**Санкт-Петербургский Государственный Университет**

**Экономический факультет**

**Кафедра информационных систем в экономике**

**ОТЧЕТ О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ**

**по направлению «Бизнес-информатика»**

**ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ РЕСТОРАНА**

**Выполнил:**

студент 2 курса

Группы Б14-э

Сорвин егор Денисович

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

**Руководитель практики от кафедры:**

Кандидат психологических наук, доцент Кафедры информационных систем в экономике

Кадура Елена Вячеславовна

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Санкт-Петербург

2023

Оглавление

[Глава 1. Анализ экономической предметной области проэктирования 2](#_Toc135886156)

[1.1 Описание области проектирования 2](#_Toc135886157)

[1.2 Информационные потребности 3](#_Toc135886158)

[1.3 Ограничение предметной области 5](#_Toc135886159)

[Глава 2.Проектирование базы данных 6](#_Toc135886160)

[2.1 Анализ входной и выходной информации 6](#_Toc135886161)

[2.2 Моделирование данных 7](#_Toc135886162)

[2.3 Нормализация информационных объектов 8](#_Toc135886163)

[2.4 Построение информационно-логической модели 11](#_Toc135886164)

[2.5 Описание используемых типов данных 13](#_Toc135886165)

[Глава 3. Машинная реализация базы данных 16](#_Toc135886166)

[3.1 Формирование таблиц в MySQL 16](#_Toc135886167)

[3.2 Разработка представлений 19](#_Toc135886168)

[3.3 Разработка процедур 24](#_Toc135886169)

[3.4 Разработка триггеров 26](#_Toc135886170)

[3.5 Разработка функций 28](#_Toc135886171)

[Заключение 31](#_Toc135886172)

**Введение**

Цель работы

Целью курсовой работы является разработка базы данных для информационной поддержки деятельности базы отдыха

Задачи

1. Анализ основных процессов деятельности базы отдыха, выделение организационно-штатной структуры, ролей и их функций.
2. Структуризация ограничений предметной области, влияющих на состав объектов базы данных.
3. Определение требований к системе через информационные потребности пользователей.
4. Анализ информационных потоков и выделение информационных объектов
5. Разработка модели данных.
6. Реализация проекта в среде СУБД MS ACCESS и создание пользовательского приложения.

# Глава 1. Анализ экономической предметной области проектирования

## 1.1 Описание области проектирования

База отдыха - это место для отдыха и развлечений, которое может включать в себя различные виды активного и пассивного отдыха, например, пляж, спортивные площадки, аренда развлекательного инвентаря, посещение бани. База отдыха может располагаться как в городе, так и в пригороде, а также на природе.

Персонал:

* Директор базы отдыха (1 сотрудник): отвечает за общее управление базой отдыха, планирует и контролирует работу персонала и бизнес-процессы.
* Менеджер по работе с клиентами (1 сотрудник): общается с клиентами, бронирует номера и услуги, отвечает на вопросы, решает проблемы.
* Администратор (1 сотрудник): занимается приемом гостей, регистрацией заезда и выезда, обеспечивает комфортное пребывание гостей на базе отдыха.
* Менеджер по продажам (1 сотрудник): занимается продажей услуг базы отдыха, разрабатывает маркетинговые стратегии и рекламные кампании.
* Менеджер по работе с персоналом (1 сотрудник): занимается наймом и обучением персонала, решает вопросы, связанные с трудовыми отношениями.
* Работник склада (1 сотрудник): занимается выдачей оборудования в аренду, заведует складом, следит за состоянием оборудования.
* Уборщики (2-3 сотрудника): убирают номера, общие зоны и территорию базы отдыха.
* Банщик (1 Сотрудник): растапливает баню, работает с парящимися людьми, отслеживая их состояние, парит, делает оздоровительную мойку и массаж, обертывание, притирание.

Характеристика организационной структуры базы отдыха:

* Администрация магазина: Директор базы отдыха, Администратор, Менеджер по

закупкам, Менеджер по работе с персоналом

* Продажи: Менеджер по работе с клиентами, Менеджер по продажам
* Склад: Работник склада.
* Обслуживающий персонал: Уборщики, Банщик

## 1.2 Информационные потребности

Информационные потребности для базы отдыха могут включать:

1. Директор базы отдыха:

* Информация о финансовых показателях базы отдыха (выручка, расходы, прибыль и т.д.);
* Данные о занятости номерного фонда;
* Информация о различных мероприятиях и мероприятиях на базе отдыха;
* Данные о работе персонала и их эффективности;
* Информация о текущих проблемах и предложениях для улучшения базы отдыха.

1. Администратор:

* Информация о бронировании номеров и доступности;
* Данные о гостях (имена, даты пребывания, дополнительные требования и т.д.);
* Информация о забронированных мероприятиях и их деталях;
* Данные о платежах и счетах гостей;
* Информация о работе персонала и их графиках.

1. Менеджер по работе с персоналом:

* Информация о кадровых вопросах, включая найм, увольнение и обучение;
* Данные о зарплатах, бонусах и других выплатах персоналу;
* Информация о рабочих графиках и графиках отпусков;
* Данные о работе и производительности персонала.

1. Менеджер по работе с клиентами:

* Информация о запросах и жалобах от гостей;
* Данные о желаниях и потребностях гостей;
* Информация о предпочтениях гостей и настройках номеров;
* Данные о мероприятиях, которые могут быть интересны для гостей.

