Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

Рязанский государственный радиотехнический университет  
имени В.Ф. Уткина

Кафедра ЭВМ

К защите

Руководитель работы:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

по дисциплине

**«Клиент серверные приложения баз данных»**

Тема:

«Разработка информационной системы общежития»

Выполнил студент группы 145

Жупин С.Ю.

дата сдачи на проверку, подпись

Руководитель работы

ассистент каф. ЭВМ

Баранова С.Н.

оценка дата защиты, подпись

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc167319412)

[1 Выявление задач автоматизации 4](#_Toc167319413)

[1.1 Описание автоматизированной области 4](#_Toc167319414)

[1.2 Обоснование актуальности разработки 5](#_Toc167319415)

[1.3 Постановка задачи 5](#_Toc167319416)

[1.4 Разработка архитектуры ИС 6](#_Toc167319417)

[2 Разработка серверной части ИС 8](#_Toc167319418)

[2.1 Инфологическое проектирование БД 8](#_Toc167319419)

[2.1.1 Требуемая информация 8](#_Toc167319420)

[2.1.2 Выделение сущностей 9](#_Toc167319421)

[2.1.3 Выделение связей 10](#_Toc167319422)

[2.1.4 Построение ER диаграммы ы(Рисунки 2-14) 10](#_Toc167319423)

[2.2 Даталогическое проектирование БД 14](#_Toc167319424)

[2.2.1 Формирование предварительных отношений 14](#_Toc167319425)

[2.2.2 Распределение атрибутов по отношениям 16](#_Toc167319426)

[2.3 Разработка объектов поддержания целостности данных 18](#_Toc167319427)

[3 Разработка клиентской части ИС 19](#_Toc167319428)

[3.1 Разработка прототипа интерфейса пользователя 19](#_Toc167319429)

[3.2 Реализация интерфейса пользователя 19](#_Toc167319430)

[4 Тестирование основных функций приложения 20](#_Toc167319431)

[ЗАКЛЮЧЕНИЕ 21](#_Toc167319432)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 22](#_Toc167319433)

[ПРИЛОЖЕНИЕ А: сценарий создания объектов БД 23](#_Toc167319434)

[ПРИЛОЖЕНИЕ Б: сценарий заполнения таблиц БД 24](#_Toc167319435)

[ПРИЛОЖЕНИЕ В:исходный код клиентского приложения 25](#_Toc167319436)

# ВВЕДЕНИЕ

На сегодняшний день информационные системы (далее ИС) применяются во множестве различных сферах деятельности человека. ИС позволяют автоматизировать и ускорить процессы обработки, передачи и создания различных данных тем самым позволяя оптимизировать различные прикладные процессы. Благодаря цифровому представлению данных можно избавиться от большого количества бумажных архивов, значительно ускорить передачу информации и упростить обработку информации, а также обеспечить предоставление информации в удобной форме.

В данном курсовом проекте будет создана информационная система общежития, которая будет состоять из сервера состоящем из базы данных MS SQL Server и клиентского приложения на платформе Windows. Данная ИС позволит хранить, добавлять и обрабатывать информацию, требуемую для администрирования общежития в удобном виде, а также позволит делать это удаленно.

# Выявление задач автоматизации

## Описание автоматизированной области

Необходимо спроектировать ИС для студенческого общежития РГРТУ, предоставляющей проживание для студентов как самого РГРТУ, так и других вузов и профессиональных училищ. ИС. ИС разрабатывается для одного общежития, но при необходимости ее можно будет улучшить для администрирования сразу множества общежитий, например, в пределе одного студенческого городка.

Система хранит информацию о проживающих в общежитии студентах с указанием их персональных данных, местом их обучения, а также о тарифе и долгах за проживание.

Система хранит информацию о структуре общежития, такую как список блоков с указанием их типов и назначениях.

Система хранит информацию о всех помещениях общежития с указанием типа комнаты, ее принадлежности к блоку и количеству доступных для заселения мест.

Система хранит информацию о работниках общежития (вахтеры, уборщики, слесари и др.) с указанием их персональных данных и занимаемой должности.

Система хранит информацию о существующих в общежитии должностях. Должность определяет базовый оклад работника.

Ведется отслеживание различного рода инвентаря (мебель, фурнитура и т.д.) и хранение информации о нем, такой как название и описание, стоимость, дата поставки и комната его размещения.

