

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет фундаментальной и прикладной информатики
Направление: 02.03.02 «Фундаментальная информатика и информационные
технологии»

ОТЧЕТ
по лабораторной работе №1
«Проектирование информационной модели для реляционных
баз данных»

Вариант 65

Выполнил:
студент группы ФИИТ-ХХ-ХХ
Иванов И.И.

Проверил:
доцент кафедры ФИИТ
Петров П.П.

Иркутск 2024

РЕФЕРАТ

Отчет по лабораторной работе №1: «Проектирование информационной модели для реляционных баз данных».

Количество страниц - 8, **количество рисунков** - 3, **количество источников** - 2.

Ключевые слова: БАЗЫ ДАННЫХ, ER-МОДЕЛЬ, ЛОГИЧЕСКАЯ МОДЕЛЬ, ФИЗИЧЕСКАЯ МОДЕЛЬ, POSTGRESQL, НОРМАЛИЗАЦИЯ, СУРОГАТНЫЙ КЛЮЧ.

Объект исследования: процесс проектирования реляционной базы данных для системы учета контактов и взаимодействий (личная CRM).

Цель работы: разработка корректной информационной модели базы данных, соответствующей требованиям реляционной теории и нормальных форм.

Методы исследования: анализ предметной области, проектирование ER-модели, преобразование в логическую и физическую модели, нормализация базы данных.

Основные результаты: разработана трехуровневая модель базы данных, устранены критические ошибки проектирования, созданы диаграммы для визуализации структуры базы данных.

Содержание

1	ВВЕДЕНИЕ	2
2	ПОСТАНОВКА ЗАДАЧИ	3
2.1	Техническое задание	3
2.2	Исходная модель	4
3	МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ	4
3.1	Уровень 1: Концептуальное проектирование	5
3.2	Уровень 2: Логическое проектирование	5
3.3	Уровень 3: Физическое проектирование	5
4	РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ	5

4.1	Анализ исходной модели	5
4.2	Корректировка модели	6
4.2.1	Введение суррогатных ключей	6
4.2.2	Корректировка типов данных	6
4.2.3	Упрощение структуры заметок	6
4.3	ER-модель	7
4.4	Логическая модель	7
4.5	Физическая модель	8
4.6	Нормализация базы данных	8
4.6.1	Первая нормальная форма (1NF)	8
4.6.2	Вторая нормальная форма (2NF)	8
4.6.3	Третья нормальная форма (3NF)	8
4.6.4	Нормальная форма Бойса-Кодда (BCNF)	8
4.7	Реализация бизнес-требований	9
4.7.1	Список предстоящих встреч на неделю	9
4.7.2	История встреч и заметок для контакта	9
5	ЗАКЛЮЧЕНИЕ	10
6	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	10

1 ВВЕДЕНИЕ

Современные информационные системы требуют тщательного проектирования структур хранения данных. Реляционные базы данных остаются одним из наиболее распространенных подходов к организации хранения структурированной информации. Правильное проектирование схемы базы данных является критически важным этапом, определяющим эффективность, надежность и масштабируемость всей информационной системы.

В данной лабораторной работе рассматривается процесс проектирования базы данных для системы учета контактов и взаимодействий (личной CRM). Такой тип систем широко используется для управления персональными и деловыми контактами, планирования встреч и ведения истории взаимодействий.

Цель работы заключается в разработке корректной информационной модели, соответствующей требованиям реляционной теории и нормальных форм. Для достижения этой цели необходимо решить следующие задачи:

1. Проанализировать предметную область и требования к системе
2. Разработать ER-модель базы данных
3. Преобразовать ER-модель в логическую модель
4. Спроектировать физическую модель для СУБД PostgreSQL
5. Провести нормализацию базы данных
6. Визуализировать результаты с помощью диаграмм

В работе использован итерационный подход к проектированию с привлечением современных инструментов визуализации и анализа.

2 ПОСТАНОВКА ЗАДАЧИ

2.1 Техническое задание

Разработать информационную модель базы данных для системы «Учет контактов и взаимодействий (личная CRM)» со следующими характеристиками:

Сущности системы:

- Контакты (ФИО, место работы, телефон)
- Встречи (дата, тема, место)
- Заметки (дата, текст заметки по контакту)

Бизнес-процессы: регистрация всех встреч и важных событий, связанных с контактами.

