**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc184573853)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc184573854)

[2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 6](#_Toc184573855)

[2.1 Этап концептуального проектирования 6](#_Toc184573856)

[2.1.2 Описание связей сущностей базы данных. 10](#_Toc184573857)

[2.1.3 Концептуальная модель данных. 12](#_Toc184573858)

[2.2 Этап логического проектирования 14](#_Toc184573859)

[2.2.1 ER-диаграмма в среде MySQL Workbench. 14](#_Toc184573860)

[2.2.2 Анализ ER-диаграммы. 14](#_Toc184573861)

[2.2.3 Окончательная ER-диаграмма 15](#_Toc184573862)

[2.3 Этап физического проектирования 15](#_Toc184573863)

[2.3.1 Генерация базы данных 15](#_Toc184573864)

[2.3.2 Схема данных в среде выбранной СУБД. 15](#_Toc184573865)

[3 ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ 16](#_Toc184573866)

[ЗАКЛЮЧЕНИЕ 16](#_Toc184573867)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 16](#_Toc184573868)

# ВВЕДЕНИЕ

В условиях современного мира банковские услуги играют ключевую роль в экономике любой страны. Кредитование является одной из важнейших функций банков, поскольку оно позволяет физическим и юридическим лицам получать необходимые средства для развития бизнеса, приобретения недвижимости, оплаты образования и других целей. Эффективность управления кредитными операциями напрямую влияет на финансовую устойчивость банка и удовлетворенность клиентов.

Кредитный процесс включает множество этапов, начиная от подачи заявки клиентом до погашения кредита. Каждый этап требует тщательного учета и анализа информации о заемщике, условиях кредитования, сроках выплат и возможных рисках. Ручная обработка таких данных может привести к ошибкам, задержкам и неэффективному использованию ресурсов. Поэтому автоматизация процесса учета кредитов становится необходимым условием успешного функционирования любого банка.

Целью данной курсовой работы является проектирование базы данных для учета кредитов в банке, которая позволит эффективно управлять процессом кредитования, минимизировать риски и повысить качество обслуживания клиентов. Проектирование такой системы направлено на создание удобного инструмента для хранения, обработки и анализа кредитных операций, а также обеспечения безопасности данных.

Для достижения поставленной цели необходимо решить следующие задачи:

* Анализ требований к системе
* Проектирование структуры базы данных
* Реализация базы данных
* Тестирование и оценка эффективности
* Разработка пользовательского интерфейса
* Документирование проекта

Разработанная база данных должна обеспечивать выполнение следующих функций:

* Хранение информации о клиентах, включая персональные данные, историю обращений за кредитами и платежную дисциплину
* Учет условий кредитования, таких как сумма кредита, процентная ставка, срок погашения и график платежей
* Формирование отчетов по кредитным операциям для руководства банка
* Соответствие требованиям контроля качества данных хранимой информации в части полноты, согласованности и соответствия общепринятым стандартам

Таким образом, данная курсовая работа направлена на решение актуальной проблемы автоматизации учета кредитов в банке, что способствует повышению эффективности работы финансовых учреждений и улучшению качества предоставляемых услуг.

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Объектом исследования является процесс учета кредитов в банке. Основная цель данного процесса – управление информацией о кредитах, выданных клиентам банка, а также контроль за своевременным погашением задолженностей и соблюдением условий договоров кредитования.

Основные объекты базы данных:

* Сотрудник банка
* Организация – клиент юридическое лицо
* Директор организации
* Область и город регистрации организации
* Кредит
* Особые условия для кредитования
* Кредитный договор
* Операции гашения кредита
* Отчеты о возвратах денежных средств

Функции системы:

* Управление клиентами
* Работа с договорами
* Расчет процентов и штрафов
* Регистрация поступивших от клиентов платежей
* Формирование отчетов

Таким образом, данная база данных позволит эффективно управлять процессом выдачи и контроля кредитов, обеспечивая удобство работы сотрудников банка и предоставляя актуальную информацию для анализа и принятия решений.

# 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

## 2.1 Этап концептуального проектирования

Концептуальное проектирование – это начальный этап разработки базы данных, который фокусируется на создании модели, отражающей реальные объекты и их взаимосвязи без учета особенностей конкретной СУБД. На этом этапе важно определить ключевые сущности, их атрибуты и связи между ними для того, чтобы корректно отразить бизнес-процессы банка в области учета кредитов.

2.1.1 Описание сущностей базы данных.

Для разработки базы данных по учету кредитов в банке, необходимо определить ключевые сущности, каждая из которых будет представлять собой отдельную таблицу в базе данных. В таблице 1 приведен список сущностей базы данных.