При заселении нового или переселении уже существующего в системе студента администратор общежития вносит соответствующие изменения.

Проживающие в общежитии могут создать заявку с жалобой или предложением, на которую администратором будет назначен ответственный за ее решение из работников общежития.

При устройстве на работу в общежитии администратор вносит нового работника в систему с указанием его персональных данных и должности.

Администратор назначает дежурных по блокам общежития из работников, а также вахтеров на вахты из ранее размеченного расписания вахт.

Администратор составляет расписание вахт в общежитии с указанием даты захода и длительностью смены.

## Обоснование актуальности разработки

При изучении предметной области было замечено что многая информация о проживающих в общежитии хранится в бумажном виде, а также отсутствует доступ о данных студентах из деканатов образовательных организаций, из-за этого возникают ситуации беготни с бумагами. Также хранение информации в бумажном виде сильно ограничивает скорость ее обработки и увеличивает шансы допуска ошибок, которые могут нарушить целостность данных.

Также электронное представление данных значительно оптимизирует работу администратора общежития что позволит повысить производительность труда.

## Постановка задачи

Для данной предметной области основными задачами автоматизации являются:

– Автоматизация операций заселения/переселения проживающих.

– Автоматизация операций с заявками и предложениями.

– Автоматизация инвентаризации.

– Автоматизация назначения работ для работников общежития.

– Удобный и быстрый поиск и отображение информации.

– Удаленный просмотр информации.

## Разработка архитектуры ИС

Для данной ИС была выбрана двухуровневая клиент-серверная архитектура. Она позволяет значительно сэкономить на серверной части, так как в отличии от распределенных систем здесь требуется только один физический сервер.

Из других преимуществ такая архитектура способна поддерживать целостность данных на обоих уровнях и равномерно распределять нагрузку между клиентом и сервером. Также такая архитектура позволяет организовать централизованную защиту данных, т.к. все данные будут храниться на сервере. Клиентское приложение в свою очередь будет отправлять запросы на сервер и выводить пользователю полученную информацию в форматированном виде. Также с клиента можно отправлять параметризированные запросы с различными ограничениями на добавление/изменение/удаление данных, которые позволят исключить нарушение целостности данных и SQL-инъекции.

Централизованное хранилище данных является не только преимуществом, но и недостатком, так как при нарушении работы сервера все экземпляры приложений-клиентов потеряют доступ к данным, а также при несанкционированном доступе будет утечка сразу всех данных.

Схема архитектуры ИС представлена на рисунке 1. Виртуальный сервер БД физически располагается на устройстве-сервере, также сервер БД представляет часть слоя логики и выполняет большинство операций по обработке данных. Клиентское приложение содержит в себе остальную часть слоя логики который формирует запросы к серверу и слой представления данных полученных от сервера.

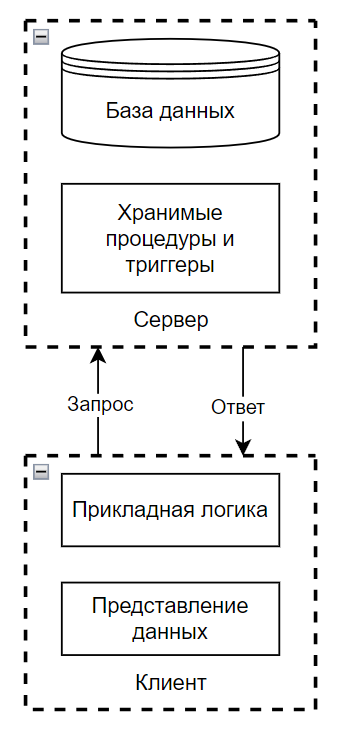


Рисунок 1 – Архитектура ИС

# Разработка серверной части ИС

## Инфологическое проектирование БД

### Требуемая информация

Исходя из предметной области в БД требуется хранить следующую информацию:

1. **Информацию о проживающих:**

– ФИО,

– Серия и номер паспорта,

– Пол (муж/жен),

– Дата рождения,

– Номер телефона для связи,

– Учебное заведение, в котором он обучается,

– Комната проживания,

– Тариф, который зависит от льгот и места обучения.

1. **Информацию о комнате:**

– Тип комнаты,

– Номер блока, в котором она находится,

– Количество койко-мест, которое зависит от типа комнаты.