1. Менеджер по продажам:

* Информация о тенденциях и предпочтениях рынка отдыха;
* Данные о конкурентах и их стратегиях продаж;
* Информация о потребностях и предпочтениях гостей;
* Данные о продуктах и услугах, которые продаются на базе отдыха.

1. Работник склада:

* Информация о приходе товаров на склад и о расходе товаров со склада;
* Данные о количестве товара на складе и его местонахождении;
* Информация о сроках годности и условиях хранения товаров;
* Данные о состоянии оборудования на складе;
* Информация о требованиях по безопасности и охране труда при работе на складе;
* Данные о ведении учета и отчетности по движению товаров на складе;
* Информация о планировании и проведении инвентаризации.

1. Уборщики:

* Информация о расписании уборки помещений на базе отдыха;
* Данные о необходимом инвентаре и средствах для уборки;
* Информация о правилах и требованиях по безопасности и охране труда при уборке;
* Данные о качественных и количественных показателях работы уборщиков;
* Информация о необходимых ремонтных работах в помещениях на базе отдыха;
* Данные о планировании и проведении специальных мероприятий на базе отдыха и их влиянии на график уборки;
* Информация о средствах индивидуальной защиты при работе с химическими и другими вредными веществами.

1. Банщик

* Записи о бронированиях
* Информация о гостях.
* Инвентарь и расходные материалы
* Инструкции и правила
* Заявки и запросы гостей
* Статус уборки
* Информация о клиентах

## 1.3 Ограничение предметной области

Ограничение предметной области для базы отдыха:

• Категории услуг: проживание на базе, аренда оборудования (инвентаря), банные процедуры.

• Каждая услуга имеет уникальный идентификатор, название, описание, цену и доступность на определенные даты.

• Услуги бронирования предоставляются на единовременно и для всех объектов бронирования.

• Гости могут бронировать одну или несколько услуг на определенные даты.

• У каждого бронирования есть уникальный идентификатор, дата и время бронирования, список забронированных услуг, общая стоимость бронирования и статус бронирования (например, ожидание оплаты, обработка, завершено).

• Гости могут выбирать различные варианты оплаты (например, наличные, банковская карта, электронные платежи).

• Сотрудники базы отдыха могут добавлять, изменять и удалять услуги из каталога, а также обрабатывать бронирования и управлять статусами бронирований.

• База отдыха может предоставлять услуги консультаций гостям о выборе оптимальных услуг в зависимости от их целей и потребностей.

• Отчеты о продажах и прибыли генерируются на основе данных о бронированиях и стоимости услуг.

• Данные о гостях и бронированиях защищены конфиденциальностью и хранятся в базе данных с доступом только для авторизованных пользователей.

Документооборот:

|  |  |  |
| --- | --- | --- |
| **Процесс** | **Ответственный персонал** | **Оперативный документ** |
| Разработка бизнес-плана | Директор базы отдыха | Бизнес-план |
| Определение ассортимента услуг | Директор базы отдыха | Каталог услуг |
| Набор персонала | Менеджер по работе с персоналом | Табель учета рабочего времени, договоры с сотрудниками, трудовые книжки |
| Обучение персонала | Менеджер по работе с персоналом | Программы обучения, учебные планы |
| Планирование мероприятий | Менеджер по работе с клиентами | Календарь мероприятий |
| Продажа услуг | Менеджер по продажам | Договор на оказание услуг, кассовый чек |
| Оформление заказов на услуги | Менеджер по работе с клиентами | Заявка на бронирование услуг |
| Прием и хранение товаров на складе | Работник склада | Товарно-транспортная накладная |
| Подготовка оборудования и инвентаря для мероприятий | Работник склада | Список оборудования и инвентаря |
| Уборка помещений и территории | Уборщики | Технологические карты уборки, акты об уборке |
| Контроль качества услуг и товаров | Менеджер по работе с клиентами | Анкеты удовлетворенности клиентов |
| Решение проблем клиентов | Менеджер по работе с клиентами | Жалобы и предложения клиентов, акты о неудовлетворительном качестве услуг или товаров |
| Проведение банных мероприятий, растопка | Банщик | Расписание аренды бань |

# Глава 2.Проектирование базы данных

## 2.1 Анализ входной и выходной информации

|  |  |
| --- | --- |
| **Наименование справочника** | **Реквизиты** |
| Справочник сотрудников | Номер сотрудника, ФИО, Дата рождения, Домашний адрес, Контактный телефон, Образование, Категория |
| Справочник клиентов | ID клиента, ФИО, Дата рождения, Адрес, Контактный телефон, Электронная почта |
| Справочник объектов | Номер номера, Тип номера, Количество комнат, Вместимость, Цена за ночь |
| Справочник услуг | Номер услуги, Наименование, Описание, Цена |
| Справочник бронирования | Номер брони, Номер клиента, Номер номера, Дата заезда, Дата выезда |
| Справочник поставщиков | Номер поставщика, Наименование, Контактное лицо, Адрес, Контактный телефон |
| Справочник отзывов | Номер отзыва, Номер клиента, Номер номера, Дата, Текст отзыва, Оценка |

Выходная информация классифицируется следующим образом:

Стандартные отчеты:

1. Отчет о загрузке объектов за период времени

2. Отчет о выручке от аренды оборудования за период времени

3. Отчет о количестве бронирований на определенный период времени

Данные для сотрудников:

1. Расписание уборки номеров

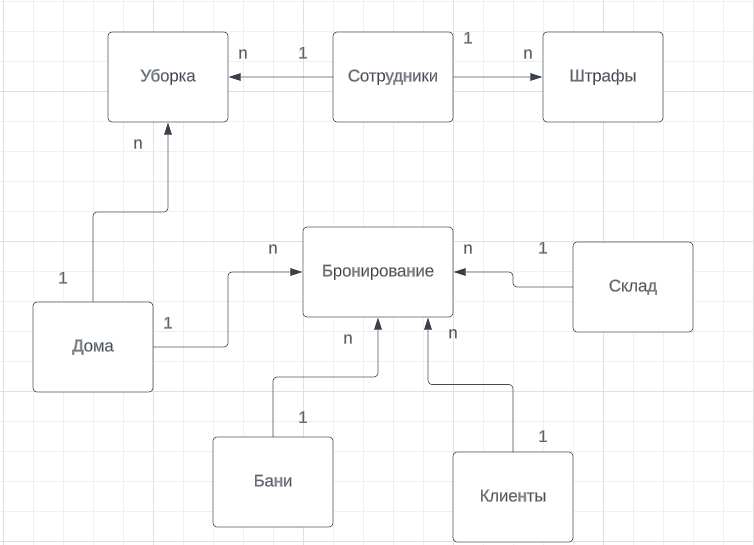
2. Расписание работы персонала

Данные для работы с клиентами:

1. Счет для оплаты проживания и дополнительных услуг

## 2.2 Моделирование данных

|  |  |
| --- | --- |
| **Информационный объект** | **Атрибуты** |
| Дома | Номер дома, Количество комнат, Количество спален, Количество ванных комнат, Цена за ночь, Дата доступности |
| Склад | Название позиции, Описание, Цена за день аренды ,Дата доступности |
| Бронирование | Номер брони , Дата бронирования, Дата начала аренды, Дата окончания аренды , Сумма оплаты , Статус оплаты, Номер клиента ,Номер дома, Название позиции на складе, Номер бани |
| Клиенты | Номер клиента, ФИО, Адрес, Телефон, Электронная почта |
| Сотрудники | Код сотрудника, ФИО, Дата рождения, Адрес, Телефон, Образование, Должность |
| Штрафы | Номер штрафа, Описание штрафа, Сумма штрафа, Код сотрудника |
| Уборка | Номер уборки, Дата уборки, Сотрудник , Номер дома |
| Бани | Номер бани, Количество комнат, Количество спален, Количество ванных комнат, Цена за ночь, Дата доступности |



* 1. Нормализация информационных объектов

Дома

• Номер\_дома

• Количество\_комнат

• Количество\_спален

• Количество\_ванных\_комнат

• Цена\_за\_ночь

• Дата\_доступности

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

Склад

• Название\_позиции

• Описание

• Цена\_за\_день\_аренды

• Дата\_доступности

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

Бронирование

• Номер\_брони

• Дата\_бронирования

• Дата\_начала\_аренды

• Дата\_окончания\_аренды

• Сумма\_оплаты

• Статус\_оплаты

• Номер\_клиента

• Номер\_дома

• Название\_позиции\_на\_складе

• Номер\_бани

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

Клиенты

• Номер\_клиента

• ФИО

• Адрес

• Телефон

• Электронная\_почта

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

Сотрудники

• Код\_сотрудника

• ФИО

• Дата\_рождения

• Адрес

• Телефон

• Образование

• Должность

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

Уборка

• Номер\_уборки

• Дата\_уборки

• Сотрудник

• Номер\_дома

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

Бани

• Номер\_бани

• Количество\_комнат

• Количество\_спален

• Количество\_ванных\_комнат

• Цена\_за\_ночь

• Дата\_доступности

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

Штрафы

• Номер\_штрафа

• Описание\_штрафа

• Сумма\_штрафа

• Код\_сотрудника

Информативный объект, имеет структуру данных, соответствующую требованиям третьей нормальной формы (3 НФ). Это означает, что каждый атрибут объекта зависит только от единственного ключа, а не от других атрибутов. Все связи между данными внутри объекта были корректно установлены и соблюдены.

2.4 Построение информационно-логической модели

1. Один дом может иметь множество бронирований. Связь один ко многим. Связующее поле – **Номер\_дома.**

Бронирование

Дома

Сотрудники

1. Одна баня может иметь множество бронирований. Связь один ко многим. Связующее поле – **Номер\_бани.**

Бронирование

Бани

Сотрудники

1. Один клиент может сделать множество бронирований. Связь один ко многим. Связующее поле – **Номер\_клиента.**

Бронирование

Клиенты

Сотрудники

1. Один сотрудник может иметь множество штрафов. Связь один ко многим. Связующее поле – **Код\_сотрудника.**

Сотрудники

Сотрудники

Штрафы

1. Один сотрудник может быть выполнять множество уборок. Связь один ко многим. Связующее поле – **Код\_сотрудника.**

Уборки

Сотрудники

Сотрудники

1. Одна позиция со склада может иметь множество бронирований. Связь один ко многим. Связующее поле – **Название\_позиции\_на\_складе.**