Таблица 1 – Выделение сущностей

|  |  |
| --- | --- |
| Наименование | Описание |
| Сотрудник | Сотрудники банка, работающие с кредитными операциями. |
| Область | Содержит перечень областей, в которых находятся клиенты банка. |
| Город | Содержит список городов, где зарегистрированы клиенты банка |
| Директор организации | Сущность содержит информацию о директоре организации-клиента |
| Организация | Юридические лица – клиенты банка. |
| Кредит | Кредит, предоставляемый банком клиентам. |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Особые условия | Дополнительные условия кредитования, которые могут варьироваться в зависимости от конкретного договора. Например: льготные ставки. |
| Договор | Сущность фиксирует договоры, заключаемые между банком и организациями-клиентами на предоставление кредита. |
| Операция гашения | Запись об операции погашения части кредита или полного закрытия долга. |

В таблице 2 представлено описание каждой сущности с указанием основных атрибутов.

Таблица 2 – Сущности с описанными атрибутами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сущность | Атрибут | Ключ | Домен | | Примечание |
| Тип | Размер |  |
| Сотрудник | Идентификатор | ПК | число | 19 |  |
| ФИО |  | текст | 155 |  |
| Должность |  | текст | 155 |  |
| Отдел | ФК | число | 19 |  |
| Email |  | текст | 155 |  |
| Номер телефона |  | текст | 12 |  |
| Область | Идентификатор | ПК | число | 19 |  |
| Название |  | текст | 155 |  |
| Город | Идентификатор | ПК | число | 19 |  |
| Название |  | текст | 155 |  |
| Код области | ФК | число | 19 |  |

Продолжение таблицы 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Директор организации | Идентификатор | ПК | число | 19 |  |
| ФИО |  | текст | 155 |  |
| Серия паспорт |  | текст | 4 |  |
| Номер паспорт |  | текст | 6 |  |
| Дата рождения |  | дата |  |  |
|  | Кем выдан |  | текст | 155 |  |
| Дата выдачи |  | дата |  |  |
| Телефон |  | текст | 12 |  |
| Организация | Идентификатор | ПК | число | 19 |  |
| Название |  | текст | 155 |  |
| ИНН |  | текст | 10 |  |
| КПП |  | текст | 9 |  |
| Организационно-правовая форма |  | текст |  |  |
| Полный адрес организации |  | текст | 255 |  |
| Директор | ФК | число | 19 |  |
| Телефон |  | текст | 11 |  |
| Электронная почта |  | текст | 155 |  |
| Код области | ФК | число | 19 |  |
| Код города | ФК | число | 19 |  |
| Дата регистрации организации |  | дата |  |  |

Продолжение таблицы 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Кредит | Идентификатор | ПК | число | 19 |  |
| Сумма |  | число | 20 |  |
| Процентная ставка |  | число | 2 |  |
|  | Дата начала кредита |  | дата |  |  |
| Дата завершения кредита |  | дата |  | предполагаемая |
| Статус | ФК | число | 19 |  |
| Организация | ФК | число | 19 |  |
| Возвращенная сумма |  | число | 50 | Сумма всех операций гашения в рамках 1- ого договора |
| Особые условия | Идентификатор | ПК | число | 19 |  |
| Описание |  | текст | 255 |  |
| Снижение процентной ставки |  | число | 2 |  |
| Договор | Идентификатор | ПК | число | 19 | Номер договора |
| Дата подписания |  | Дата |  |  |
| Срок действия |  | Дата |  |  |
| Идентификатор кредита | ФК | число | 19 |  |

Продолжение таблицы 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Идентификатор сотрудника банка | ФК | число | 19 |  |
| Особые условия | ФК | число | 19 |  |
|  | Процентная ставка итог |  | число | 2 | Кредит.процентная ставка - Особые условия.сниженние процентной ставки |
| Пени |  | число | 2 | процент штрафа за каждый просроченный день |
| Операция гашения | Идентификатор | ПК | число | 19 |  |
| Идентификатор кредита | ФК | число | 19 |  |
| Дата операции |  | дата |  |  |
| Сумма |  | число | 50 |  |

### 2.1.2 Описание связей сущностей базы данных.

Описание связей сущностей на этапе концептуального анализа является важным шагом при проектировании базы данных, так как оно помогает понять структуру данных и установить правила взаимодействия между ними. Это позволяет создать эффективную и устойчивую систему управления данными. В таблице 3 приведено описание связей сущностей базы данных.

