

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО  
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г.  
ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

**Лабораторная работа №3**

по дисциплине: Базы данных

тема: «Нормализация отношений в базе данных»

Выполнил: ст. группы ПВ-223  
Игнатъев Артур Олегович

Проверил:  
Панченко Максим Владимирович

Белгород 2024 г.

**Цель работы:** изучить нормальные формы отношений, получить навыки последовательной нормализации отношений.

**Задание:**

1. Нормализовать структуру базы данных, разработанную в предыдущих лабораторных работах. Доказать соответствие 3 NF, выписав функциональные зависимости для каждой сущности.

2. Привести данные в задании отношение к 3 NF согласно номеру варианта.

**Вариант 3**

Инвентарный номер	Название книги	Автор	Категория	ФИО читателя	Телефон	Дата выдачи	Дата возврата
32	Кот в сапогах	Ш. Перро	Сказки	Шаров. Д. М.	12-12-02	02.05.13	16.05.13
33	Windows 95 изнутри	А. Кинг	Пользовательский курс	Шил И. А.	34-01-23	05.05.13	08.05.13
33	Windows 95 изнутри	А. Кинг	Пользовательский курс	Акимов а А. А.	44-56-11	08.05.13	22.05.13
35	Русалочка	Х. Кристиан Андерсен	Сказки	Левин Н. А.	12-01-33	13.05.13	24.05.13
35	Русалочка	Х. Кристиан Андерсен	Сказки	Акимов а А. А.	44-56-11	30.05.13	
36	Чистый код	К. Мартин Роберт	Программирование	Разуман Н. В.	71-09-23	14.05.13	28.06.13
37	Совершенный код	М. Стив	Программирование	Шаров Д. М.	12-12-02	28.05.13	01.07.13
38	3D Studio MAX в примерах	Э. Потс	Графика	Акимов а А. А.	44-56-11	29.05.13	12.07.13
38	3D Studio MAX в примерах	Э. Потс	Графика	Черных Л. С.	34-22-34	30.08.13	
37	Совершенный код	М. Стив	Программирование	Шаров Д. М.	12-12-02	31.07.13	

**Задание 1.** Нормализовать структуру базы данных, разработанную в предыдущих лабораторных работах. Доказать соответствие 3 NF, выписав функциональные зависимости для каждой сущности.

База данных из предыдущей работы

```
-- Сначала создаем таблицу статусов, чтобы можно было ссылаться на неё из
других таблиц
CREATE TABLE IF NOT EXISTS status (
    id BIGSERIAL PRIMARY KEY, -- Уникальный идентификатор статуса
    name VARCHAR(50) NOT NULL -- Название статуса (например, "В процессе"),
не может быть NULL
);

-- Таблица водителей
CREATE TABLE IF NOT EXISTS driver (
    id BIGSERIAL PRIMARY KEY, -- Уникальный идентификатор водителя
    license_number INTEGER NOT NULL, -- Номер водительского удостоверения,
не может быть NULL
    name VARCHAR(100) NOT NULL -- Имя и ФИО водителя, не может быть NULL
);

-- Таблица транспортных компаний
CREATE TABLE IF NOT EXISTS transport_company (
    id BIGSERIAL PRIMARY KEY, -- Уникальный идентификатор компании
    name VARCHAR(100) NOT NULL, -- Название компании, не может быть NULL
    driver_count INTEGER DEFAULT 0 -- Количество водителей, по умолчанию 0
);

-- Таблица поставщиков
CREATE TABLE IF NOT EXISTS provider (
    id BIGSERIAL PRIMARY KEY, -- Уникальный идентификатор поставщика
    name VARCHAR(100) NOT NULL -- Название поставщика, не может быть NULL
);

-- Таблица потребителей
CREATE TABLE IF NOT EXISTS consumer (
    id BIGSERIAL PRIMARY KEY, -- Уникальный идентификатор потребителя
    contact_details BIGINT NOT NULL, -- Контактные данные (например, номер
телефона), не может быть NULL
    name VARCHAR(100) NOT NULL -- Имя и ФИО потребителя, не может быть NULL
);

-- Таблица заказов
CREATE TABLE IF NOT EXISTS orders (
    id BIGSERIAL PRIMARY KEY, -- Уникальный идентификатор заказа
    weight INTEGER NOT NULL, -- Вес заказа, не может быть NULL
    status_id BIGINT REFERENCES status(id) ON DELETE SET NULL, -- Статус
заказа с внешним ключом, если статус удаляется, поле становится NULL
    consumer_id BIGINT REFERENCES consumer(id) ON DELETE SET NULL -- Внешний
ключ к потребителю, при удалении потребителя значение становится NULL
);

-- Таблица продуктов
CREATE TABLE IF NOT EXISTS product (
    id BIGSERIAL PRIMARY KEY, -- Уникальный идентификатор продукта
    name VARCHAR(100) NOT NULL -- Название продукта, не может быть NULL
);

-- Таблица, связывающая водителей и транспортные компании (работа водителя в
компании)
CREATE TABLE IF NOT EXISTS driver_company (
    driver_id BIGINT NOT NULL REFERENCES driver(id) ON DELETE CASCADE, --
Внешний ключ к водителю, при удалении водителя запись удаляется
```

```

    company_id BIGINT NOT NULL REFERENCES transport_company(id) ON DELETE
    CASCADE, -- Внешний ключ к компании, при удалении компании запись удаляется
    PRIMARY KEY (driver_id, company_id) -- Составной первичный ключ для
    уникальности пар (водитель, компания)
);

-- Таблица, связывающая поставщиков и продукты (товары, поставляемые
поставщиками)
CREATE TABLE IF NOT EXISTS provider_product (
    provider_id BIGINT NOT NULL REFERENCES provider(id) ON DELETE CASCADE, -
    -- Внешний ключ к поставщику, при удалении поставщика запись удаляется
    product_id BIGINT NOT NULL REFERENCES product(id) ON DELETE CASCADE, --
    -- Внешний ключ к продукту, при удалении продукта запись удаляется
    PRIMARY KEY (provider_id, product_id) -- Составной первичный ключ для
    уникальности пар (поставщик, продукт)
);