Бронирование

Склад

Сотрудники

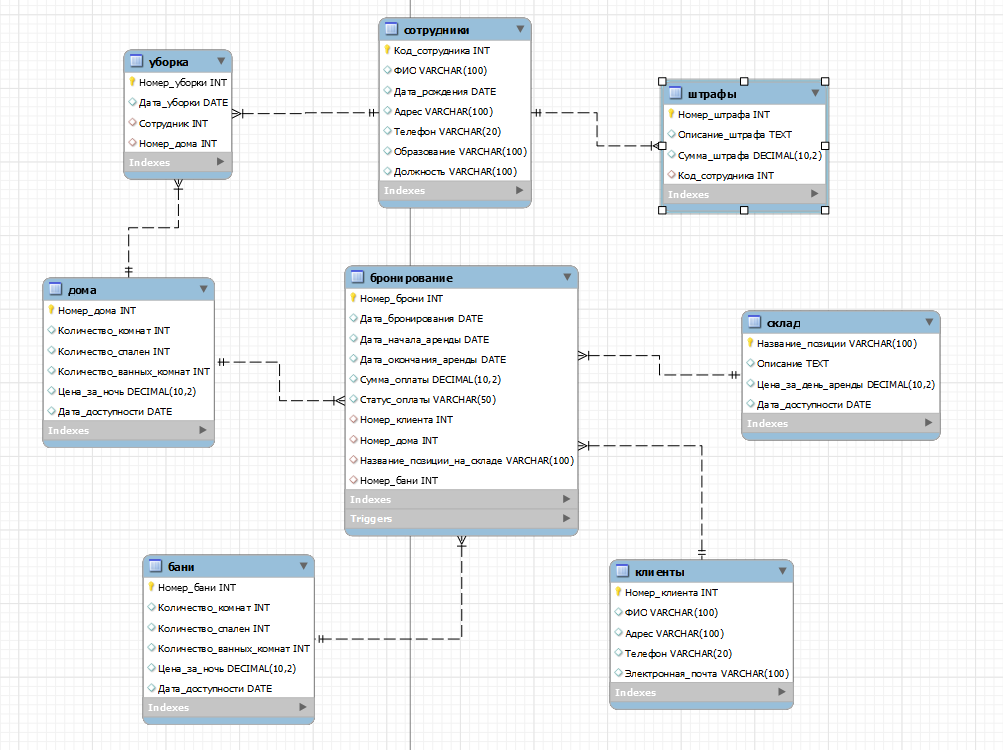
1. Один дом убирается много раз. Связь один ко многим. Связующее поле – **Номер\_дома.**

Уборки

Дома

Сотрудники

Построим EER- диаграмму в MySQL Workbench



2.5 Описание используемых типов данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Информационный объект** | **Атрибуты** | **Объект в MySQL** | **Атрибуты и значения** |
| Дома | Номер\_дома, Количество\_комнат, Количество\_спален, Количество\_ванных\_комнат, Цена\_за\_ночь, Дата\_доступности | Дома | Номер\_дома INT (PK), Количество\_комнат INT, Количество\_спален INT, Количество\_ванных\_комнат INT, Цена\_за\_ночь DECIMAL(10, 2), Дата\_доступности DATE |
| Бани | Номер\_бани, Количество\_комнат, Количество\_спален, Количество\_ванных\_комнат, Цена\_за\_ночь, Дата\_доступности | Бани | Номер\_бани INT (PK), Количество\_комнат INT, Количество\_спален INT, Количество\_ванных\_комнат INT, Цена\_за\_ночь DECIMAL(10, 2), Дата\_доступности DATE |
| Склад | Название\_позиции, Описание, Цена\_за\_день\_аренды, Дата\_доступности | Склад | Название\_позиции VARCHAR(100) (PK), Описание TEXT, Цена\_за\_день\_аренды DECIMAL(10, 2), Дата\_доступности DATE |
| Бронирование | ID\_бронирования, Номер\_дома, Номер\_бани, Название\_позиции, Дата\_начала, Дата\_окончания, Сумма | Бронирование | ID\_бронирования INT (PK), Номер\_дома INT, Номер\_бани INT, Название\_позиции VARCHAR(100), Дата\_начала DATE, Дата\_окончания DATE, Сумма DECIMAL(10, 2) |
| Сотрудники | Номер\_сотрудника, ФИО, Должность, Номер\_телефона, Электронная\_почта | Сотрудники | Номер\_сотрудника INT (PK), ФИО VARCHAR(255), Должность VARCHAR(255), Номер\_телефона VARCHAR(20), Электронная\_почта VARCHAR(255) |

|  |  |  |  |
| --- | --- | --- | --- |
| Уборки | ID\_уборки, Дата\_уборки, Номер\_дома, Номер\_бани | Уборки | ID\_уборки INT (PK), Дата\_уборки DATE, Номер\_дома INT, Номер\_бани INT |
| Штрафы | ID\_штрафа, ID\_бронирования, Сумма, Причина | Штрафы | ID\_штрафа INT (PK), ID\_бронирования INT, Сумма DECIMAL(10, 2), Причина TEXT |
| Клиенты | ID\_клиента, ФИО, Номер\_телефона, Электронная\_почта | Клиенты | ID\_клиента INT (PK), ФИО VARCHAR(255), Номер\_телефона VARCHAR(20), Электронная\_почта VARCHAR(255) |

# Глава 3. Машинная реализация базы данных

3.1 Формирование таблиц в MySQL

В соответствии с разработанной ИЛМ были созданы следующие таблицы: Дома, Бани, Склад, Сотрудники, Клиенты, Бронирование, Штрафы, Уборки.