Ограничения предметной области:

1. С одним контактом можно организовать несколько встреч
2. Встречи происходят только с одним контактом

3. Система предполагает использование СУБД PostgreSQL

Выходные документы:

1. Список предстоящих встреч на неделю с указанием контактов и тем, отсортированный по дате и времени
2. История всех встреч и заметок для заданного контакта, отсортированная по дате

2.2 Исходная модель

Первоначально была предложена следующая модель базы данных:

```
Contact(family_name, work_place, phone) -- первичный ключ: phone
Meeting(contact_phone, meeting_time, topic, place) -- первичный ключ: meeting_time
Note(pk, contact_phone, meeting_time, note) -- вторичный ключ: (contact_phone, meeting_time)
```

Типы данных для атрибутов:

- family_name::varchar
- work_place::varchar
- phone::Decimal(10)
- meeting_time::datetime
- topic::text
- place::text
- pk::integer
- contact_phone::Decimal(10)
- note::text

3 МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

Проектирование базы данных выполнялось с использованием трехуровневой архитектуры:

3.1 Уровень 1: Концептуальное проектирование

На данном уровне разрабатывается ER-модель (Entity-Relationship model), которая описывает сущности предметной области и отношения между ними без привязки к конкретной СУБД.

3.2 Уровень 2: Логическое проектирование

Преобразование ER-модели в реляционную схему с определением таблиц, атрибутов, первичных и внешних ключей.

3.3 Уровень 3: Физическое проектирование

Реализация логической модели в конкретной СУБД (PostgreSQL) с определением типов данных, индексов, ограничений.

Для визуализации моделей использовался язык Mermaid, поддерживаемый системой GitHub в файлах README.md.

4 РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

4.1 Анализ исходной модели

При анализе первоначальной модели были выявлены следующие критические проблемы:

1. **Некорректный выбор первичных ключей:** использование телефонного номера и времени встречи в качестве первичных ключей нарушает принципы реляционного проектирования
2. **Нарушение нормальных форм:** наличие транзитивных зависимостей в таблице заметок
3. **Некорректные типы данных:** использование числового типа для хранения телефонных номеров
4. **Избыточная сложность:** жесткая привязка заметок одновременно к контактам и встречам

4.2 Корректировка модели

В процессе итерационного проектирования с привлечением экспертной системы были внесены следующие исправления:

4.2.1 Введение суррогатных ключей

Вместо естественных ключей (`phone`, `meeting_time`) введены суррогатные ключи (`contact_id`, `meeting_id`):

```
-- Было: Contact(phone, family_name, work_place)
-- Стало: Contact(contact_id, family_name, work_place, phone)

-- Было: Meeting(contact_phone, meeting_time, topic, place)
-- Стало: Meeting(meeting_id, contact_id, meeting_time, topic, place)
```

4.2.2 Корректировка типов данных

```
-- Было: phone::Decimal(10)
-- Стало: phone VARCHAR(20)

-- Было: meeting_time::datetime
-- Стало: meeting_time TIMESTAMP
```

4.2.3 Упрощение структуры заметок

Упрощена структура таблицы заметок путем удаления обязательной привязки к встречам:

```
-- Было: Note(pk, contact_phone, meeting_time, note)
-- Стало: Note(note_id, contact_id, note_text, created_at)
```

