

Секция 3

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В БИЗНЕСЕ

В. В. Вавуло

Институт бизнеса БГУ, г. Минск, Беларусь

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ИНФОРМАЦИИ О ДЕЯТЕЛЬНОСТИ БАНКА

В статье рассматривается процесс создания информационной системы учёта информации о деятельности банка, предложены методы работы с информацией, выбран язык программирования и система управления базой данных.

Ключевые слова: база данных, система баз данных, *sql server*, информационная система, банковская деятельность

На сегодняшний день базы данных о банковской деятельности используются повсеместно, поэтому разработка информационной системы является одной из наиболее актуальных тем. Приложение должно быть практичным, адаптивным, а также позволять хранить большое количество информации.

Предметной областью разрабатываемой базы данных является банк. Ведение базы данных по всем банковским операциям позволяет решать сразу несколько задач. Необходимо это и для самого банка, и для контролирующих организаций. Основная задача такой базы – предоставлять исчерпывающие данные, необходимые для дальнейшего моделирования поведения клиента.

Сюда входит: информация о клиентах: их номера кодов ИНН, ведомости о правах собственности на то или иное имущество; данные о депозитах и кредитах как физических, так и юридических лиц; вся информация о проходящих через банк платежах, в том числе, переводов на карточки, оплат по счет-фактуре.

Изучив ее, эксперт может сказать, ответственно ли относится потребитель к имеющимся кредитам, являются ли просрочки для него делом обычным, либо же он платит точно в срок. На основании такой информации, вероятно, будет принято окончательное решение по заявке. И так происходит практически в каждом департаменте.

Еще одно назначение такого источника информации – погашение задолженностей банка перед вкладчиками в случае признания его неплатежеспособным. Ее наличие – единственная возможность для вкладчиков вернуть собственные средства в полном объеме, не прибегая к судебным разбирательствам.

Таким образом, информационная система должна быть простой в использовании, безопасной в работе, а также достоверной в части выдаваемой информации. Это позволит ускорить обработку информации и повысить точность учета.

Проектирование базы данных представляет собой процесс отображения исследуемых явлений реального мира, называемых предметной областью, в виде данных в памяти компьютера.

Исходя из результатов рассматриваемой предметной области нами выделены следующие типы сущностей и их атрибуты:

– Customer (Клиент): Customer_ID (Клиент_ID); Customer_Type_ID (Клиент_Тип_ID); Customer_Name (Клиент_Имя); Customer_Phone (Клиент_Телефон); Customer_Email (Клиент_Email); Date_Registration (Дата_Регистрации); Login (Логин); Password (Пароль).

– Customer_Type (Клиент_Тип): Customer_Type_ID (Клиент_Тип_ID); Customer_Type (Клиент_Тип).

– Merchant (Продавец): Merchant_ID (Продавец_ID); Merchant_Name (Продавец_Имя); Merchant_Phone (Продавец_Телефон); Merchant_Email (Продавец_Email); Other_Details (Другие_Подробности).

– Products_Services (Продукты_услуга): Products_Service_ID (Продукт_Услуга_ID); Products_Service (Продукт_Услуга); Merchant_ID (Продавец_ID).

- Account (Счет): Account_ID (Счет_ID); Account_Type_ID (Счет_Тип_ID); Customer_ID (Клиент_ID); Account_Name (Счет_Название); Date_Opened (Дата_Открытия); Other_Details (Другие_Подробности).
- Account_Type (Счет_Тип): Account_Type_ID (Счет_Тип_ID); Account_Type (Счет_Тип).
- Customer_Purchase (Клиент_Покупка): Purchase_ID (Покупка_ID); Customer_ID (Клиент_ID); Products_Service_ID (Продукт_Услуга_ID); Date (Дата); Quantity (Количество); Other_Details (Другие_Подробности).
- Transactions (Транзакции): Transaction_ID (Транзакция_ID); Transaction_Type_ID (Транзакция_Тип_ID); Account_ID (Счет_ID); Customer_ID (Клиент_ID); Purchase_ID (Клиент_Покупка_ID); Date (Дата); Amount_Transaction (Размер_Транзакция); Other_Details (Другие_Подробности).
- Transaction_Type (Транзакции_Тип): Transaction_Type_ID (Транзакция_Тип_ID); Transaction_Type (Транзакция_Тип).

Между сущностями установлена связь один-ко-многим: «Customer_Type» – «Customer», «Customer» – «Account», «Customer» – «Customer_Purchase», «Account_Type» – «Account», «Merchant» – «Products_Services», «Products_Services» – «Customer_Purchase», «Customer_Purchase» – «Transactions», «Transaction_Type» – «Transactions», «Account» – «Transactions». В данном случае связи одна из связываемых сущностей является родительской (главной), а другая – дочерней (подчиненной) [1].