CREATE TABLE Дома (

Номер\_дома INT PRIMARY KEY,

Количество\_комнат INT,

Количество\_спален INT,

Количество\_ванных\_комнат INT,

Цена\_за\_ночь DECIMAL(10, 2),

Дата\_доступности DATE

);

-- Создание таблицы "Склад"

CREATE TABLE Склад (

Название\_позиции VARCHAR(100) PRIMARY KEY,

Описание TEXT,

Цена\_за\_день\_аренды DECIMAL(10, 2),

Дата\_доступности DATE

);

-- Создание таблицы "Бронирование"

CREATE TABLE Бронирование (

Номер\_брони INT PRIMARY KEY,

Дата\_бронирования DATE,

Дата\_начала\_аренды DATE,

Дата\_окончания\_аренды DATE,

Сумма\_оплаты DECIMAL(10, 2),

Статус\_оплаты VARCHAR(50),

Номер\_клиента INT,

Номер\_дома INT,

Название\_позиции\_на\_складе VARCHAR(100),

Номер\_бани INT,

FOREIGN KEY (Номер\_клиента) REFERENCES Клиенты(Номер\_клиента),

FOREIGN KEY (Номер\_дома) REFERENCES Дома(Номер\_дома),

FOREIGN KEY (Номер\_бани) REFERENCES Бани(Номер\_бани),

FOREIGN KEY (Название\_позиции\_на\_складе) REFERENCES Склад(Название\_позиции)

);

-- Создание таблицы "Клиенты"

CREATE TABLE Клиенты (

Номер\_клиента INT PRIMARY KEY,

ФИО VARCHAR(100),

Адрес VARCHAR(100),

Телефон VARCHAR(20),

Электронная\_почта VARCHAR(100)

);

-- Создание таблицы "Сотрудники"

CREATE TABLE Сотрудники (

Код\_сотрудника INT PRIMARY KEY,

ФИО VARCHAR(100),

Дата\_рождения DATE,

Адрес VARCHAR(100),

Телефон VARCHAR(20),

Образование VARCHAR(100),

Должность VARCHAR(100)

);

-- Создание таблицы "Уборка"

CREATE TABLE Уборка (

Номер\_уборки INT PRIMARY KEY,

Дата\_уборки DATE,

Сотрудник INT,

Номер\_дома INT,

FOREIGN KEY (Сотрудник) REFERENCES Сотрудники(Код\_сотрудника),

FOREIGN KEY (Номер\_дома) REFERENCES Дома(Номер\_дома)

);

-- Создание таблицы "Бани"

CREATE TABLE Бани (

Номер\_бани INT PRIMARY KEY,

Количество\_комнат INT,

Количество\_спален INT,

Количество\_ванных\_комнат INT,

Цена\_за\_ночь DECIMAL(10, 2),

Дата\_доступности DATE,

);

-- Создание таблицы "Штрафы"

CREATE TABLE Штрафы (

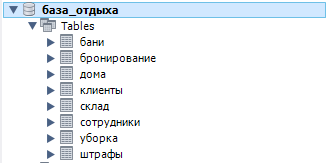
Номер\_штрафа INT PRIMARY KEY,

Описание\_штрафа TEXT,

Сумма\_штрафа DECIMAL(10, 2),

Код\_сотрудника INT,

FOREIGN KEY (Код\_сотрудника) REFERENCES Сотрудники(Код\_сотрудника)



3.2 Разработка представлений

Представление 1 "доступные\_объекты" показывает список доступных объектов базы отдыха (домов, бань и склада) с указанием их идентификатора, типа объекта и даты доступности. Оно используется для отображения объектов, которые в данный момент доступны для аренды, исключая объекты, занятые в текущий период.

В представлении используется объединение (UNION) трех запросов. Первый запрос выбирает дома, удовлетворяющие условию даты доступности и не занятые в текущий период. Второй запрос выбирает бани с аналогичными условиями. Третий запрос выбирает позиции со склада, также удовлетворяющие условиям даты доступности и не занятые в текущий период.

Каждый выбранный объект имеет три атрибута: "Идентификатор" - номер дома, бани или название позиции на складе, "Тип\_объекта" - указывает тип объекта ("Дом", "Баня" или "Склад"), "Дата\_доступности" - дата, начиная с которой объект доступен для аренды.

Представление "доступные\_объекты" полезно для информирования пользователей о доступных объектах базы отдыха и их дате доступности, что помогает им выбрать подходящий объект для аренды.

