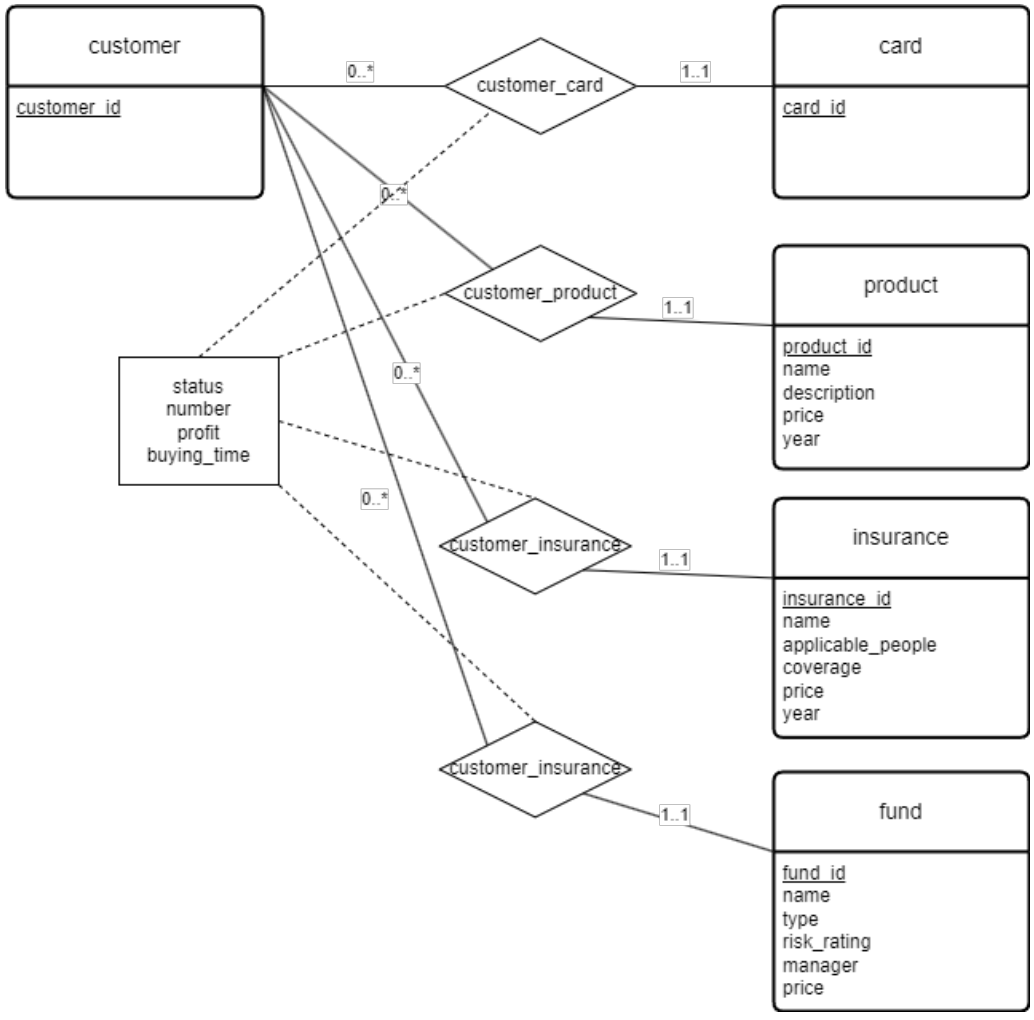
4. 根据需求创建合适的视图、存储过程、函数、触发器（提交代码）；（20 分）

5. 设计一个简单的用户界面，通过 JDBC 连接到该金融数据库，并测试完成简单的功能（测试代码和演示视频）。（10 分）

注意：本次作业注重数据库设计（包括 E-R 图、关系模式转换，以及数据库实践 SQL 操作的综合练习）；第 5 步仅做一个简单的 UI 界面即可（语言不限），不用太复杂。