1. **Информацию о блоках:**

– Тип блока, который определяет какие типы проживающих рекомендуется заселять в блок,

– Принадлежность блока к крылу и этажу,

– Информацию о работнике являющийся дежурным по блоку.

1. **Информацию об инвентаре**

– Наименование и описание,

– Дата поставки,

– Стоимость,

– Номер комнаты размещения.

1. **Информацию о вахтах:**

– Тип вахты,

– Ставка почасового оклада за вахту в зависимости от типа вахты

– Дата и время заступления на вахту,

– Длительность вахты в часах

– Назначенный на вахту работник общежития.

1. **Информацию о работниках общежития:**

– ФИО,

– Серия и номер паспорта,

– Телефон для связи,

– Должность,

– Базовый оклад, зависимый от должности.

1. **Информацию о заявках** –

– Тема для удобного поиска и сортировки,

– Основной текст,

– Статус заявки (создана, закрыта, в процессе),

– Дата создания,

– Составитель из проживающих,

### Выделение сущностей

Для хранения вышеуказанной информации необходимо выделить следующие сущности:

* Проживающий (Код проживающего)
* Тип проживающего (Название типа)
* Комната (Номер комнаты)
* Тип комнаты (Название типа)
* Блок (Номер блока)
* Тип блока (Название типа)
* Инвентарь (Наименование)
* Работник (Код работника)
* Должность (Название должности)
* Вахта (Номер вахты)
* Тип вахты (Название типа)
* Заявка (Номер заявки)

### Выделение связей

Проживающий имеет тип проживающего;

Проживающий проживает в комнате;

Заявка составляется проживающим;

Комната имеет тип комнаты;

Комната содержит инвентарь;

Блок содержит комнаты;

Работник дежурит в блоке;

Работник дежурит на вахтах;

Работник занимает должность;

Вахта имеет тип вахты;

