

Уральский государственный экономический университет
ФОРМАЛИЗАЦИЯ ИНФОРМАЦИИ И Big Data

02.03.03 -Математическое обеспечение и администрирование информационных систем,
направленность (профиль) -разработка и администрирование информационных систем

<http://vikchas.ru>

Лабораторная работа 5. Тема
«Формализация информации —
технология баз данных»

Часовских Виктор Петрович
доктор технических наук, профессор кафедры
ШИиКМ, ФГБОУ ВО «Уральский государственный
экономический университет

Екатеринбург 2026

Цель работы

Изучить основные принципы формализации информации и проектирования базы данных. Научиться выделять сущности предметной области, определять их атрибуты, устанавливать связи между сущностями и представлять данные в виде таблиц реляционной базы данных.

Теоретические сведения

База данных — это организованная совокупность данных, предназначенная для хранения, обработки и быстрого поиска информации.

Система управления базами данных (СУБД) — программное обеспечение, которое позволяет создавать, изменять и использовать базы данных.

Примеры СУБД:

- ADABAS
- Microsoft Access;
- MySQL;
- PostgreSQL;
- SQLite;
- Oracle Database.

Реляционная база данных хранит информацию в виде таблиц. Каждая таблица состоит из строк и столбцов.

- **Таблица** — объект базы данных, содержащий информацию об однотипных объектах.
- **Поле** — столбец таблицы, описывающий свойство объекта.
- **Запись** — строка таблицы, содержащая данные об одном объекте.
- **Ключевое поле** — поле, которое однозначно идентифицирует запись.
- **Связь** — логическая зависимость между таблицами.

Основные типы связей:

1. Один к одному

Одной записи одной таблицы соответствует одна запись другой таблицы.

2. Один ко многим

Одной записи одной таблицы соответствует несколько записей другой таблицы.

3. Многие ко многим

Нескольким записям одной таблицы соответствуют несколько записей другой таблицы. Обычно реализуется через промежуточную таблицу.

Постановка задачи

Необходимо выполнить формализацию информации для выбранной предметной области и разработать структуру базы данных.

В качестве примера рассмотрим предметную область:

База данных библиотеки

База данных должна хранить информацию о книгах, авторах, читателях и выдаче книг.

Анализ предметной области

В библиотеке имеются книги. Каждая книга имеет название, жанр, год издания и количество экземпляров.

Книги могут быть написаны одним или несколькими авторами. Один автор может написать несколько книг.

Читатели библиотеки могут брать книги на определённый срок. Нужно фиксировать дату выдачи и дату возврата книги.

Выделение сущностей

Для данной предметной области можно выделить следующие сущности:

- 1. Книги**
- 2. Авторы**
- 3. Читатели**
- 4. Выдача книг**
- 5. Авторы книг**

Определение атрибутов сущностей

Таблица «Книги»

Поле	Тип данных	Описание
ID_книги	Счётчик / INT	Уникальный идентификатор книги
Название	Текст	Название книги
Жанр	Текст	Жанр книги
Год_издания	Число	Год издания
Количество	Число	Количество экземпляров

Таблица «Авторы»

Поле	Тип данных	Описание
ID_автора	Счётчик / INT	Уникальный идентификатор автора
Фамилия	Текст	Фамилия автора
Имя	Текст	Имя автора
Отчество	Текст	Отчество автора
Год_рождения	Число	Год рождения автора

Таблица «Читатели»

Поле	Тип данных	Описание
ID_читателя	Счётчик / INT	Уникальный идентификатор читателя
Фамилия	Текст	Фамилия читателя
Имя	Текст	Имя читателя
Отчество	Текст	Отчество читателя

Поле	Тип данных	Описание
Телефон	Текст	Номер телефона
Адрес	Текст	Адрес проживания

Таблица «Выдача книг»

Поле	Тип данных	Описание
ID_выдачи	Счётчик / INT	Уникальный идентификатор выдачи
ID_книги	INT	Ссылка на книгу
ID_читателя	INT	Ссылка на читателя
Дата_выдачи	Дата	Дата выдачи книги
Дата_возврата	Дата	Дата возврата книги

Таблица «Авторы книг»

Так как одна книга может иметь нескольких авторов, а один автор может написать несколько книг, используется промежуточная таблица.

Поле	Тип данных	Описание
ID_книги	INT	Ссылка на книгу
ID_автора	INT	Ссылка на автора

Определение ключей

Первичные ключи

- Книги.ID_книги
- Авторы.ID_автора
- Читатели.ID_читателя
- Выдача_книг.ID_выдачи

Для таблицы «Авторы книг» можно использовать составной первичный ключ:

- ID_книги + ID_автора

Внешние ключи

- Выдача_книг.ID_книги ссылается на Книги.ID_книги
- Выдача_книг.ID_читателя ссылается на Читатели.ID_читателя
- Авторы_книг.ID_книги ссылается на Книги.ID_книги
- Авторы_книг.ID_автора ссылается на Авторы.ID_автора

Связи между таблицами

Таблица 1	Таблица 2	Тип связи
Книги	Выдача книг	Один ко многим
Читатели	Выдача книг	Один ко многим
Книги	Авторы книг	Один ко многим
Авторы	Авторы книг	Один ко многим
Книги	Авторы	Многие ко многим через таблицу «Авторы книг»