Таблица 3 – Описание связей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сущность 1 | Связь | Сущность 2 | Тип | Пояснение |
| Сотрудник | Заключает | Договор | 1:М | Несколько сотрудников могут участвовать в оформлении одного кд, и один сотрудник может оформлять много кд. |
| Область | Находится | Город | 1:M | В каждой области находится несколько городов, но город принадлежит только одной области. |
| Директор организации | Принадлежит | Организация | 1:M | У одной организации может быть только один директор, но один человек может руководить несколькими организациями. |
| Организация | Находится | Город | 1:М | В каждом городе может быть зарегистрировано несколько организаций, но одна организация может быть зарегистрирована в одном городе |
| Берет | Кредит | 1:M | Каждая организация может иметь несколько кредитов, но каждый кредит выдан конкретной организации. |

Продолжение таблицы 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кредит | Имеет | Операция гашения | 1:M | По каждому кредиту может проводиться множество операций погашения, но каждая операция привязана к конкретному кредиту. |
| Договор | Содержит | Кредит | 1:1 | Каждый кредит имеет свой договор, и каждый договор относится к конкретному кредиту. |
| Содержит | Особые условия | M:M | Для одного кредита могут применяться различные особые условия, но одно особое условие может относиться к нескольким кредитам. |

### 2.1.3 Концептуальная модель данных.

Концептуальная модель данных в стандарте Чена представляет собой диаграмму «сущность-связь» (ERD), которая отображает сущности и взаимосвязи между ними. В данном случае мы имеем дело с базой данных учета кредитов в банке, где каждая сущность будет представлена в виде прямоугольника, а связи между ними – линиями с указанием типа связи (один к одному, один ко многим, многие ко многим). На рисунке 1 представлена концептуальная модель данных в стандарте Чена. Сущности для диаграммы были представлены в таблице 1, а их взаимосвязи в таблице 3.

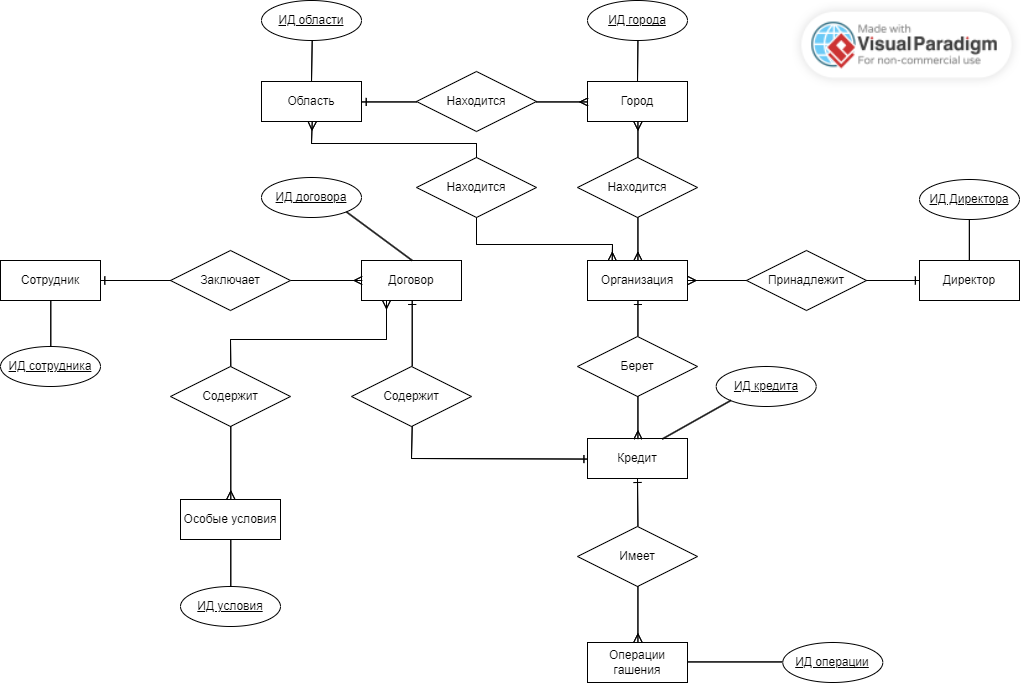


Рисунок 1 – Концептуальная модель данных в стандарте Чена

## 2.2 Этап логического проектирования

Логическое проектирование базы данных – это важный этап разработки системы учета кредитов в банке. Его цель заключается в создании модели данных, которая будет отражать бизнес-процессы банка и поддерживать их эффективное выполнение. На этом этапе производится проектирование ER-диаграммы, ее анализ и оптимизация.