```

## Рассмотрим каждую таблицу и нормализуем

### 1. Таблица статусов (status)

```

CREATE TABLE IF NOT EXISTS status (
    id BIGSERIAL PRIMARY KEY,
    name VARCHAR(50) NOT NULL
);

```

**Функциональные зависимости:**  $id \rightarrow name$  (первичный ключ однозначно определяет название статуса).

**1NF:** Все атрибуты атомарны. Многозначных значений нет.

**2NF:** Первичный ключ —  $id$  (простой ключ). Атрибут  $name$  зависит от  $id$ .

**3NF:** Нет транзитивных зависимостей. Атрибут  $name$  зависит только от первичного ключа  $id$ .

**Вывод:** Таблица соответствует 3NF.

### 2. Таблица водителей (driver)

```

CREATE TABLE IF NOT EXISTS driver (
    id BIGSERIAL PRIMARY KEY,
    license_number INTEGER NOT NULL,
    name VARCHAR(100) NOT NULL
);

```

**Функциональные зависимости:**  $id \rightarrow license\_number, name$ .

**1NF:** Все атрибуты атомарны.

**2NF:** Первичный ключ —  $id$ . Все атрибуты зависят от  $id$ .

**3NF:** Нет транзитивных зависимостей.

**Вывод:** Таблица соответствует 3NF.

### 3. Таблица транспортных компаний (transport\_company)

```

CREATE TABLE IF NOT EXISTS transport_company (
    id BIGSERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    driver_count INTEGER DEFAULT 0
);

```

**Функциональные зависимости:**  $id \rightarrow name, driver\_count$ .

**1NF:** Атрибуты атомарны.

**2NF:** Первичный ключ — id. Все атрибуты зависят от id.

**3NF:** Нет транзитивных зависимостей.

**Вывод:** Таблица соответствует 3NF.

4. Таблица поставщиков (provider)

```
CREATE TABLE IF NOT EXISTS provider (  
    id BIGSERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL  
);
```

**Функциональные зависимости:** id→name.

**1NF:** Атрибуты атомарны.

**2NF:** Первичный ключ — id. Атрибут name зависит от id.

**3NF:** Нет транзитивных зависимостей.

**Вывод:** Таблица соответствует 3NF.

5. Таблица потребителей (consumer)

```
CREATE TABLE IF NOT EXISTS consumer (  
    id BIGSERIAL PRIMARY KEY,  
    contact_details BIGINT NOT NULL,  
    name VARCHAR(100) NOT NULL  
);
```

**Функциональные зависимости:** id→contact\_details, name.

**1NF:** Все атрибуты атомарны.

**2NF:** Первичный ключ — id. Все атрибуты зависят от id.

**3NF:** Нет транзитивных зависимостей.

**Вывод:** Таблица соответствует 3NF.

6. Таблица заказов (orders)

```
CREATE TABLE IF NOT EXISTS orders (  
    id BIGSERIAL PRIMARY KEY,  
    weight INTEGER NOT NULL,  
    status_id BIGINT REFERENCES status(id) ON DELETE SET NULL,  
    consumer_id BIGINT REFERENCES consumer(id) ON DELETE SET NULL  
);
```

**Функциональные зависимости:** id→weight, status\_id, consumer\_id.

**1NF:** Атрибуты атомарны.

**2NF:** Первичный ключ — id. Все атрибуты зависят от id.

**3NF:** Внешние ключи status\_id и consumer\_id зависят только от id (прямых транзитивных зависимостей нет).

**Вывод:** Таблица соответствует 3NF.

7. Таблица продуктов (product)

```
CREATE TABLE IF NOT EXISTS product (  
    id BIGSERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL  
);
```

