

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 №2
по дисциплине: Базы данных
тема: Создание объектов базы данных в СУБД

Выполнил: студент ПВ-233
Мороз Роман Алексеевич

Проверил:
Панченко Максим Владимирович

Белгород 2025 г.

Цель работы: изучить основные возможности языка SQL для создания структуры базы данных. Научиться создавать базы данных, таблицы, связи, ограничения, а также создавать, изменять и удалять данные.

Задание к работе

1. Составить SQL-запросы для создания структуры базы данных, полученной в результате лабораторной работы №1. Указать используемые типы данных, ограничения значений полей; для связей: действия с записями подчинённой таблицы при удалении и изменении соответствующей записи главной таблицы.

База данных "Сервисный центр" состоит из шести таблиц, связанных между собой через внешние ключи. При создании таблиц были использованы различные ограничения для обеспечения целостности данных.

Общие принципы проектирования:

- Для первичных ключей использован тип SERIAL, который обеспечивает автоматическую генерацию уникальных идентификаторов
- Для текстовых полей применялись типы VARCHAR с ограничением длины и TEXT для полей с неограниченным объемом текста
- Для денежных значений использован тип DECIMAL(10,2), обеспечивающий точное хранение финансовых данных
- На большинство критически важных полей наложено ограничение NOT NULL

Особенности ограничений в таблицах:

Ограничения CHECK:

- В таблице products: warranty_period > 0 - гарантийный период должен быть положительным числом
- В таблице products: base_repair_cost >= 0 - базовая стоимость ремонта не может быть отрицательной
- В таблице repairs: actual_repair_cost >= 0 и client_cost >= 0 - стоимости не могут быть отрицательными
- В таблице repairs: chk_repair_dates - дата окончания ремонта не может быть раньше даты начала

Внешние ключи с различными стратегиями:

- ON DELETE RESTRICT - запрещает удаление записей, на которые существуют ссылки (защита целостности данных)
- ON DELETE CASCADE - автоматическое удаление связанных записей (используется для таблицы shop_specializations)
- ON UPDATE CASCADE - автоматическое обновление внешних ключей при изменении первичных ключей

```
-- DROP DATABASE IF EXISTS repair_service;
```

```
CREATE DATABASE repair_service;
```

```
-- Таблица 'manufacturers'
```

```
CREATE TABLE manufacturers (  
    manufacturer_id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    contact_info TEXT  
);
```

```
-- Таблица 'cities'
```

```
CREATE TABLE cities (  
    city_id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL  
);
```

```
-- Таблица 'products'
```

```
CREATE TABLE products (  
    product_id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    category VARCHAR(50) NOT NULL,  
    manufacturer_id INT NOT NULL  
);
```

```

product_id SERIAL PRIMARY KEY,

manufacturer_id INTEGER NOT NULL,

model_name VARCHAR(100) NOT NULL,

category VARCHAR(50) NOT NULL,

warranty_period INTEGER NOT NULL CHECK (warranty_period > 0),

base_repair_cost DECIMAL(10,2) NOT NULL CHECK (base_repair_cost >= 0),


CONSTRAINT fk_products_manufacturer

    FOREIGN KEY (manufacturer_id)

    REFERENCES manufacturers (manufacturer_id)

    ON DELETE RESTRICT      -- Запрещает удаление производителя, если есть
его товары

    ON UPDATE CASCADE      -- Обновляет manufacturer_id в товарах при
изменении в manufacturers
);

-- Таблица 'repair_shops'

CREATE TABLE repair_shops (

    shop_id SERIAL PRIMARY KEY,

    city_id INTEGER NOT NULL,

    address VARCHAR(255) NOT NULL,

    contact_phone VARCHAR(20),

    CONSTRAINT fk_repair_shops_city

        FOREIGN KEY (city_id)

        REFERENCES cities (city_id)

        ON DELETE RESTRICT

        ON UPDATE CASCADE

```

```

);

-- Таблица 'shop_specializations'

CREATE TABLE shop_specializations (

    specialization_id SERIAL PRIMARY KEY,

    shop_id INTEGER NOT NULL,

    product_id INTEGER NOT NULL,

    CONSTRAINT uq_shop_product UNIQUE (shop_id, product_id),

    CONSTRAINT fk_specializations_shop

        FOREIGN KEY (shop_id)

        REFERENCES repair_shops(shop_id)

        ON DELETE CASCADE -- Удаляет специализацию при удалении мастерской

        ON UPDATE CASCADE,

    CONSTRAINT fk_specializations_product

        FOREIGN KEY (product_id)

        REFERENCES products(product_id)

        ON DELETE CASCADE -- Удаляет специализацию при удалении товара

        ON UPDATE CASCADE

);

-- 6. Таблица 'repairs'

CREATE TABLE repairs (

    repair_id SERIAL PRIMARY KEY,

    product_id INTEGER NOT NULL,

    shop_id INTEGER NOT NULL,

    start_date DATE NOT NULL,