### 2.2.1 ER-диаграмма в среде MySQL Workbench.

ER-диаграмма в MySQL Workbench представляет собой графическое отображение структуры базы данных, включая сущности (таблицы), их атрибуты (колонки) и связи между ними. Она помогает визуально представить и понять, как данные будут организованы и связаны друг с другом.

На основе описанных ранее сущностей (таблицы 1 и 2) и их взаимосвязей (таблица 3) построим ER-диаграмму, представленную на рисунке 2.

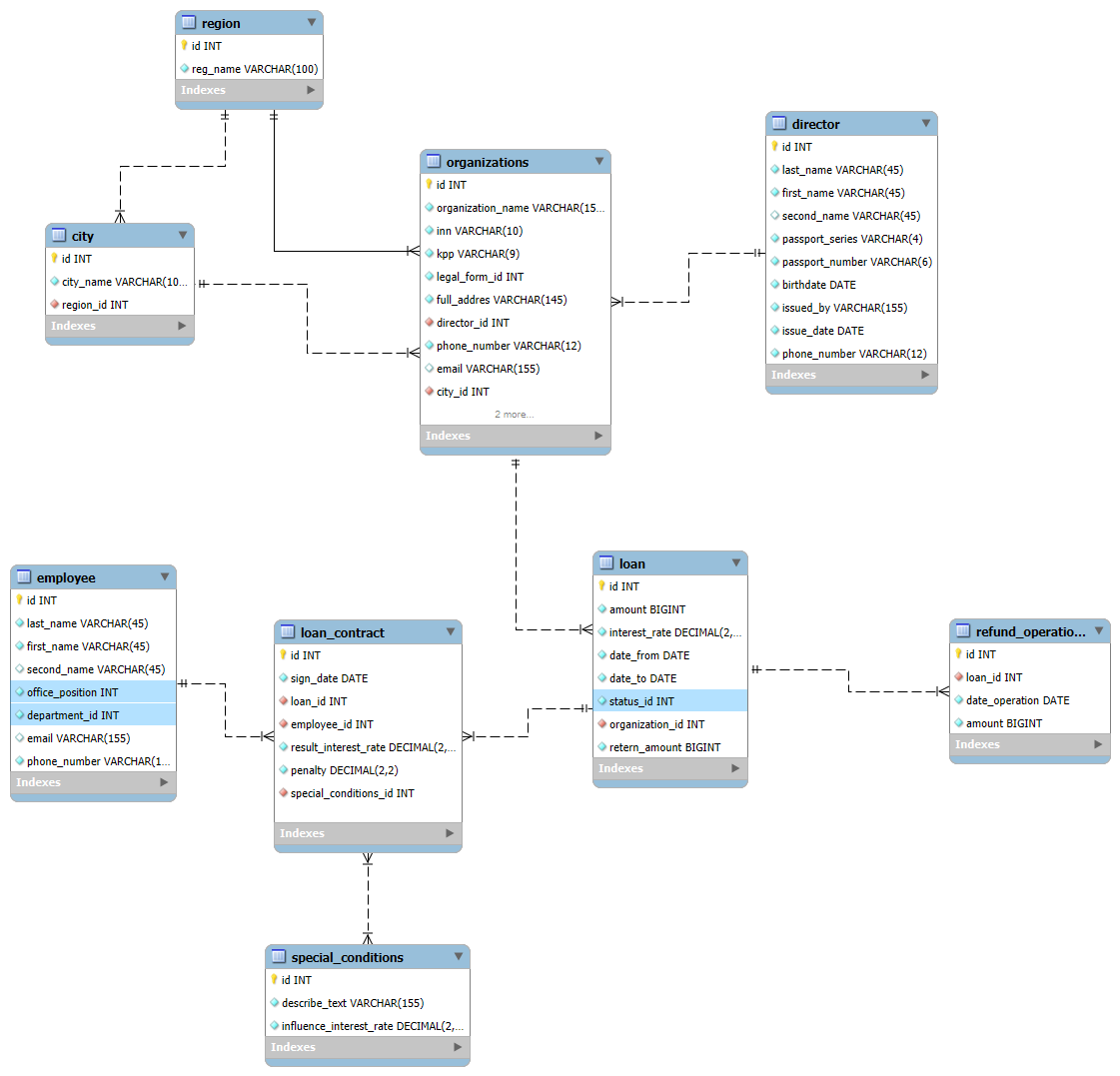


Рисунок 2 – ER-диаграмма в MySQL Workbench

### 2.2.2 Анализ ER-диаграммы.

Нормализация структуры данных в базе данных — это процесс приведения данных к такому виду, при котором исключаются избыточность, противоречивость и потенциальные аномалии при внесении, изменении и удалении данных. Этот процесс является одним из важнейших этапов проектирования реляционных баз данных и направлен на повышение качества и надежности хранения данных.