CREATE VIEW Доступные\_объекты AS

SELECT

Д.Номер\_дома AS Идентификатор,

'Дом' AS Тип\_объекта,

Д.Дата\_доступности AS Дата\_доступности

FROM

Дома Д

WHERE

Д.Дата\_доступности <= CURDATE()

AND Д.Номер\_дома <> 0

AND Д.Номер\_дома NOT IN (SELECT Номер\_дома FROM Бронирование WHERE CURDATE() BETWEEN Дата\_начала\_аренды AND Дата\_окончания\_аренды)

UNION

SELECT

Б.Номер\_бани AS Идентификатор,

'Баня' AS Тип\_объекта,

Б.Дата\_доступности AS Дата\_доступности

FROM

Бани Б

WHERE

Б.Дата\_доступности <= CURDATE()

AND Б.Номер\_бани <> 0

AND Б.Номер\_бани NOT IN (SELECT Номер\_бани FROM Бронирование WHERE CURDATE() BETWEEN Дата\_начала\_аренды AND Дата\_окончания\_аренды)

UNION

SELECT

С.Название\_позиции AS Идентификатор,

'Склад' AS Тип\_объекта,

С.Дата\_доступности AS Дата\_доступности

FROM

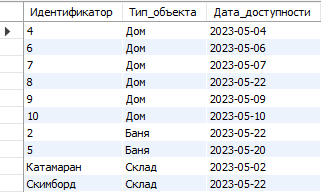
Склад С

WHERE

С.Дата\_доступности <= CURDATE()

AND С.Название\_позиции <> '0'

AND С.Название\_позиции NOT IN (SELECT Название\_позиции\_на\_складе FROM Бронирование WHERE CURDATE() BETWEEN Дата\_начала\_аренды AND Дата\_окончания\_аренды);



Представление 2 `доступныепозициинаскладе` выдаёт нам информацию о доступных для аренды на данный момент позициях склада.

CREATE VIEW ДоступныеПозицииНаСкладе AS

SELECT Название\_позиции, Описание, Цена\_за\_день\_аренды, Дата\_доступности

FROM Склад

WHERE Дата\_доступности <= CURDATE();

-- СтатистикаПоКлиентам

CREATE VIEW СтатистикаПоКлиентам AS

SELECT

К.Номер\_клиента AS Идентификатор\_клиента,

К.ФИО AS Клиент,

COUNT(Б.Номер\_брони) AS Количество\_бронирований,

SUM(Б.Сумма\_оплаты) AS Общая\_сумма\_оплаты

FROM

Клиенты К

LEFT JOIN Бронирование Б ON К.Номер\_клиента = Б.Номер\_клиента

GROUP BY

Идентификатор\_клиента;



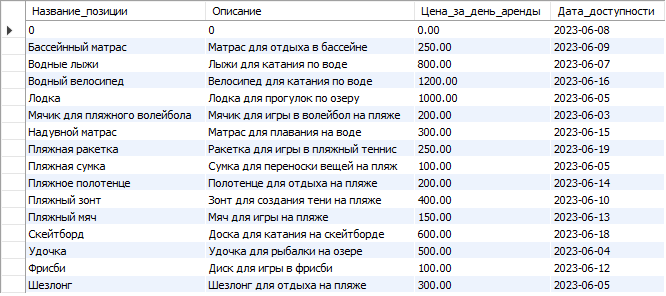
Представление 3 `занятыепозициинаскладе` выдаёт информацию о арендованных на данный момент позициях склада. Это может понадобиться для того, чтобы понять когда они освободятся.

CREATE VIEW ЗанятыеПозицииНаСкладе AS

SELECT Название\_позиции, Описание, Цена\_за\_день\_аренды, Дата\_доступности

FROM Склад

WHERE Дата\_доступности > CURDATE();



Преставление 4 `статистикапоклиентам` выдаёт статистику по каждому клиенту, а именно сколько бронирований он совершил и на какую общую сумму.

CREATE VIEW СтатистикаПоКлиентам AS

SELECT

К.Номер\_клиента AS Идентификатор\_клиента,

К.ФИО AS Клиент,

COUNT(Б.Номер\_брони) AS Количество\_бронирований,

SUM(Б.Сумма\_оплаты) AS Общая\_сумма\_оплаты

FROM

Клиенты К

LEFT JOIN Бронирование Б ON К.Номер\_клиента = Б.Номер\_клиента

GROUP BY

Идентификатор\_клиента;



Представление 5 `статистикапосотрудникам` выводит информацию о сотрудниках, а именно сколько уборок провели уборщики и сколько штрафов получили работники в целом.

CREATE VIEW СтатистикаПоСотрудникам AS

SELECT

С.Код\_сотрудника AS Идентификатор\_сотрудника,

С.ФИО AS Сотрудник,

COUNT(У.Номер\_уборки) AS Количество\_уборок,

COUNT(Ш.Номер\_штрафа) AS Количество\_штрафов

FROM

Сотрудники С

LEFT JOIN Уборка У ON С.Код\_сотрудника = У.Сотрудник

LEFT JOIN Штрафы Ш ON С.Код\_сотрудника = Ш.Код\_сотрудника

GROUP BY

Идентификатор\_сотрудника;



Представление 6 `суммарнаяприбыль` предоставляет информацию о суммарной прибыли по месяцам и годам.

CREATE VIEW СуммарнаяПрибыль AS

SELECT

YEAR(Б.Дата\_бронирования) AS Год,

MONTH(Б.Дата\_бронирования) AS Месяц,

SUM(Б.Сумма\_оплаты) AS Общая\_прибыль

FROM

Бронирование Б

GROUP BY

Год, Месяц;



3.3 Разработка процедур

Процедура 1 `ДобавитьБронирование` добавляет запись о бронировании в таблицу Бронирование.