解答

1. 分析系统功能，提交相应的 E-R 图（可以用画图软件（如 visio）进行 E-R 图设计，或者手绘后拍照上传超星平台）；(30 分)



2. 根据设计完成的 E-R 图，转换对应的关系模式，包括对象的具体属性描述和对象之间的关系描述，需至少满足 3NF 要求；(30 分)

转换对应的关系模式为：

customer(customer_id)

card(card_id)

product(product_id, name, description, price, year)

insurance(insurance_id, name, appliacable_people, coverage, price, year)

fund(fund_id, name, type, risk_rating, manager, price)

customer_card(customer_id, card_id, status, number, profit, buying_time)

customer_product(customer_id, product_id, status, number, profit, buying_time)

customer_insurance(customer_id, insurance_id, status, number, profit, buying_time)

customer_fund(customer_id, fund_id, status, number, profit, buying_time)

检查是否满足第三范式。这里有个问题是，例如 product，name 是否能确定 description, price, year，也就是是否允许同名产品存在。这里为了方便起见，就认为不存在这样的函数依赖，也就是允许同名产品存在。

那么这样的关系模式就满足第三范式了。

3. 在 MySQL 中创建数据库，并加入适量的测试数据（某个关系截图，不用所有的关系都截图即可）；(10 分)

```
create table customer(
    customer_id int auto_increment primary key
);

create table card(
    card_id int auto_increment primary key
);

create table product(
    product_id int auto_increment primary key comment '产品编号',
    name varchar(255) comment '产品名称',
    description varchar(255) comment '产品描述',
    price decimal(10, 2) comment '购买金额',
    year year comment '理财年限'
);

create table insurance(
```

```
    insurance_id int auto_increment primary key comment '保险编号',
    name varchar(255) comment '保险名称',
    applicable_people varchar(255) comment '适用人群',
    coverage varchar(255) comment '保险项目',
    price decimal(10, 2) comment '保险金额',
    year year comment '保险年限'
);

create table fund(
    fund_id int auto_increment primary key comment '基金编号',
    name varchar(255) comment '基金名称',
    type varchar(128) comment '基金类型',
    risk_rating int(3) comment '风险等级',
    manager varchar(64) comment '基金管理者',
    price decimal(10, 2) comment '基金金额'
);

create table customer_card(
    customer_id int,
    card_id int,
    status varchar(64) comment '状态',
    number int comment '购买数量',
    profit decimal(10, 2) comment '收益',
    buying_time datetime comment '购买时间',
    foreign key (customer_id) references customer(customer_id),
    foreign key (card_id) references card(card_id)
);

create table customer_product(
    customer_id int,
    product_id int,
    status varchar(64) comment '状态',
    number int comment '购买数量',
    profit decimal(10, 2) comment '收益',
    buying_time datetime comment '购买时间',
    foreign key (customer_id) references customer(customer_id),
    foreign key (product_id) references product(product_id)
);

create table customer_insurance(
    customer_id int,
    insurance_id int,
```

```
    status varchar(64) comment '状态',
    number int comment '购买数量',
    profit decimal(10, 2) comment '收益',
    buying_time datetime comment '购买时间',
    foreign key (customer_id) references customer(customer_id),
    foreign key (insurance_id) references insurance(insurance_id)
);

create table customer_fund(
    customer_id int,
    fund_id int,
    status varchar(64) comment '状态',
    number int comment '购买数量',
    profit decimal(10, 2) comment '收益',
    buying_time datetime comment '购买时间',
    foreign key (customer_id) references customer(customer_id),
    foreign key (fund_id) references fund(fund_id)
);
```

customer_insurance [root@localhost]