Выделим дополнительные сущности:

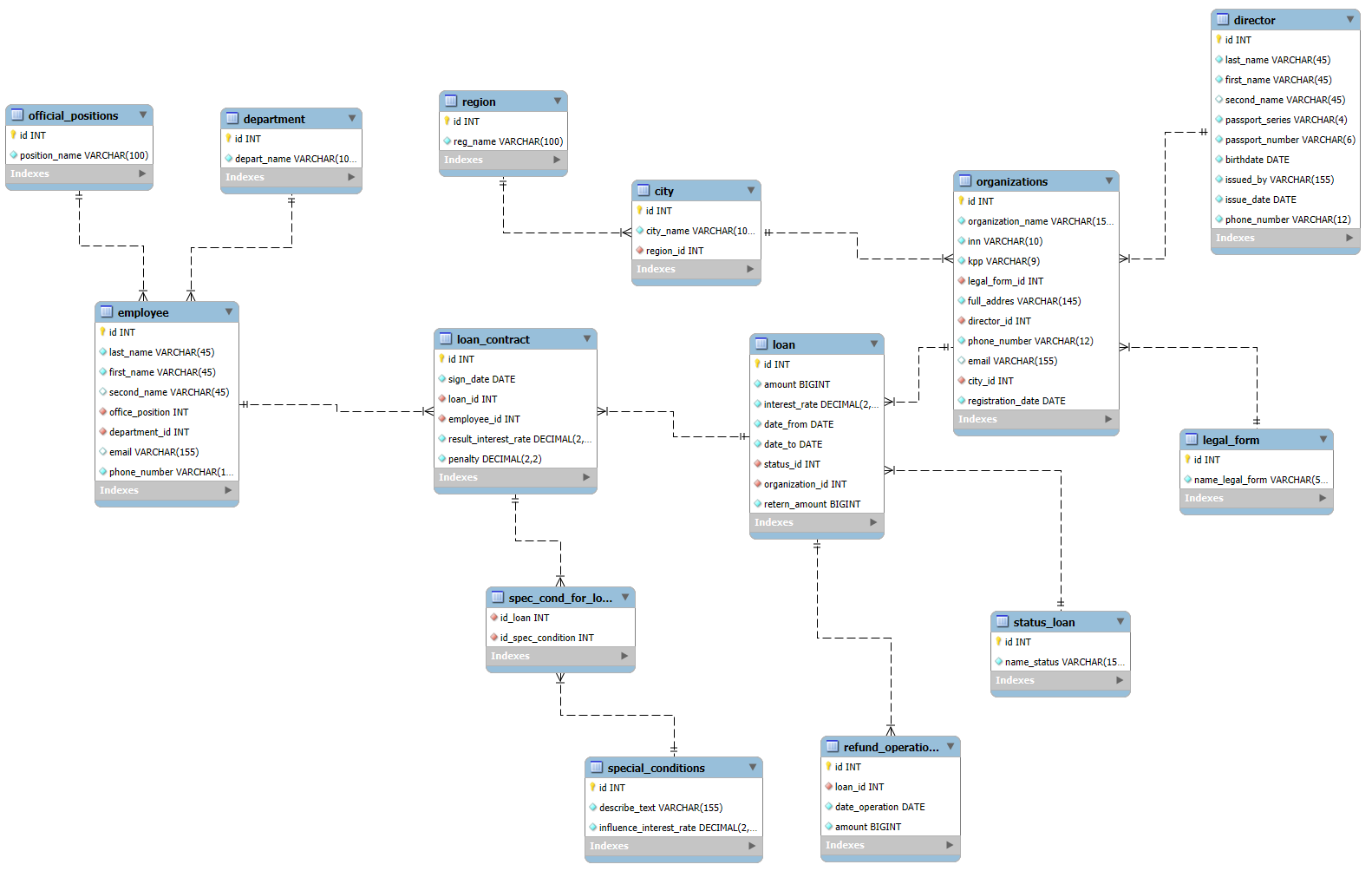
* Отдел
* Должность
* Организационно-правовая форма предприятия
* Статус кредита

Данный шаг поможет устранить дублирование данных, снизить риск возникновения аномалий обновления и удаления, а также обеспечить лучшую структурированность и управляемость данных.

Показатель кардинальности между сущностями «Договор» и «Особые условия» - многие ко многим, что приводит к избыточной сложности и трудностям при управлении данными. Для решения этой проблемы вводится дополнительная таблица, которая будет содержать внешние ключи на обе исходные таблицы. Эта новая таблица называется связующей или ассоциативной таблицей.