CREATE PROCEDURE ДобавитьБронирование(

IN p\_Номер\_брони INT,

IN p\_Дата\_начала\_аренды DATE,

IN p\_Дата\_окончания\_аренды DATE,

IN p\_Сумма\_оплаты DECIMAL(10, 2),

IN p\_Статус\_оплаты VARCHAR(20),

IN p\_Номер\_клиента INT,

IN p\_Номер\_дома INT,

IN p\_Номер\_бани INT,

IN p\_Название\_позиции\_на\_складе VARCHAR(100)

)

BEGIN

INSERT INTO Бронирование (Номер\_брони, Дата\_бронирования, Дата\_начала\_аренды, Дата\_окончания\_аренды, Сумма\_оплаты, Статус\_оплаты, Номер\_клиента, Номер\_дома, Номер\_бани, Название\_позиции\_на\_складе)

VALUES (p\_Номер\_брони, p\_Дата\_бронирования, p\_Дата\_начала\_аренды, p\_Дата\_окончания\_аренды, p\_Сумма\_оплаты, p\_Статус\_оплаты, p\_Номер\_клиента, p\_Номер\_дома, p\_Номер\_бани, p\_Название\_позиции\_на\_складе);

END //

Процедура 2 `ДобавитьПозициюНаСкладе`` добавляет запись о новом предмете доступном для аренды в таблицу Склад.

CREATE PROCEDURE ДобавитьПозициюНаСкладе (

IN Название\_позиции VARCHAR(100),

IN Описание TEXT,

IN Цена\_за\_день\_аренды DECIMAL(10, 2),

IN Дата\_доступности DATE

)

BEGIN

INSERT INTO Склад (Название\_позиции, Описание, Цена\_за\_день\_аренды, Дата\_доступности)

VALUES (Название\_позиции, Описание, Цена\_за\_день\_аренды, Дата\_доступности);

END //

Процедура 3 `ДобавитьСотрудника` добавляет запись о новом сотруднике в таблицу Сотрудники.

CREATE PROCEDURE ДобавитьСотрудника (

IN ФИО VARCHAR(100),

IN Дата\_рождения DATE,

IN Адрес VARCHAR(100),

IN Телефон VARCHAR(20),

IN Образование VARCHAR(100),

IN Должность VARCHAR(100)

)

BEGIN

INSERT INTO Сотрудники (ФИО, Дата\_рождения, Адрес, Телефон, Образование, Должность)

VALUES (ФИО, Дата\_рождения, Адрес, Телефон, Образование, Должность);

END //

Процедура 4 `ДобавитьУборку` добавляет запись о запланированной уборке в таблицу Уборка.

CREATE PROCEDURE ДобавитьУборку (

IN Дата\_уборки DATE,

IN Сотрудник INT,

IN Номер\_дома INT

)

BEGIN

INSERT INTO Уборка (Дата\_уборки, Сотрудник, Номер\_дома)

VALUES (Дата\_уборки, Сотрудник, Номер\_дома);

END //

Процедура 5 `ДобавитьШтраф` добавляет запись о новом штрафе конкретному сотруднику в таблицу Штрафы.

CREATE PROCEDURE ДобавитьШтраф (

IN Описание\_штрафа TEXT,

IN Сумма\_штрафа DECIMAL(10, 2),

IN Код\_сотрудника INT

)

BEGIN

INSERT INTO Штрафы (Описание\_штрафа, Сумма\_штрафа, Код\_сотрудника)

VALUES (Описание\_штрафа, Сумма\_штрафа, Код\_сотрудника);

END //

3.4 Разработка триггеров

В процессе разработки базы данных возникла потребность в создании триггеров, которые автоматически обновляют поля в таблицах, используя определенные хранимые процедуры. Триггеры позволяют реагировать на изменения данных и выполнять дополнительные операции, такие как вставка, обновление или удаление записей в других таблицах. Они предоставляют механизм для поддержки целостности данных и автоматизации определенных действий при изменении данных в базе данных.

Триггер 1 `подсчет\_суммы\_оплаты` автоматически подсчитывает общую стоимость бронирования после добавление записи в таблицу Бронирование.

CREATE DEFINER=`root`@`localhost` TRIGGER `подсчет\_суммы\_оплаты` BEFORE INSERT ON `бронирование` FOR EACH ROW BEGIN

DECLARE house\_price DECIMAL(10, 2);

DECLARE bathhouse\_price DECIMAL(10, 2);

DECLARE inventory\_price DECIMAL(10, 2);

DECLARE rental\_days INT;

DECLARE total\_sum DECIMAL(10, 2);

-- Получение цены дома

SET house\_price = (

SELECT Цена\_за\_ночь

FROM Дома

WHERE Номер\_дома = NEW.Номер\_дома

);

-- Получение цены бани

SET bathhouse\_price = (

SELECT Цена\_за\_ночь

FROM Бани

WHERE Номер\_бани = NEW.Номер\_бани

);

-- Получение суммы цен арендованных предметов

SET inventory\_price = (

SELECT SUM(Цена\_за\_день\_аренды)

FROM Склад

WHERE Название\_позиции = NEW.Название\_позиции\_на\_складе

);

-- Вычисление количества дней аренды

SET rental\_days = DATEDIFF(NEW.Дата\_окончания\_аренды, NEW.Дата\_начала\_аренды) + 1;

-- Вычисление суммы оплаты

SET total\_sum = (house\_price + bathhouse\_price + inventory\_price) \* rental\_days;

-- Обновление суммы оплаты в таблице Бронирование

SET NEW.Сумма\_оплаты = total\_sum;

END

Триггер 2 `ОбновитьДатуДоступности` автоматически обновляет дату доступности дома, бани, или предмета со склада после того как появилась новая запись о его бронировании.