100 行 | Tx: 自动 | DDL | CSV

WHERE | ORDER BY

	customer_id	insurance_id	status	number	profit	buying_time
1	8	81	meFMKKBGzO	314	675.74	2004-05-12 06:58:53
2	51	56	BcOp3uqWSU	291	513.48	2005-06-16 04:24:32
3	82	6	lgYRZORb4d	200	527.19	2013-12-06 05:14:11
4	67	93	tCn300yeez	242	647.76	2021-08-28 15:26:23
5	74	26	teSJtA31ZI	919	96.82	2005-04-20 13:44:51
6	43	66	mHDVXqZdJe	824	724.13	2018-03-21 22:47:12
7	67	12	ZAZxCKvjhm	652	32.02	2005-04-07 17:06:05
8	97	51	1rqcQiLRSD	548	848.49	2011-06-24 03:08:12
9	47	90	jzyfmS2Iw2	361	4.45	2017-09-10 14:38:18
10	77	69	X67PMx4p6v	510	155.28	2023-12-23 23:49:35
11	7	99	vp0XSSBZ0I	405	133.91	2006-05-13 14:09:42
12	97	88	yYMBWnAt2B	684	894.89	2003-06-03 10:40:49
13	2	12	xaAQl2pDBi	898	767.84	2005-08-12 22:47:37
14	73	36	H9wd8uNMDu	43	882.41	2006-09-11 20:02:25
15	47	92	1J1gVrkJoF	76	466.36	2011-11-30 15:33:57
16	39	89	1HFh5dQjrw	179	74.99	2023-11-12 03:14:17
17	61	5	TqSe8tMdBt	388	503.18	2016-09-25 15:50:12
18	48	43	peFIUHcSEN	640	962.33	2002-01-30 18:27:32
19	82	58	ZIREenXre6	360	279.41	2005-12-01 20:46:37
20	67	67	Pdi8O64Lz6	322	29.54	2017-03-18 17:05:44
21	54	42	jBuqvbGPK7	930	265.88	2017-07-16 21:27:49
22	83	11	NsoxnbAYVZ	518	861.37	2012-02-29 06:36:59
23	94	38	ICM351tVcX	35	890.27	2020-09-25 00:54:47
24	50	33	NwTBc1svWs	307	233.15	2009-01-22 10:44:49
25	42	55	3Lc9LFcSBA	988	854.22	2009-03-27 04:21:52
26	77	91	66rOJOKd24	835	40.06	2006-06-08 09:47:26
27	59	47	dBoZLMjeJK	145	633.51	2000-11-27 11:35:18

4. 根据需求创建合适的视图、存储过程、函数、触发器（提交代码）；（20 分）

外键应该已经创建好了。那还剩下金额了。

```
alter table product add constraint _price_gt_0 check ( price > 0 );
alter table insurance add constraint _price_gt_0 check ( price > 0 );
alter table fund add constraint _price_gt_0 check ( price > 0 );
```