### 2.2.3 Окончательная ER-диаграмма

Проведя анализ исходной диаграммы, получили окончательную ER-диаграмму, представленную на рисунке 3 в которой создали хорошо организованную и эффективную структуру базы данных, минимизирующую избыточность данных и обеспечивающую целостность информации.

Рисунок 3 – ER-диаграмма после анализа

### 2.2.3 Описание структуры базы данных

После создание конечной версии базы данных необходимо актуализировать описание сущностей и сопоставить техническое название сущностей и их атрибутов с бизнес описанием. Описание сущностей и атрибутов представлено в таблице 4.

Таблица 4 – Сущности с описанными атрибутами

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Техническое название таблицы | Бизнес название таблицы | Техническое название атрибута | Бизнес описание атрибута | Ключ | Домен | |
| Тип | Размер |
| official\_positions | Должность | id | Идентификатор | PK | INT | 19 |
| position\_name | Название должности |  | VARCHAR | 100 |
| department | Отдел | id | Идентификатор | PK | INT | 19 |
| depart\_name | Название отдела |  | VARCHAR | 100 |

Продолжение таблицы 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Employee | Сотрудник | id | Идентификатор | PK | INT | 19 |
| first\_name | Имя |  | VARCHAR | 45 |
| last\_name | Фамилия |  | VARCHAR | 45 |
| second\_name | Отчество |  | VARCHAR | 45 |
| office\_position | Идентификатор должности | FK | INT | 19 |
| phone\_number | Номер телефона |  | VARCHAR | 12 |
| email | Электронная почта |  | VARCHAR | 155 |
| department\_id | Идентификатор отдела | FK | INT | 19 |
| Region | Область | id | Идентификатор | PK | INT | 19 |
| reg\_name | Название |  | VARCHAR | 100 |
| City | Город | id | Идентификатор | PK | INT | 19 |
| city\_name | Название |  | VARCHAR | 100 |
| reg\_id | Код области | FK | INT | 19 |
| Director | Директор организации | id | Идентификатор | PK | INT | 19 |
| first\_name | Имя |  | VARCHAR | 45 |
| last\_name | Фамилия |  | VARCHAR | 45 |
| second\_name | Отчество |  | VARCHAR | 45 |
| passport\_series | Серия паспорт |  | VARCHAR | 4 |
| passport\_number | Номер паспорт |  | VARCHAR | 6 |
| birthdate | Дата рождения |  | DATE |  |
| issued\_by | Кем выдан |  | VARCHAR | 155 |
| issued\_date | Дата выдачи |  | DATE |  |
| phone\_number | Телефон |  | VARCHAR | 12 |

Продолжение таблицы 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Legal\_form | Организационно-правовая форма предприятия | id | Идентификатор | PK | INT | 19 |
| name\_legal\_form | Название |  | VARCHAR | 155 |
| Organizations | Организация | id | Идентификатор | PK | INT | 19 |
| organization\_name | Название |  | VARCHAR | 155 |
| inn | ИНН |  | INT | 10 |
| kpp | КПП |  | INT | 9 |
| legal\_form\_id | Код организационно-правовой формы | FK | INT | 19 |
| full\_addres | Полный адрес организации |  | VARCHAR | 145 |
| director\_id | Идентификатор директора | FK | INT | 19 |
| phone\_number | Телефон |  | VARCHAR | 12 |
| email | Электронная почта |  | VARCHAR | 155 |
| city\_id | Код города | FK | INT | 19 |
| registration\_date | Дата регистрации организации |  | DATE |  |
| Status\_loan | Статус кредита | id | Идентификатор | PK | INT | 19 |
| name\_status | Статус |  | VARCHAR | 155 |
| Loan | Кредит | id | Идентификатор | PK | INT | 19 |
| amount | Сумма |  | BIGINT |  |
| interest\_rate | Процентная ставка |  | DECIMALL | 2, 2 |
| date\_from | Дата начала кредита |  | DATE |  |
| date\_to | Дата завершения кредита |  | DATE |  |