Проектирование базы данных разбивают на два основных этапа: инфологический (концептуальный) и даталогический. В первую очередь разрабатывается концептуальная схема БД.

Концептуальная модель отражает семантику предметной области в виде совокупности сущностей, их характеристик (атрибутов) и связей (ассоциативных отношений между сущностями).

Логический уровень модели базы данных в ERwin Data Modeler.

В физической модели сущности преобразуются в таблицы, атрибуты и кортежи – соответственно в столбцы и строки таблиц, а их домены отображаются в типы данных, принятые в конкретной СУБД.

На этапе даталогического моделирования концептуальная схема преобразуется в логическую с использованием терминов выбранной СУБД. Результат моделирования представлен на рис. 1.

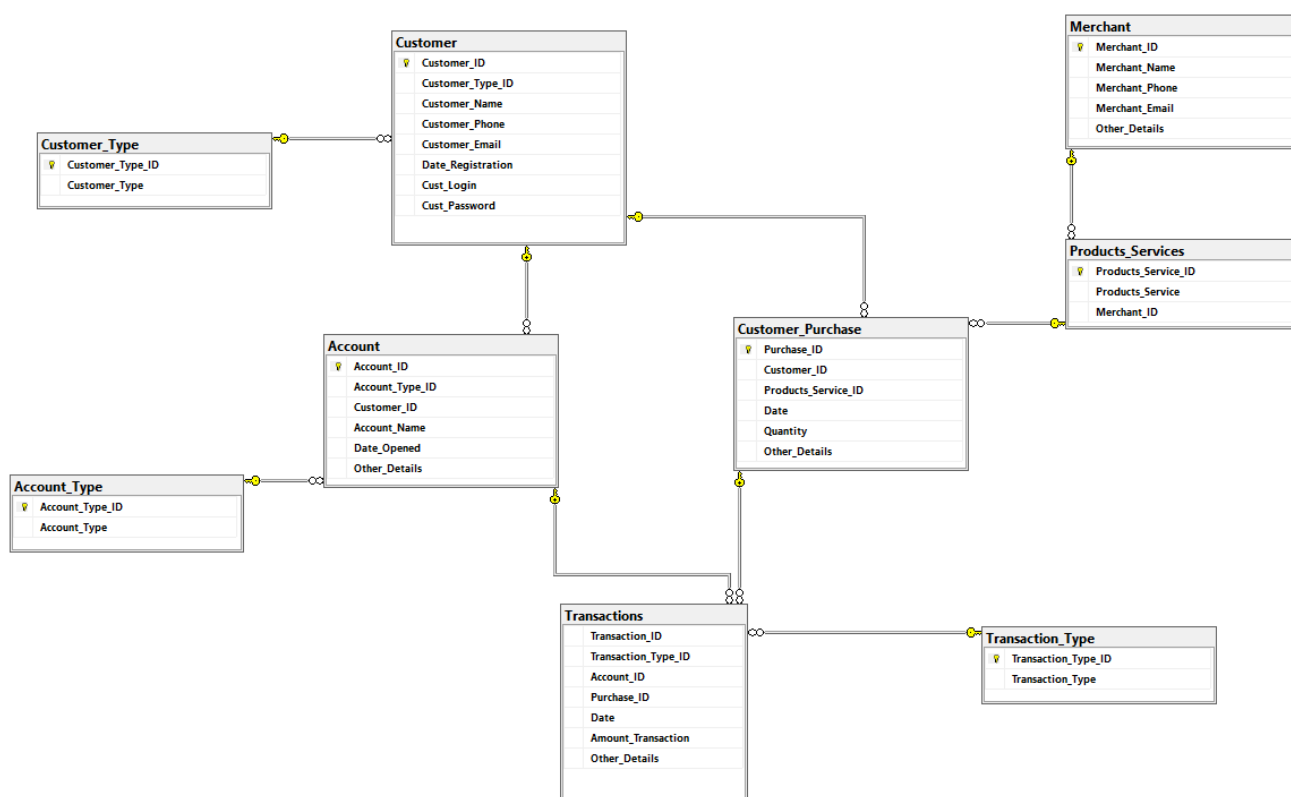


Рис. 1. Диаграмма базы данных

В ходе выполнения этапа разработки базы данных были сделаны запросы для создания таблиц с учетом ограничений целостности данных, ограничений на значения и т. п. [2].

В результате запроса в обозревателе объектов базы данных мы можем наблюдать созданные таблицы.

С помощью кода, были созданы запросы для создания уникальных индексов соответствующих таблиц.

Представление (Views) – виртуальная таблица, представляющая собой поименованный запрос, который будет подставлен как подзапрос при использовании представления. Для его создания используется команда CREATE VIEW [3].