CREATE DEFINER=`root`@`localhost` TRIGGER `ОбновитьДатуДоступности` AFTER INSERT ON `бронирование` FOR EACH ROW BEGIN

UPDATE Дома

SET Дата\_доступности = (SELECT MIN(Дата\_окончания\_аренды) FROM база\_отдыха.бронирование

WHERE Номер\_дома = NEW.Номер\_дома

AND Дата\_окончания\_аренды NOT IN (SELECT Дата\_начала\_аренды FROM Бронирование WHERE Номер\_дома = NEW.Номер\_дома) )

WHERE Номер\_дома = NEW.Номер\_дома;

UPDATE Бани

SET Дата\_доступности = (SELECT MIN(Дата\_окончания\_аренды) FROM база\_отдыха.бронирование

WHERE Номер\_дома = NEW.Номер\_дома

AND Дата\_окончания\_аренды NOT IN (SELECT Дата\_начала\_аренды FROM Бронирование WHERE Номер\_дома = NEW.Номер\_дома) )

WHERE Номер\_бани = NEW.Номер\_бани;

UPDATE Склад

SET Дата\_доступности = (SELECT MIN(Дата\_окончания\_аренды) FROM база\_отдыха.бронирование

WHERE Номер\_дома = NEW.Номер\_дома

AND Дата\_окончания\_аренды NOT IN (SELECT Дата\_начала\_аренды FROM Бронирование WHERE Номер\_дома = NEW.Номер\_дома) )

WHERE Название\_позиции = NEW.Название\_позиции\_на\_складе;

END

3.5 Разработка функций

В процессе разработки базы данных была выявлена необходимость использования функций, которые представляют собой набор инструкций, выполняющих определенные операции и возвращающих результат.

Функция 1 `ПроверкаДомаНаДоступность` проверяет доступен ли конкретный дом для аренды в конкретную дату. На вход функции подаются два значения: номер дома и желаемая дата брони.

CREATE FUNCTION ПроверкаДомаНаДоступность(Дата DATE, Номер\_дома INT) RETURNS VARCHAR(10)

DETERMINISTIC

BEGIN

DECLARE status VARCHAR(10);

SELECT CASE

WHEN COUNT(\*) > 0 THEN 'Занят'

ELSE 'Свободен'

END INTO status

FROM Дома

WHERE Номер\_дома = Номер\_дома

AND дата\_доступности > Дата;

RETURN status;

Функция 2 `ПроверкаДоступностиБани` проверяет доступна ли конкретная баня для аренды в конкретную дату. На вход функции подаются два значения: номер бани и желаемая дата брони.

CREATE FUNCTION ПроверкаДоступностиБани(датаПроверки DATE, номерБани INT) RETURNS VARCHAR(10)

DETERMINISTIC

BEGIN

DECLARE статус VARCHAR(10);

SELECT CASE

WHEN COUNT(\*) > 0 THEN 'Занята'

ELSE 'Свободна'

END INTO статус

FROM Бани

WHERE Номер\_бани = номерБани

AND Дата\_доступности > датаПроверки;

RETURN статус;

END //

Функция 3 `ПроверкаДоступностиНаCкладе` проверяет доступна ли конкретная позиция склада для аренды в конкретную дату. На вход функции подаются два значения: название позиции со склада и желаемая дата брони.

CREATE FUNCTION ПроверкаДоступностиНаCкладе(датаПроверки DATE, названиеПозиции VARCHAR(100)) RETURNS VARCHAR(10)

DETERMINISTIC

BEGIN

DECLARE статус VARCHAR(10);

SELECT CASE

WHEN COUNT(\*) > 0 THEN 'Занято'

ELSE 'Свободно'

END INTO статус

FROM Склад

WHERE Название\_позиции = названиеПозиции

AND Дата\_доступности > датаПроверки;

RETURN статус;

END //

# Заключение

В процессе разработки был проведен подробный анализ предметной области, связанной с деятельностью аптеки. Анализ включал изучение структуры аптеки, ее персонала и информационных потребностей, а также анализ документооборота и определение ограничений, связанных с базой данных.

На основе этого анализа была выполнена детальная оценка входной и выходной информации, выделены информационные объекты и учтены особенности документооборота. Затем сущности были приведены к третьей нормальной форме, используя связь "один ко многим". Были определены типы данных для атрибутов сущностей, и информационно-логическая модель была создана с помощью MySQL Workbench, инструмента с графическим интерфейсом для управления базами данных.

В результате работы была успешно разработана база данных, которая оптимизирует бизнес-процессы в аптеке и удовлетворяет требованиям предметной области. Созданная база данных поможет повысить эффективность и надежность операций, связанных с управлением клиентами, заказами, поставщиками, продуктами и другими аспектами работы аптеки.