### Построение ER диаграммы (Рисунки 2-11)

**Проживающий имеет тип**

**–** Проживающий имеет только один тип

**–** Проживающий обязательно имеет тип

**–** Один тип может иметь множество проживающих

**–** Тип не обязательно должен иметь проживающих



Рисунок 2 – Проживающий имеет тип

**Проживающий проживает в комнате**

**–** Проживающий живет только в одной комнате

**–** Проживающий обязательно должен проживать в комнате

**–** В одной комнате могут жить несколько проживающих

**–** В комнате не обязательно должны проживать

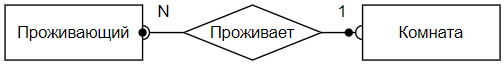


Рисунок 3 – Проживающий проживает в комнате

**Заявка составляется проживающим**

**–** Проживающий может составить множество заявок

**–** Проживающему не обязательно составлять заявки

**–** Одну заявку составляет только один проживающий

**–** Заявка обязательно содержит информацию о составителе

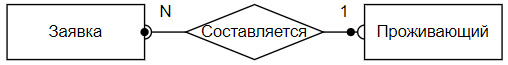


Рисунок 4 – Заявка составляется проживающим

**Комната имеет тип**

**–** Комната обязательно имеет тип

**–** Комната может иметь только один тип

**–** Одному типу может соответствовать множество комнат

**–** Типу не обязательно иметь соответствующую ему комнату



Рисунок 5 – Комната имеет тип

**Комната содержит инвентарь**

**–** В одной комнате может содержаться множество инвентаря

**–** В комнате может не быть инвентаря

**–** Инвентарь обязательно имеет комнату, в которой он размещается

**–** Единица инвентаря размещается только в одной комнате

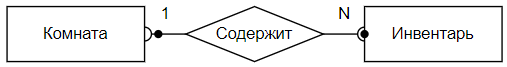


Рисунок 6 – Комната содержит инвентарь

**Блок содержит комнаты**

**–** В одном блоке может быть множество комнат

**–** Комната обязательно должна иметь блок, в котором она находится

**–** Комната может находиться только в одном блоке

**–** Блок необязательно должен иметь комнаты

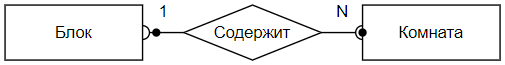


Рисунок 7 – Блок содержит комнаты

**Работник дежурит в блоке**

**–** В одном блоке может дежурить только один работник

**–** В блоке необязательно должны дежурить

**–** Работник может дежурить в множестве блоках

**–** Работнику необязательно дежурить

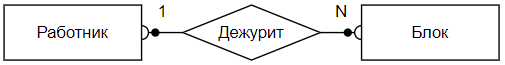


Рисунок 8 – Работник дежурит в блоке

**Работник дежурит на вахтах**

**–** Работнику необязательно дежурить на вахте

**–** Работник может дежурить на нескольких вахтах

**–** На одну вахту назначается только один работник

**–** Вахте обязательно должен быть назначен работник



Рисунок 9 – Работник дежурит на вахтах

**Работник занимает должность**

**–** Работник занимает только одну должность

**–** Работнику обязательно занимать должность

**–** На одной должности может состоять множество работников

**–** Должность может никто не занимать

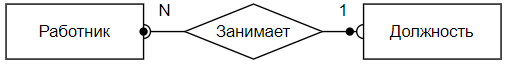


Рисунок 10 – Работник занимает должность

**Вахта имеет тип вахты**

**–** Вахта имеет только один тип вахты

**–** Вахта обязательно имеет тип

**–** Одному типу может соответствовать множество вахт

**–** Тип может не иметь соответствующих ему вахт



Рисунок 11 – Вахта имеет тип вахты

**Полная ER диаграмма** представлена на рисунке 12.

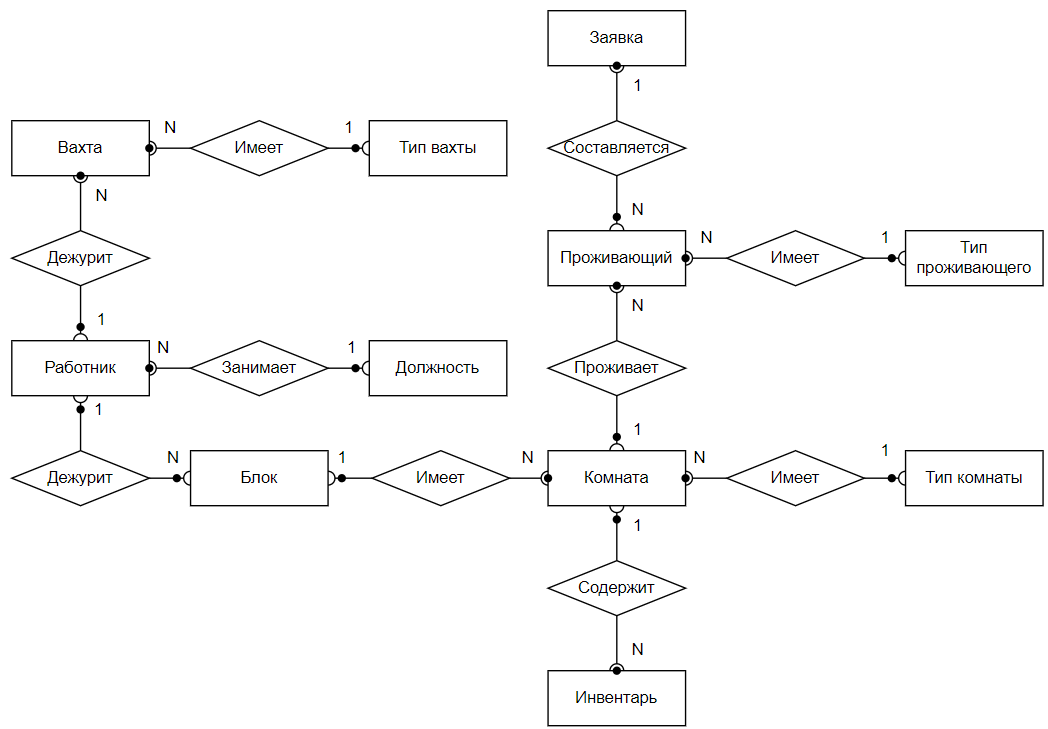


Рисунок 12 – ER диаграмма

## Даталогическое проектирование БД

### Формирование предварительных отношений

**Связь «проживающий имеет тип проживающего»** по правилу 2 формирует 2 отношения:

Проживающий(Проживающий, ТипПроживающего)

ТипПроживающего(ТипПрожевающего)