4.3 ER-модель

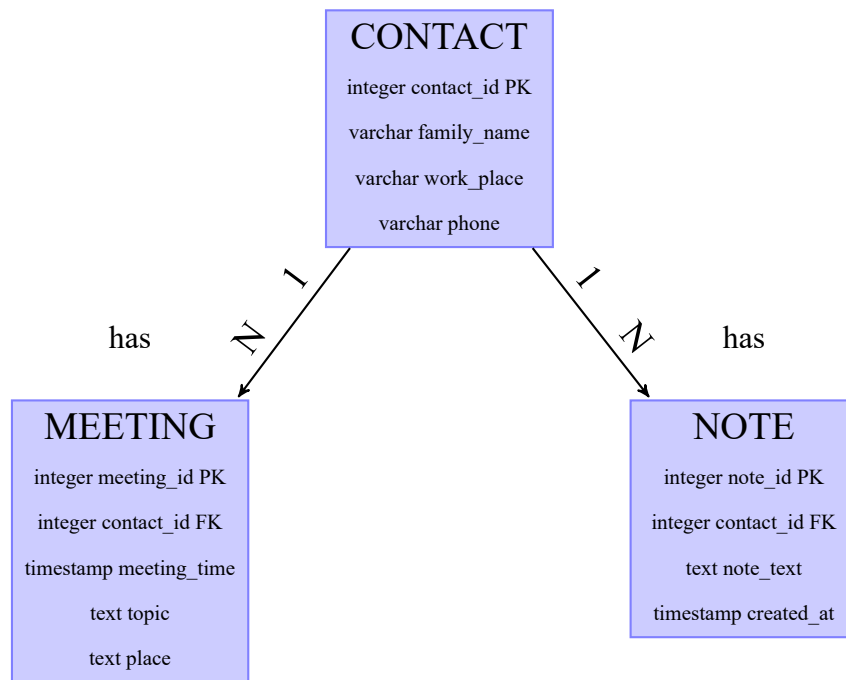


Рис. 1: ER-диаграмма базы данных

4.4 Логическая модель

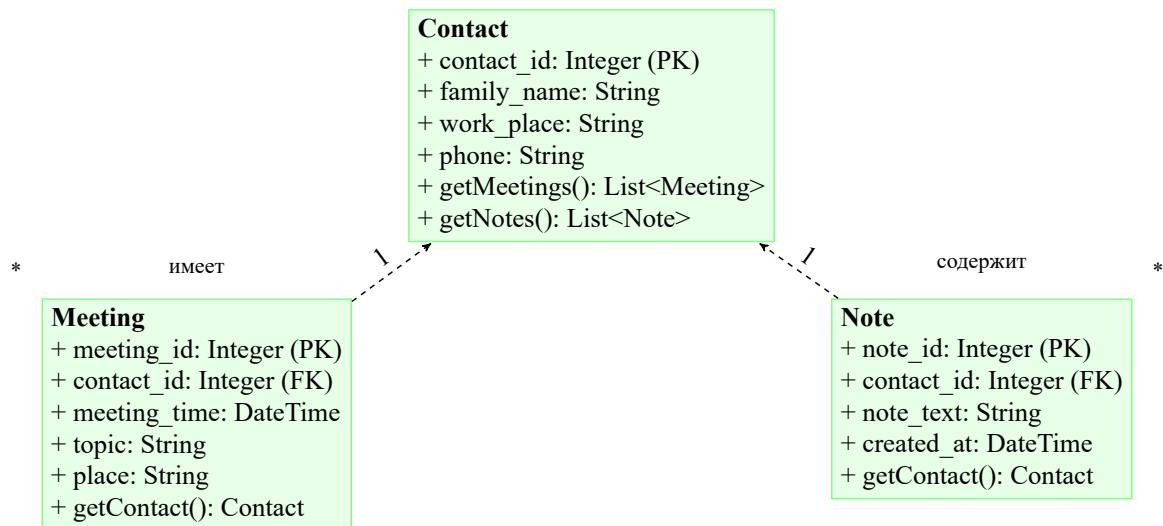


Рис. 2: Логическая модель в виде диаграммы классов UML

4.5 Физическая модель

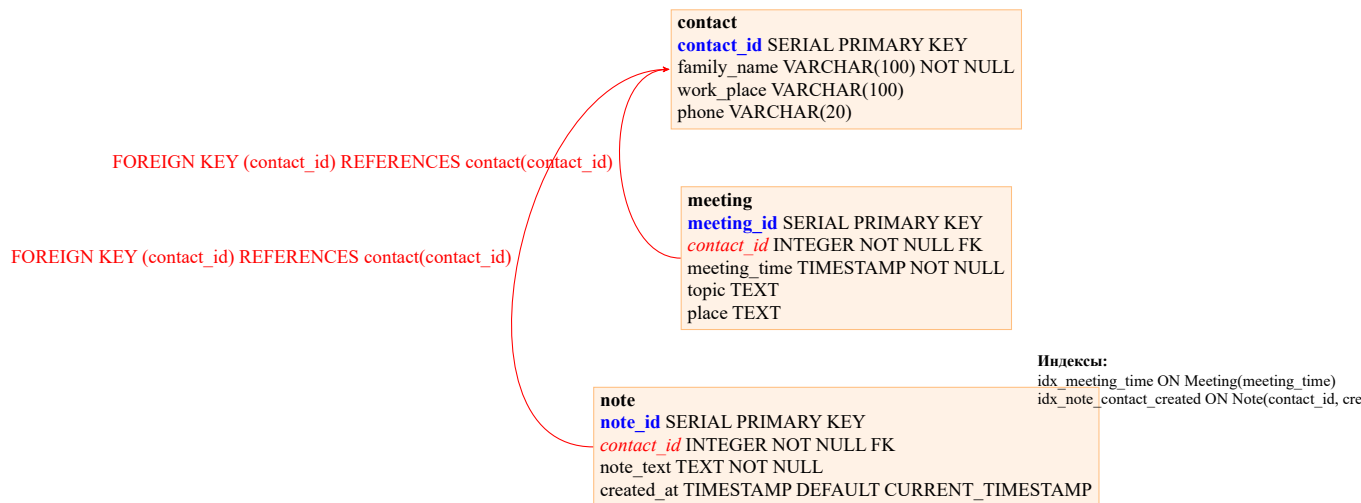


Рис. 3: Физическая модель базы данных

4.6 Нормализация базы данных

Проведена проверка соответствия нормальным формам:

4.6.1 Первая нормальная форма (1NF)

Все атрибуты содержат атомарные значения, повторяющиеся группы отсутствуют.

4.6.2 Вторая нормальная форма (2NF)

Все неключевые атрибуты полностью зависят от целого первичного ключа.

4.6.3 Третья нормальная форма (3NF)

Отсутствуют транзитивные зависимости неключевых атрибутов от неключевых.

4.6.4 Нормальная форма Бойса-Кодда (BCNF)

Каждый детерминант является потенциальным ключом.

4.7 Реализация бизнес-требований

Разработаны SQL-запросы для формирования требуемых выходных документов:

4.7.1 Список предстоящих встреч на неделю

```
SELECT
    c.family_name,
    m.meeting_time,
    m.topic,
    m.place
FROM Meeting m
JOIN Contact c ON m.contact_id = c.contact_id
WHERE m.meeting_time >= CURRENT_DATE
    AND m.meeting_time < CURRENT_DATE + INTERVAL '7 days'
ORDER BY m.meeting_time;
```

4.7.2 История встреч и заметок для контакта

```
SELECT
    m.meeting_time AS event_date,
    'Meeting' AS event_type,
    'Тема: ' || m.topic AS event_description,
    m.place AS details
FROM Meeting m
WHERE m.contact_id = 1

UNION ALL

SELECT
    n.created_at AS event_date,
    'Note' AS event_type,
    n.note_text AS event_description,
    NULL AS details