5. 设计一个简单的用户界面，通过 JDBC 连接到该金融数据库，并测试完成简单的功能（测试代码和演示视频）。（10 分）

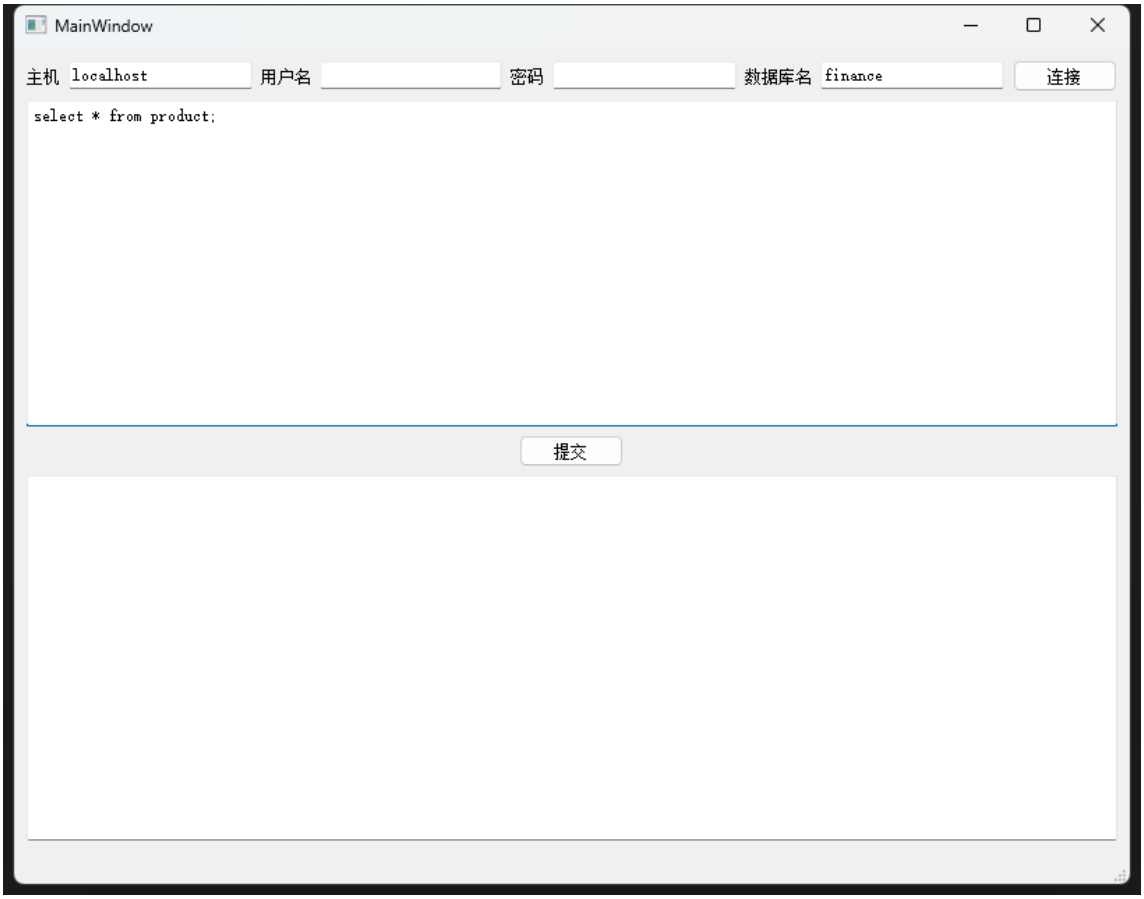
```
# -*- coding: utf-8 -*-

################################################################################
# Form generated from reading UI file 'designerLjvxhA.ui'
##
# Created by: Qt User Interface Compiler version 5.15.2
##
# WARNING! All changes made in this file will be lost when recompiling UI
↪  file!
################################################################################

import sys

from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
import pymysql


class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        if not MainWindow.objectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.resize(800, 600)
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.verticalLayout_2 = QVBoxLayout(self.centralwidget)
        self.verticalLayout_2.setObjectName(u"verticalLayout_2")
        self.horizontalLayout = QHBoxLayout()
        self.horizontalLayout.setObjectName(u"horizontalLayout")
```

```
self.label = QLabel(self.centralwidget)
self.label.setObjectName(u"label")

self.horizontalLayout.addWidget(self.label)

self.host_edit = QLineEdit(self.centralwidget)
self.host_edit.setObjectName(u"host_edit")

self.horizontalLayout.addWidget(self.host_edit)

self.label_2 = QLabel(self.centralwidget)
self.label_2.setObjectName(u"label_2")

self.horizontalLayout.addWidget(self.label_2)

self.user_edit = QLineEdit(self.centralwidget)
self.user_edit.setObjectName(u"user_edit")

self.horizontalLayout.addWidget(self.user_edit)

self.label_3 = QLabel(self.centralwidget)
self.label_3.setObjectName(u"label_3")

self.horizontalLayout.addWidget(self.label_3)

self.password_edit = QLineEdit(self.centralwidget)
self.password_edit.setObjectName(u"password_edit")

self.horizontalLayout.addWidget(self.password_edit)

self.label_4 = QLabel(self.centralwidget)
self.label_4.setObjectName(u"label_4")

self.horizontalLayout.addWidget(self.label_4)

self.database_edit = QLineEdit(self.centralwidget)
self.database_edit.setObjectName(u"database_edit")

self.horizontalLayout.addWidget(self.database_edit)

self.connect_button = QPushButton(self.centralwidget)
self.connect_button.setObjectName(u"connect_button")
```

```
self.horizontalLayout.addWidget(self.connect_button)


self.verticalLayout_2.addLayout(self.horizontalLayout)

self.verticalLayout = QVBoxLayout()
self.verticalLayout.setObjectName(u"verticalLayout")
self.sql_edit = QPlainTextEdit(self.centralwidget)
self.sql_edit.setObjectName(u"sql_edit")

self.verticalLayout.addWidget(self.sql_edit)

self.horizontalLayout_2 = QHBoxLayout()
self.horizontalLayout_2.setObjectName(u"horizontalLayout_2")
self.submit_button = QPushButton(self.centralwidget)
self.submit_button.setObjectName(u"submit_button")
sizePolicy = QSizePolicy(QSizePolicy.Maximum, QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)

↪  sizePolicy.setHeightForWidth(self.submit_button.sizePolicy().hasHeightForWidth