**Связь «проживающий проживает в комнате»** по правилу 2 формирует 2 отношения:

Проживающий(Проживающий, ТипПроживающего, Комната)

Комната(Комната)

**Связь «Заявка составляется проживающим»** по правилу 2 формирует 2 отношения:

Заявка(Заявка, Проживающий)

Проживающий(Проживающий, ТипПроживающего)

**Связь «Комната имеет тип комнаты»** по правилу 2 формирует 2 отношения:

Комната(Комната, ТипКомнаты)

ТипКомнаты(ТипКомнаты)

**Связь «Комната содержит инвентарь»** по правилу 2 формирует 2 отношения:

Инвентарь(Инвентарь, Комната)

Комната(Комната, ТипКомнаты)

**Связь «Блок содержит комнаты»** по правилу 2 формирует 2 отношения:

Комната(Комната, ТипКомнаты,Блок)

Блок(Блок)

**Связь «Работник дежурит в блоке»** по правилу 5 формирует 3 отношения:

Дежурство(Блок, Работник)

Блок(Блок)

Работник(Работник)

**Связь «Работник дежурит на вахте»** по правилу 2 формирует 2 отношения:

Вахта(Вахта, Работник)

Работник(Работник)

**Связь «Работник занимает должность»** по правилу 2 формирует 2 отношения:

Работник(Работник, Должность)

Должность(Должность)

**Связь «Вахта имеет тип вахты»** по правилу 2 формирует 2 отношения:

Вахта(Вахта, Работник ,ТипВахты)

ТипВахты(ТипВахты)