Продолжение таблицы 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | status\_id | Код статуса кредита | FK | INT | 19 |
| organization\_id | Идентификатор организации | FK | INT | 19 |
| retern\_amount | Возвращенная сумма |  | BIGINT |  |
| Refund\_operations | Операция гашения | id | Идентификатор | PK | INT | 19 |
| loan\_id | Идентификатор кредита | FK | INT | 19 |
| date\_operation | Дата операции |  | DATE |  |
| amount | Сумма |  | BIGINT |  |
| Special\_conditions | Особые условия | id | Идентификатор | PK | INT | 19 |
| describe\_text | Описание |  | VARCHAR | 155 |
| influence\_interest\_rate | Снижение процентной ставки |  | DECIMALL | 2, 2 |
| Loan\_contract | Договор | id | Идентификатор | PK | INT | 19 |
| sign\_date | Дата подписания |  | DATE |  |
| loan\_id | Идентификатор кредита | FK | INT | 19 |
| employee\_id | Идентификатор сотрудника банка | FK | INT | 19 |
| id\_spec\_condition | Особые условия | FK | INT | 19 |
| result\_interest\_rate | Процентная ставка итог |  | DECIMALL | 2, 2 |
| penalty | Пени |  | DECIMALL | 2, 2 |
| Spec\_cond\_for\_loan | Сопоставление спец. условий для КД | id\_loan | Идентификатор кредитного договора | FK | INT | 19 |
| id\_spec\_condition | Идентификатор специального условия | FK | INT | 19 |

## 2.3 Этап физического проектирования

Физическое проектирование базы данных – это процесс создания физической структуры базы данных, которая будет использоваться для хранения информации. Этот этап следует за логическим проектированием, где определяются сущности, их атрибуты и связи между ними. На этапе физического проектирования важно учитывать особенности конкретной СУБД (системы управления базами данных), а также оптимизировать структуру базы данных для повышения производительности.

### 2.3.1 Генерация базы данных

Таблица official\_position предназначена для хранения информации о различных должностях сотрудников банка. Столбец id является уникальным идентификатором должности и автоматически увеличивается. Position\_name – хранит название должности сотрудника длиной до 100 символов и требует обязательного заполнения этого поля. Скрипт создания таблицы official\_position представлен на рисунке 4

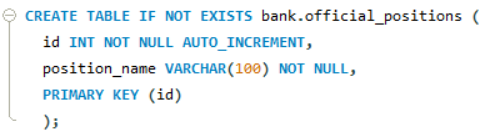


Рисунок 4 – Создание таблицы official\_position

Таблица employee предназначена для хранения информации о сотрудниках банка. Для хранения электронного адреса сотрудника создаем условие с помощью CHECK(email LIKE '%@%.%'), которое проверяет, что введенный адрес соответствует шаблону электронной почты. При создании внешних соединений прописываем инструкции UPDATE и DELETE, чтобы при изменении или удалении записей внешних таблиц соответствующие изменения применялись к таблице с сотрудниками. Полный скрипт создания таблицы employee представлен на рисунке 5

### 

Рисунок 5 – Создание таблицы employee

Для создания ассоциативной таблицы spec\_cond\_for\_loan которая связывает кредиты с их особыми условиями используем скрипт, представленный на рисунке 6.

### 

Рисунок 6 – Создание таблицы spec\_cond\_for\_loan

### Составной первичный ключ устанавливается на основе комбинации значений идентификатора кредитного договора и идентификатора специального условия, что гарантирует уникальность каждой пары (кредит + условие), предотвращая дублирование связей.

### При создании внешних ключе используем инструкция NO ACTION которая означает, что при изменении или удалении записей в родительской таблице никакие действия не будут выполнены над дочерней записью. Это предотвращает автоматическое обновление или удаление связанных данных.

### Таблицы были созданы в соответствии с ER-диаграмма приведенной на рисунке 3, описание сущностей и атрибутов представлено в таблице 4, полный листинг кода создания таблиц приведен в приложении

### 2.3.2 Схема данных в среде выбранной СУБД.

# 3 ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

# ЗАКЛЮЧЕНИЕ

# ПРИЛОЖЕНИЕ

-- -----------------------------------------------------

-- SCHEMA bank

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS bank;

USE bank;

-- -----------------------------------------------------

-- Table bank.official\_positions

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.official\_positions (

id INT NOT NULL AUTO\_INCREMENT,

position\_name VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

);

-- -----------------------------------------------------

-- Table bank.department

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.department(

id INT,

depart\_name VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

);

-- -----------------------------------------------------

-- Table bank.employee

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.employee(

id INT NOT NULL AUTO\_INCREMENT,

last\_name VARCHAR(45) NOT NULL,

first\_name VARCHAR(45) NOT NULL,

second\_name VARCHAR(45) NULL DEFAULT NULL ,

office\_position INT NOT NULL,

department\_id INT NOT NULL,

email VARCHAR(155) NULL DEFAULT NULL CHECK(email LIKE '%@%.%'),

phone\_number VARCHAR(12) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (office\_position)

REFERENCES official\_positions(id)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (department\_id)