**Функциональные зависимости:**  $id \rightarrow name$ .

**1NF:** Атрибуты атомарны.

**2NF:** Первичный ключ —  $id$ .  $name$  зависит от  $id$ .

**3NF:** Нет транзитивных зависимостей.

**Вывод:** Таблица соответствует 3NF.

8. Связующая таблица водителей и компаний (driver\_company)

```
CREATE TABLE IF NOT EXISTS driver_company (  
    driver_id BIGINT NOT NULL REFERENCES driver(id) ON DELETE  
    CASCADE,  
    company_id BIGINT NOT NULL REFERENCES transport_company(id) ON  
    DELETE CASCADE,  
    PRIMARY KEY (driver_id, company_id)  
);
```

**Функциональные зависимости:**  $(driver\_id, company\_id) \rightarrow$  (составной ключ, нет других атрибутов).

**1NF:** Все атрибуты атомарны.

**2NF:** Все атрибуты (составной ключ) зависят от всего ключа.

**3NF:** Нет транзитивных зависимостей.

**Вывод:** Таблица соответствует 3NF.

9. Связующая таблица поставщиков и продуктов (provider\_product)

```
CREATE TABLE IF NOT EXISTS provider_product (  
    provider_id BIGINT NOT NULL REFERENCES provider(id) ON DELETE  
    CASCADE,  
    product_id BIGINT NOT NULL REFERENCES product(id) ON DELETE  
    CASCADE,  
    PRIMARY KEY (provider_id, product_id)  
);
```

**Функциональные зависимости:**  $(provider\_id, product\_id) \rightarrow$  (составной ключ, нет других атрибутов).

**1NF:** Атрибуты атомарны.

**2NF:** Все атрибуты зависят от составного ключа.

**3NF:** Нет транзитивных зависимостей.

**Вывод:** Таблица соответствует 3NF.

Все таблицы в данной структуре данных соответствуют 3NF. Каждая таблица имеет первичный ключ, функциональные зависимости правильно определены, отсутствуют частичные и транзитивные зависимости.

**Задание 2.** Привести данные в задании отношение к 3 NF согласно номеру варианта.

**Отношение уже находится в 1NF. Все атрибуты атомарны.**

Нарушаются требования 2NF и 3NF: некоторые данные частично зависят от первичного ключа (ФИО и телефон читателя не зависят от инвентарного номера, который является первичным ключом). Кроме того, наблюдаются повторяющиеся данные (одинаковые названия книг) и избыточность информации (многократное дублирование ФИО читателей и их номеров телефонов).

Отношение описывает сразу две сущности — книги и читателей книг при первичном ключе «Инвентарный номер». Для приведения отношения ко **2 NF** разделим его на 2 отношения:

**Отношение 1:**

<b>Инвентарный номер</b>	<b>Название книги</b>	<b>Автор</b>	<b>Категория</b>
32	Кот в сапогах	Ш. Перро	Сказки
33	Windows 95 изнутри	А. Кинг	Пользовательский курс
35	Русалочка	Х. Кристиан Андерсен	Сказки
36	Чистый код	К. Мартин Роберт	Программирование
38	3D Studio MAX в примерах	Э. Потс	Графика
37	Совершенный код	М. Стив	Программирование

**Функциональные зависимости сущности:**

Инвентарный номер → Название книги — название книги зависит от инвентарного номера

Инвентарный номер → Автор — Автор книги зависит от инвентарного номера

Инвентарный номер → Категория — Категория книги зависит от инвентарного номера

**Отношение 2:**

<b>ФИО читател я</b>	<b>Телефон</b>	<b>Дата выдачи</b>	<b>Дата возврата</b>	<b>Инвентарный номер</b>
Шаров. Д. М.	12-12-02	02.05.13	16.05.13	32
Шил И. А.	34-01-23	05.05.13	08.05.13	33
Акимова А. А.	44-56-11	08.05.13	22.05.13	35
Левин Н. А.	12-01-33	13.05.13	24.05.13	36
Акимова А. А.	44-56-11	30.05.13		38
Разуман Н. В.	71-09-23	14.05.13	28.06.13	37
Шаров Д. М.	12-12-02	28.05.13	01.07.13	37
Акимова А. А.	44-56-11	29.05.13	12.07.13	38
Черных Л. С.	34-22-34	30.08.13		38
Шаров Д. М.	12-12-02	31.07.13		37

**Функциональные зависимости сущности:**

ФИО -> Телефон – номер телефона зависит только от ФИО читателя

ФИО -> Дата выдачи – дата выдачи книги зависит только от ФИО читателя

ФИО -> Дата возврата – дата возврата книги зависит только от ФИО читателя

ФИО -> Инвентарный номер – инвентарный номер книги зависит только от ФИО читателя

**Отношения 1 и 2 уже находятся в 3 NF**, так как нет транзитивных зависимостей между неключевыми атрибутами.

**Вывод:** на этой лабораторной работе изучили нормальные формы отношений, получили навыки последовательной нормализации отношений.