**Полная предварительная диаграмма** отношений представлена на рисунке 14

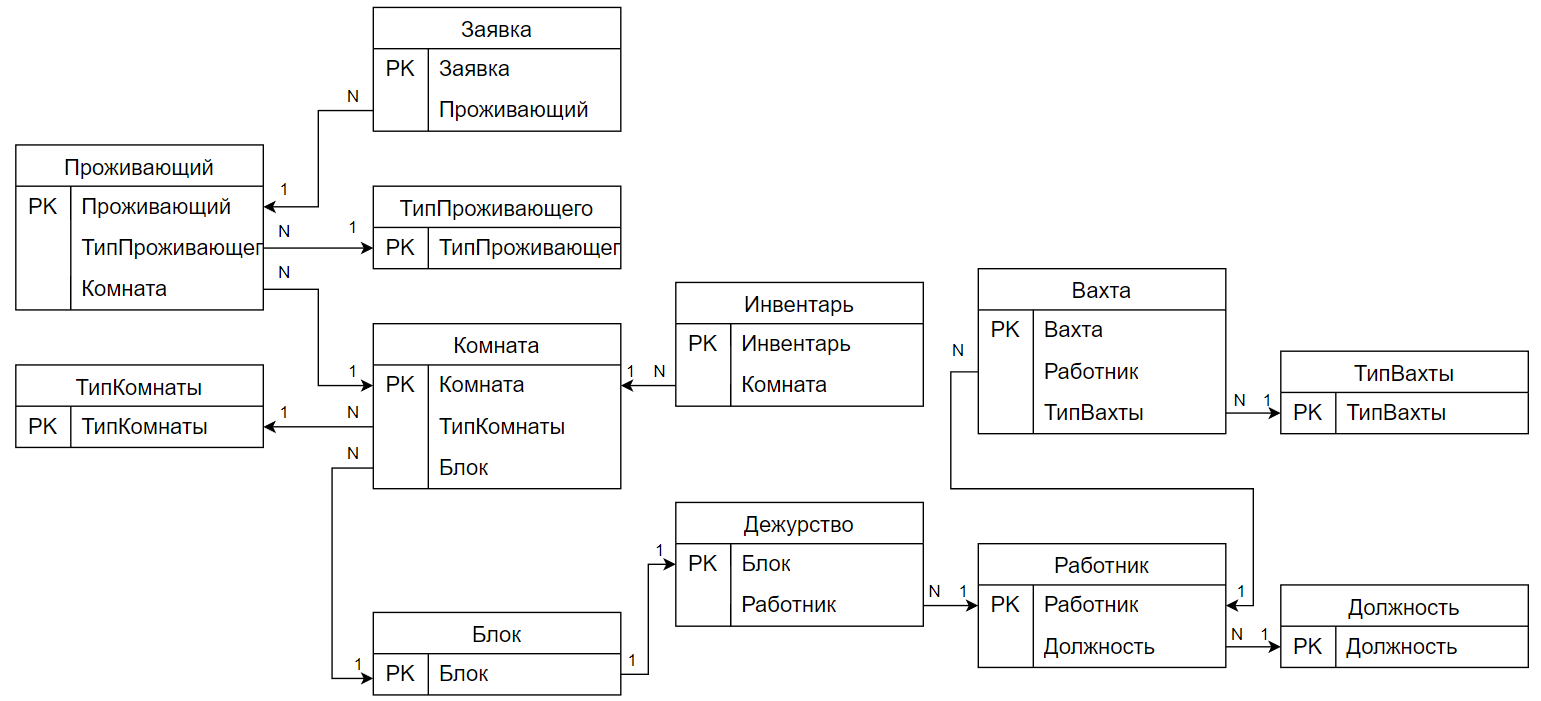


Рисунок 13 – Предварительная диаграмма отношений

### Распределение атрибутов по отношениям

**Отношение Комната содержит атрибуты:**

– Номер комнаты

– Тип

– Количество мест

– Блок

**Отношение Проживающий содержит атрибуты:**

– Код проживающего (серия и номер паспорта)

– ФИО

– Телефон

– Пол

– Дата рождения

– Тип

– Номер комнаты

**Отношение Работник содержит атрибуты:**

– Кол работника (серия и номер паспорта

– ФИО

– Телефон

– Должность

**Отношение Блок содержит атрибуты:**

– Код блока

– Этаж

– Крыло

– Тип

**Отношение Инвентарь содержит атрибуты:**

– Код инвентаря

– Название

– Дата поставки

– Стоимость

– Комната

**Отношение Заявки содержит атрибуты:**

– Номер

– Тема

– Текст

– Статус

– Дата составления

– Составитель

**Отношение Вахты содержит атрибуты:**

– Номер

– Тип

– Дата захода

– Длительность

– Вахтер

**Отношение Дежурство содержит атрибуты:**

– Блок

– Дежурный

**Отношение ТипПроживающего содержит атрибуты:**

– Название (место обучения + льготы)

– Тариф

**Отношение ТипКомнаты содержит атрибуты:**

– Название

– Количество мест

**Отношение Должность содержит атрибуты:**

– Название

– Оклад

**Отношение ТипВахты содержит атрибуты:**

– Название

– Ставка

### Проверка отношений на БКНФ

Все отношения находятся:

– В 1 НФ, так как все артибуты являются атомарными,

– Во 2НФ поскольку все неключевые элементы функционально полно зависят от первичного ключа,

– В 3НФ так как в нем нет транзитивных зависимостей,

– В БКНФ потому что детерминант функциональных зависимостей является единственным потенциальным ключом (первичным).