self.submit_button.setSizePolicy(sizePolicy)

self.horizontalLayout_2.addWidget(self.submit_button)


self.verticalLayout.addLayout(self.horizontalLayout_2)


self.verticalLayout_2.addLayout(self.verticalLayout)

self.result_text = QTextBrowser(self.centralwidget)
self.result_text.setObjectName(u"result_text")

self.verticalLayout_2.addWidget(self.result_text)

MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QMenuBar(MainWindow)
self.menubar.setObjectName(u"menubar")
self.menubar.setGeometry(QRect(0, 0, 800, 23))
MainWindow.setMenuBar(self.menubar)
```

```
        self.statusbar = QStatusBar(MainWindow)
        self.statusbar.setObjectName(u"statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)

        QMetaObject.connectSlotsByName(MainWindow)

        self.setup_other()
    # setupUi

    def retranslateUi(self, MainWindow):
        MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow",
        ↪  u"MainWindow", None))
        self.label.setText(QCoreApplication.translate("MainWindow",
        ↪  u"\u4e3b\u673a", None))
        self.label_2.setText(QCoreApplication.translate("MainWindow",
        ↪  u"\u7528\u6237\u540d", None))
        self.label_3.setText(QCoreApplication.translate("MainWindow",
        ↪  u"\u5bc6\u7801", None))
        self.label_4.setText(QCoreApplication.translate("MainWindow",
        ↪  u"\u6570\u636e\u5e93\u540d", None))
        self.connect_button.setText(QCoreApplication.translate("MainWindow",
        ↪  u"\u8fde\u63a5", None))
        self.submit_button.setText(QCoreApplication.translate("MainWindow",
        ↪  u"\u63d0\u4ea4", None))
    # retranslateUi

    def setup_other(self):
        self.host_edit.setText("localhost")
        self.user_edit.setText("test_user2")
        self.password_edit.setText("Rie4OhYi")
        self.database_edit.setText("finance")
        self.connect_button.clicked.connect(self.on_connect_clicked)
        self.submit_button.clicked.connect(self.on_submit_clicked)

    def on_connect_clicked(self):
        # 打开数据库连接
        self.conn = pymysql.connect(
            host=self.host_edit.text(), user=self.user_edit.text(),
            ↪  passwd=self.password_edit.text())
        self.conn.select_db(self.database_edit.text())
```

```
            self.result_text.setText(" 连接成功")

        def on_submit_clicked(self):
            cur = self.conn.cursor()
            cur.execute(self.sql_edit.toPlainText())
            all_result = ""
            while 1:
                res=cur.fetchone()
                if res is None:
                    # 表示已经取完结果集
                    break
                all_result += str(res) + "\n"
            self.result_text.setText(all_result)


    if __name__ == '__main__':
        app = QApplication(sys.argv)
        window = QMainWindow()
        main_window = Ui_MainWindow()
        main_window.setupUi(window)
        window.show()
        app.exec_()
```


MainWindow
主机
localhost
用户名
密码
数据库名
finance
连接
select * from product;
提交