Логическая модель базы данных

Логическая структура базы данных:

АВТОРЫ

ID_автора (PK)

Фамилия

Имя

Отчество

Год_рождения

КНИГИ

ID_книги (PK)

Название

Жанр

Год_издания

Количество

ЧИТАТЕЛИ

ID_читателя (PK)

Фамилия

Имя

Отчество

Телефон

Адрес

ВЫДАЧА_КНИГ

ID_выдачи (PK)

ID_книги (FK)

ID_читателя (FK)

Дата_выдачи

Дата_возврата

АВТОРЫ_КНИГ

ID_книги (PK, FK)

ID_автора (PK, FK)

Пример заполнения таблиц

Таблица «Книги»

ID_книги	Название	Жанр	Год_издания	Количество
1	Война и мир	Роман	1869	5
2	Преступление и наказание	Роман	1866	4
3	Мастер и Маргарита	Роман	1967	3

Таблица «Авторы»

ID_автора	Фамилия	Имя	Отчество	Год_рождения
1	Толстой	Лев	Николаевич	1828
2	Достоевский	Фёдор	Михайлович	1821
3	Булгаков	Михаил	Афанасьевич	1891

Таблица «Читатели»

ID_читателя	Фамилия	Имя	Отчество	Телефон	Адрес
1	Иванов	Иван	Петрович	89001234567	Москва
2	Петрова	Анна	Сергеевна	89007654321	Казань
3	Сидоров	Олег	Андреевич	89005554433	Самара

Таблица «Авторы книг»

ID_книги	ID_автора
1	1
2	2
3	3

Таблица «Выдача книг»

ID_выдачи	ID_книги	ID_читателя	Дата_выдачи	Дата_возврата
1	1	1	01.10.2025	15.10.2025
2	2	2	03.10.2025	17.10.2025
3	3	1	05.10.2025	19.10.2025

Пример SQL-кода создания базы данных

```
CREATE TABLE Authors (  
    author_id INTEGER PRIMARY KEY,  
    last_name VARCHAR(50) NOT NULL,  
    first_name VARCHAR(50) NOT NULL,  
    middle_name VARCHAR(50),  
    birth_year INTEGER  
);
```

```
CREATE TABLE Books (  
    book_id INTEGER PRIMARY KEY,  
    title VARCHAR(100) NOT NULL,  
    genre VARCHAR(50),  
    publication_year INTEGER,  
    quantity INTEGER NOT NULL  
);
```

```
CREATE TABLE Readers (  
    reader_id INTEGER PRIMARY KEY,  
    last_name VARCHAR(50) NOT NULL,
```

```
first_name VARCHAR(50) NOT NULL,  
middle_name VARCHAR(50),  
phone VARCHAR(20),  
address VARCHAR(100)  
);
```

```
CREATE TABLE Book_Authors (  
    book_id INTEGER NOT NULL,  
    author_id INTEGER NOT NULL,  
    PRIMARY KEY (book_id, author_id),  
    FOREIGN KEY (book_id) REFERENCES Books(book_id),  
    FOREIGN KEY (author_id) REFERENCES Authors(author_id)  
);
```

```
CREATE TABLE Book_Issues (  
    issue_id INTEGER PRIMARY KEY,  
    book_id INTEGER NOT NULL,  
    reader_id INTEGER NOT NULL,  
    issue_date DATE NOT NULL,  
    return_date DATE,  
    FOREIGN KEY (book_id) REFERENCES Books(book_id),  
    FOREIGN KEY (reader_id) REFERENCES Readers(reader_id)  
);
```

Примеры SQL-запросов

1. Вывести список всех книг

```
SELECT * FROM Books;
```

2. Найти книги определённого автора

```
SELECT Books.title, Authors.last_name, Authors.first_name
FROM Books
JOIN Book_Authors ON Books.book_id = Book_Authors.book_id
JOIN Authors ON Book_Authors.author_id = Authors.author_id
WHERE Authors.last_name = 'Толстой';
```

Вывести книги, выданные читателям

```
SELECT
    Readers.last_name,
    Readers.first_name,
    Books.title,
    Book_Issues.issue_date,
    Book_Issues.return_date
FROM Book_Issues
JOIN Readers ON Book_Issues.reader_id = Readers.reader_id
JOIN Books ON Book_Issues.book_id = Books.book_id;
```

Найти читателя, который взял книгу «Война и мир»

```
SELECT Readers.last_name, Readers.first_name, Books.title
FROM Book_Issues
JOIN Readers ON Book_Issues.reader_id = Readers.reader_id
JOIN Books ON Book_Issues.book_id = Books.book_id
WHERE Books.title = 'Война и мир';
```

Вывод

В ходе лабораторной работы была выполнена формализация информации для предметной области «Библиотека». Были выделены основные сущности,

определены их атрибуты, установлены связи между таблицами и выбраны первичные и внешние ключи.

В результате была разработана структура реляционной базы данных, которая позволяет хранить информацию о книгах, авторах, читателях и выдаче книг. Такая база данных обеспечивает удобное хранение, поиск и обработку информации.

Отчет

Описание и интерпретация созданной базы данных. Скриншоты всех действий.