Схемы функциональных зависимостей показаны на рисунках 14-21

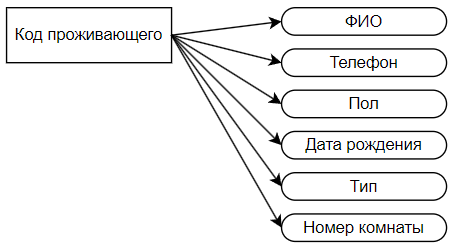


Рисунок 14 – зависимости отношения Проживающий

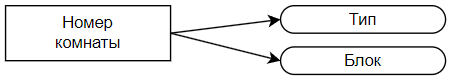


Рисунок 15 – зависимости отношения Комната

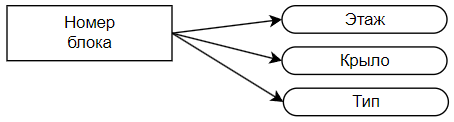


Рисунок 16 – зависимости отношения Блок

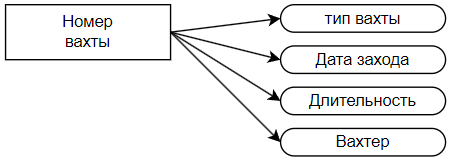


Рисунок 17 – зависимости отношения Вахта

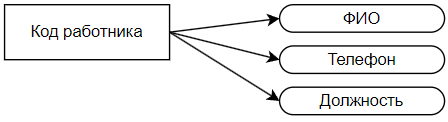


Рисунок 18 – зависимости отношения Работник

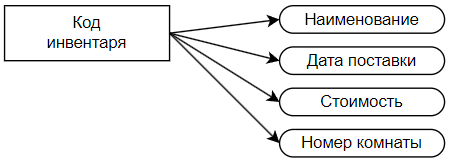


Рисунок 19 – зависимости отношения Инвентарь

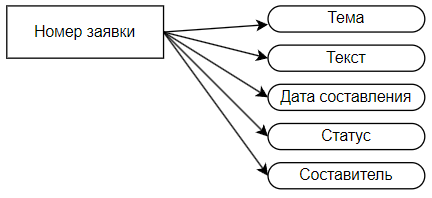


Рисунок 20 – зависимости отношения Заявка

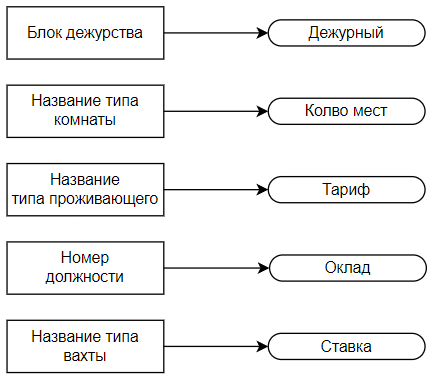


Рисунок 21 – зависимости отношений Дежурство, ТипКомнаты, ТипПроживающего, Должность и ТипВахты