FROM Note n
WHERE n.contact_id = 1

ORDER BY event_date DESC;
```

5 ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы была успешно разработана информационная модель базы данных для системы учета контактов и взаимодействий. Основные достижения и выводы:

1. **Устранены критические ошибки проектирования:** заменены естественные ключи на суррогатные, что обеспечило стабильность структуры данных при изменениях бизнес-информации
2. **Оптимизирована структура данных:** упрощена модель хранения записей, что повысило гибкость системы и упростило запросы
3. **Обеспечено соответствие нормальным формам:** проведена полная нормализация базы данных до BCNF, что гарантирует отсутствие аномалий при операциях обновления
4. **Создана комплексная документация:** разработаны трехуровневые диаграммы (ER-модель, логическая модель, физическая модель), обеспечивающие наглядное представление структуры базы данных
5. **Реализованы бизнес-требования:** разработаны эффективные SQL-запросы для формирования требуемых выходных документов

Полученная модель демонстрирует соответствие лучшим практикам реляционного проектирования и обеспечивает надежную основу для реализации системы личной CRM. Рекомендуется использовать разработанную модель в качестве основы для дальнейшей реализации приложения.

6 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дейт К. Дж. Введение в системы баз данных. — 8-е изд. — М.: Вильямс, 2005. — 1328 с.
2. PostgreSQL 16.2 Documentation [Электронный ресурс]. — Режим доступа: <https://www.postgresql.org/docs/16/index.html> (дата обращения: 10.11.2024).