Данное представление будет отображать информацию о транзакциях клиентов, включая их имена, названия счетов, типы транзакций, даты и суммы транзакций. В запросе используются внутренние соединения для объединения таблиц Transactions, Account, Customer и Transaction_Type. Это представление можно использовать для быстрого доступа к информации о транзакциях клиентов без необходимости написания сложных запросов каждый раз.

Далее мы создаем представление, в которое добавляем вычисляемое поле: «Transaction_Category», а также сортировку сначала по имени клиента, а затем по дате транзакции.

В данной работе были разработаны хранимые процедуры для решения актуальных задач в банке с различным числом и типами данных входных и выходных параметров.

Также в SQL Server есть возможность создавать процедуру, функцию, триггер с использованием курсора.

Курсор – механизм, предоставляющий пользователю доступ к любой строке набора данных, связанного с курсором и формируемого с помощью команды выборки данных SELECT.

Триггеры – подпрограммы на языке SQL, которые хранятся в базах данных. Они выполняются автоматически при операциях вставки, обновления и удаления данных в таблицах базы данных.

В нашей работе были созданы триггеры различных типов для того, чтобы превратить сервер в систему, оперативно реагирующую на изменения.

В рамках администрирования базы данных созданы роли и пользователи, которым были назначены полномочия на объекты базы данных.

С помощью команды CREATE ROLE были разработаны пользовательские роли базы данных, а также назначены их полномочия.

Здесь роль SystemAdmin имеет полный доступ ко всем объектам базы данных, роль SecuritySpecialist имеет доступ только к таблицам Customer, Account и Transactions для проверки безопасности, роль DepartmentManager имеет доступ к таблицам Merchant, Customer_Purchase и Products_Services для управления отделом, роль Cashier имеет доступ к таблицам Customer, Account и Transactions для выполнения операций на кассе, а роль Customer имеет доступ только к своим собственным данным, таким как информация о счетах и транзакциях.

Также был разработан план резервного копирования и создана резервная копия базы данных.

Для разработки интерфейса приложения была использована платформа.NET (язык C#).

На рис. 2 представлена первая форма, которая является главной страницей приложения. С помощью формы «Menu.cs» осуществляется переход на такие формы, как «Клиенты и их покупки», «Счета и транзакции» и «Продавцы, товары и услуги».

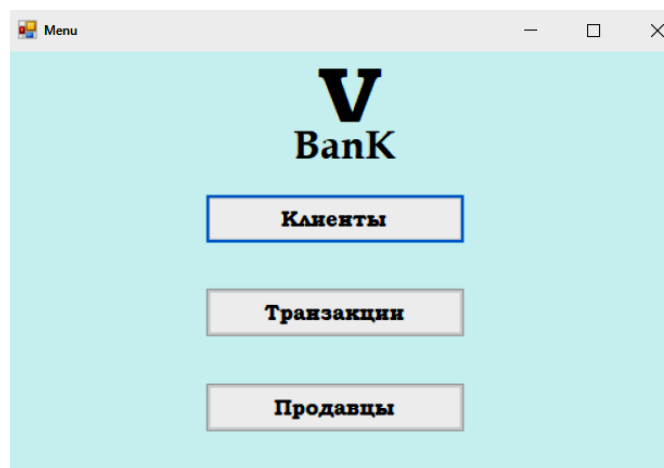


Рис. 2. Главная страница приложения

Над каждой таблицей располагается специальная панель навигации (BindingNavigator), позволяющая добавлять, удалять данные из таблицы, а также сохранять их.

На формах «Клиенты и их покупки» и «Счета и транзакции» находится поиск по типу клиента, типу счёта и типу транзакции с помощью кнопки «Поиск».

Таким образом, было создано приложение для улучшения пользовательского опыта созданной базы данных. Благодаря информационной системе все данные стали быстро доступны и хорошо структурированы, можно легко добавлять, сохранять, удалять и редактировать необходимую информацию, а также осуществлять поиск по заданным критериям. Приложение обладает удобным и понятным интерфейсом, что позволяет без проблем выполнять определенные задачи.

Список использованных источников

1. Внешние ключи и связи [Электронный ресурс] // Metanit.com – 2017. – Режим доступа: <https://metanit.com/sql/tutorial/1.3.php#:~:text=Связь%20один%20ко%20многим,%2C%20а%20таблица%20статей%20-%20дочерней.> – Дата доступа: 25.09.2023.

2. Создание и удаление таблиц [Электронный ресурс] // Metanit.com – 2017. – Режим доступа: <https://metanit.com/sql/sqlserver/3.2.php>. – Дата доступа: 15.10.2023.

3. Создание и удаление базы данных [Электронный ресурс] // Metanit.com – 2017. – Режим доступа: <https://metanit.com/sql/sqlserver/3.1.php#:~:text=Для%20создания%20базы%20данных%20используется%20команда%20CREATE%20DATABASE.> – Дата доступа: 10.10.2023.