**Полная схема БД** представлена на рисунке 22.

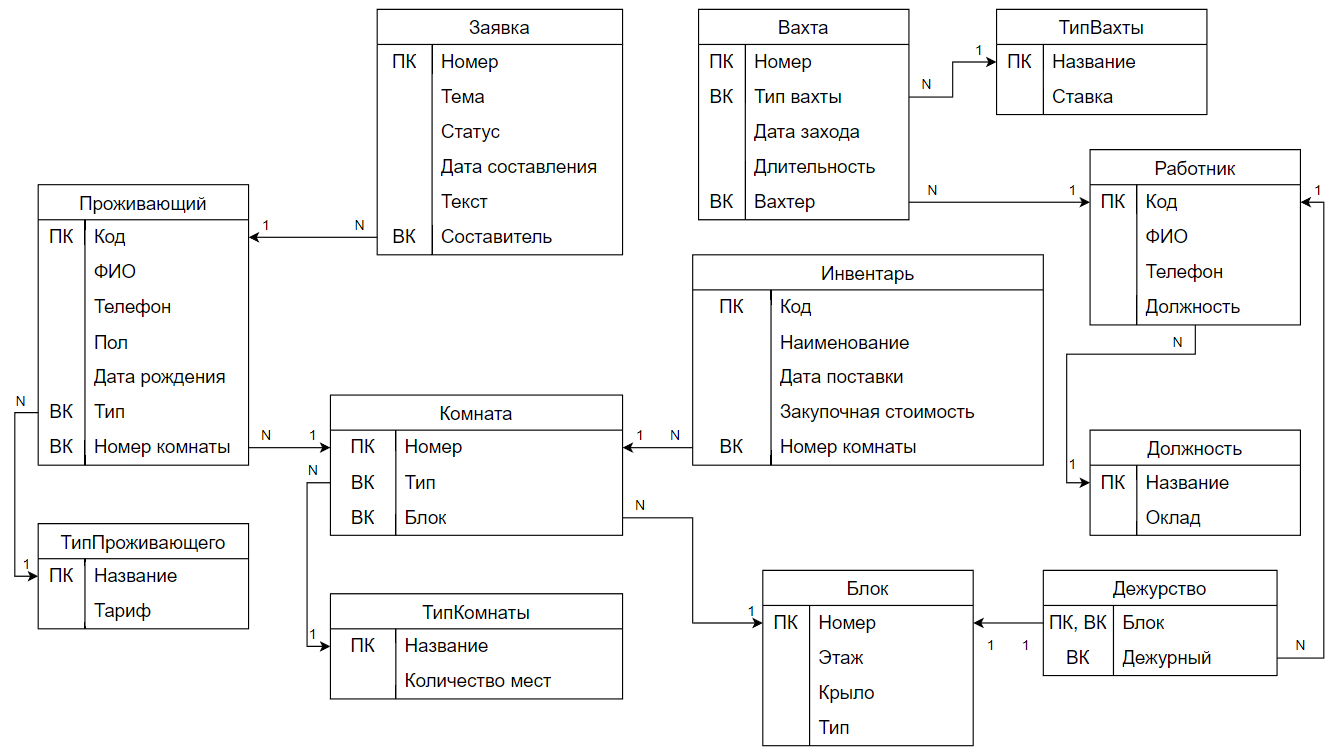


Рисунок 22 –Полная схема БД

## Разработка объектов поддержания целостности данных

Сценарии создания объектов БД представлены в [ПРИЛОЖЕНИИ А](#_ПРИЛОЖЕНИЕ_А:_сценарий).

### Разработка правил, умолчаний и типов

**Таблица «Проживающие»**

Атрибут «Код» (номер паспорта) является первичным ключем, не может принимать NULL значения, значения находятся в диапазоне 1000000000 – 9999999999.

Атрибут «ФИО» не может принимать NULL значения, значения должны соответствовать шаблону [А-Я]% [А-Я]% [А-Я]%.

Атрибут «Телефон» не может принимать NULL значения, значения должны соответствовать шаблону +7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9] или 0([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9].

Атрибут «Пол» не может принимать NULL значения, значения могут быть только «муж» или «жен».

Атрибут «Дата рождения» не может принимать NULL значения.

Атрибут «Тип» не может принимать NULL значения.

Атрибут «Комната» не может принимать NULL значения.

**Таблица «Заявки»**

Атрибут «Номер» является первичным ключем, не может принимать NULL значения.

Атрибут «Тема» не может принимать NULL значения, значение по умолчанию «Без темы».

Атрибут Статус не может принимать NULL значения, список допустимых значений: «Открыта», «Выполняется», «Отклонена» или «Выполнена», значение по умолчанию «Открыта».

Атрибут «ДатаСоставления» не может принимать NULL значения, по умолчанию устанавливается текущая дата и время.

Атрибут Текст является необязательным.

Атрибут Составитель не может принимать NULL значения.

**Таблица «Вахты»**

Атрибут «Номер» является первичным ключем, не может принимать NULL значения.

Атрибут «Тип» не может принимать NULL значения.

Атрибут «ДатаНачала» не может принимать NULL значения, по умолчанию устанавливается текущая дата и время.

Атрибут «Длительность» не может принимать NULL значения и значения меньше нуля.

Атрибут «Вахтер» не может принимать NULL значения.

**Таблица «Инвентарь»**

Атрибут «Код» является первичным ключем, не может принимать NULL значения.

Атрибут «Наименование» не может принимать NULL значения.

Атрибут «ДатаПоставки» не может принимать NULL значения, по умолчанию устанавливается текущая дата и время.

Атрибут «Стоимость» не может принимать NULL значения и отрицательные значения.

Атрибут «Комната» не может принимать NULL значения.

**Таблица «Блоки»**

Атрибут «Номер» является первичным ключем, не может принимать NULL значения.

Атрибут «Этаж» не может принимать NULL значения.

Атрибут «Крыло» не может принимать NULL значения.

Атрибут «Тип» не может принимать NULL значения, список допустимых значений: «Мужской», «Женский», «Общий» «Преподавательский», «Семейный» или «Служебный» .

**Таблица «Работники»**

Атрибут «Код» (номер паспорта) является первичным ключем, не может принимать NULL значения, значения находятся в диапазоне 1000000000 – 9999999999.

Атрибут «ФИО» не может принимать NULL значения, значения должны соответствовать шаблону [А-Я]% [А-Я]% [А-Я]%.

Атрибут «Телефон» не может принимать NULL значения, значения должны соответствовать шаблону +7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9] или 0([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9].

Атрибут «Должность» не может принимать NULL значения.

# Разработка клиентской части ИС

## Разработка прототипа интерфейса пользователя

## Реализация интерфейса пользователя

# Тестирование основных функций приложения

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

# ПРИЛОЖЕНИЕ А: сценарий создания объектов БД

# ПРИЛОЖЕНИЕ Б: сценарий заполнения таблиц БД

# ПРИЛОЖЕНИЕ В:исходный код клиентского приложения