REFERENCES department(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

-- -----------------------------------------------------

-- Table bank.region

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.region(

id INT NOT NULL AUTO\_INCREMENT,

reg\_name VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

);

-- -----------------------------------------------------

-- Table bank.city

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.city(

id INT NOT NULL AUTO\_INCREMENT,

city\_name VARCHAR(100) NOT NULL,

region\_id INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (region\_id)

REFERENCES region(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

-- -----------------------------------------------------

-- Table bank.director

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.director(

id INT NOT NULL AUTO\_INCREMENT,

last\_name VARCHAR(45) NOT NULL,

first\_name VARCHAR(45) NOT NULL,

second\_name VARCHAR(45) NULL DEFAULT NULL ,

passport\_series VARCHAR(4) NOT NULL,

passport\_number VARCHAR(6) NOT NULL,

birthdate DATE NOT NULL,

issued\_by VARCHAR(155) NOT NULL,

issue\_date DATE NOT NULL,

phone\_number VARCHAR(12) NOT NULL,

PRIMARY KEY (id)

);

-- -----------------------------------------------------

-- Table bank.legal\_form

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.legal\_form(

id INT NOT NULL AUTO\_INCREMENT,

name\_legal\_form VARCHAR(50) NOT NULL,

PRIMARY KEY (id)

);

-- -----------------------------------------------------

-- Table bank.organization

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.organizations(

id INT NOT NULL AUTO\_INCREMENT,

organization\_name VARCHAR(155) NOT NULL,

inn VARCHAR(10) NOT NULL,

kpp VARCHAR(9) NOT NULL,

legal\_form\_id INT NOT NULL,

full\_addres VARCHAR(145) NOT NULL,

director\_id INT NOT NULL,

phone\_number VARCHAR(12) NOT NULL,

email VARCHAR(155) NULL DEFAULT NULL CHECK(email LIKE '%@%.%'),

city\_id INT NOT NULL,

registration\_date DATE NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (legal\_form\_id)

REFERENCES legal\_form(id)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (director\_id)

REFERENCES director(id)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (city\_id)

REFERENCES city(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

-- -----------------------------------------------------

-- Table bank.status\_loan

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.status\_loan(

id INT NOT NULL AUTO\_INCREMENT,

name\_status VARCHAR(155) NOT NULL,

PRIMARY KEY (id)

);

-- -----------------------------------------------------

-- Table bank.loan

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.loan(

id INT NOT NULL AUTO\_INCREMENT,

amount BIGINT NOT NULL,

interest\_rate DECIMAL(2, 2) NOT NULL,

date\_from DATE NOT NULL,

date\_to DATE NOT NULL,

status\_id INT NOT NULL,

organization\_id INT NOT NULL,

retern\_amount BIGINT NOT NULL DEFAULT 0,

PRIMARY KEY (id),

FOREIGN KEY (status\_id)

REFERENCES status\_loan(id)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (organization\_id)

REFERENCES organizations(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

-- -----------------------------------------------------

-- Table bank.special\_conditions

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.special\_conditions(

id INT NOT NULL AUTO\_INCREMENT,

describe\_text VARCHAR(155) NOT NULL,

influence\_interest\_rate DECIMAL(2, 2) NOT NULL DEFAULT 0,

PRIMARY KEY (id)

);

-- -----------------------------------------------------

-- Table bank.loan\_contract

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.loan\_contract(

id INT NOT NULL AUTO\_INCREMENT,

sign\_date DATE NOT NULL,

loan\_id INT NOT NULL,

employee\_id INT NOT NULL,

result\_interest\_rate DECIMAL(2, 2) NOT NULL,

penalty DECIMAL(2, 2) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (loan\_id)

REFERENCES loan(id)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (employee\_id)

REFERENCES employee(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

-- -----------------------------------------------------

-- Table bank.spec\_cond\_for\_loan

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.spec\_cond\_for\_loan(

id\_loan INT NOT NULL,

id\_spec\_condition INT NOT NULL,

PRIMARY KEY (id\_loan, id\_spec\_condition),

FOREIGN KEY (id\_spec\_condition)

REFERENCES special\_conditions(id)

ON UPDATE NO ACTION

ON DELETE NO ACTION,

FOREIGN KEY (id\_loan)

REFERENCES loan\_contract(id)

ON UPDATE NO ACTION

ON DELETE NO ACTION

);

-- -----------------------------------------------------

-- Table bank.refund\_operations

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS bank.refund\_operations(

id INT NOT NULL AUTO\_INCREMENT,

loan\_id INT NOT NULL,

date\_operation DATE NOT NULL,

amount BIGINT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (loan\_id)

REFERENCES loan(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