```

```

end_date DATE,

    actual_repair_cost DECIMAL(10,2) NOT NULL CHECK (actual_repair_cost >=
0),

    client_cost DECIMAL(10,2) NOT NULL CHECK (client_cost >= 0),

    serial_number VARCHAR(50),

    defect_description TEXT,


    CONSTRAINT chk_repair_dates CHECK (end_date IS NULL OR end_date >=
start_date),

    CONSTRAINT fk_repairs_product

        FOREIGN KEY (product_id)

            REFERENCES products(product_id)

            ON DELETE RESTRICT

            ON UPDATE CASCADE,

    CONSTRAINT fk_repairs_shop

        FOREIGN KEY (shop_id)

            REFERENCES repair_shops(shop_id)

            ON DELETE RESTRICT

            ON UPDATE CASCADE

);

```

2. С помощью SQL-запросов выполнить добавление 3–4 записей в каждую таблицу, изменение и удаление нескольких записей.

Добавление данных

Для каждой таблицы были добавлены тестовые данные, демонстрирующие различные сценарии:

В первой записи:

- Ремонт выполнен по гарантии (client_cost = 0)
- Указаны серийный номер и описание дефекта
- Ремонт завершен (указана end_date)

Во второй записи:

- Ремонт платный (client_cost = 12000.00)
- Затраты мастерской нулевые (возможно, ремонт субсидируется производителем)

Изменение данных

Важно отметить использование условия WHERE, которое точно идентифицирует изменяемую запись. Без этого условия произошло бы изменение всех записей в таблице.

Удаление данных

Удаляется конкретная специализация (мастерская 4 больше не ремонтирует холодильники), при этом остальные данные остаются нетронутыми.

```
-- Добавление городов

INSERT INTO cities (name) VALUES

    ('Москва'),

    ('Санкт-Петербург'),

    ('Новосибирск');

-- Добавление производителей

INSERT INTO manufacturers (name, contact_info) VALUES

    ('Xiaomi', 'Китай, Пекин, ул. Центральная, 1. Тел.: +86-10-1234567'),

    ('Samsung', 'Южная Корея, Сувон. Тел.: +82-31-200-1111'),

    ('Bosch', 'Германия, Штутгарт. Тел.: +49-711-811-0');
```

-- Добавление товаров

```
INSERT INTO products (manufacturer_id, model_name, category, warranty_period, base_repair_cost) VALUES
```

```
(1, 'Redmi Note 13', 'смартфоны', 365, 5000.00),  
(2, 'Galaxy S24', 'смартфоны', 730, 15000.00),  
(3, 'KGN39VI32R', 'холодильники', 1095, 12000.00),  
(1, 'Robot Vacuum X10', 'пылесосы', 180, 8000.00);
```

-- Добавление мастерских

```
INSERT INTO repair_shops (city_id, address, contact_phone) VALUES
```

```
(1, 'ул. Тверская, 25', '+7-495-111-22-33'),  
(1, 'пр-т Мира, 100', '+7-495-444-55-66'),  
(2, 'Невский пр-т, 50', '+7-812-777-88-99'),  
(3, 'ул. Ленина, 1', '+7-383-123-45-67');
```

-- Добавление специализаций мастерских

```
INSERT INTO shop_specializations (shop_id, product_id) VALUES
```

```
(1, 1), -- Мастерская 1 ремонтирует Xiaomi smartphones  
(1, 2), -- Мастерская 1 ремонтирует Samsung smartphones  
(2, 3), -- Мастерская 2 ремонтирует холодильники  
(3, 1), -- Мастерская 3 ремонтирует Xiaomi smartphones  
(3, 4), -- Мастерская 3 ремонтирует пылесосы  
(4, 2), -- Мастерская 4 ремонтирует Samsung smartphones  
(4, 3); -- Мастерская 4 ремонтирует холодильники
```

-- Добавление записей о ремонтах


```
INSERT INTO repairs (product_id, shop_id, start_date, end_date,
actual_repair_cost, client_cost, serial_number, defect_description) VALUES

    (1, 1, '2024-01-15', '2024-01-20', 3000.00, 0.00, 'XN13123456', 'Разбит
экран, гарантийный случай'),

    (2, 1, '2024-02-01', '2024-02-05', 0.00, 12000.00, 'GS24123456', 'Замена
аккумулятора, не гарантия'),

    (3, 2, '2024-02-10', NULL, 8500.00, 0.00, 'BOSCH98765', 'Утечка фреона,
гарантийный ремонт'),

    (4, 3, '2024-01-20', '2024-01-25', 5000.00, 5000.00, 'X10054321', 'Не
работает двигатель, постгарантия');

-- Изменение контактного телефона мастерской с shop_id = 2

UPDATE repair_shops

SET contact_phone = '+7-495-999-00-11'

WHERE shop_id = 2;

-- Увеличение базовой стоимости ремонта для всех смартфонов на 5%

UPDATE products

SET base_repair_cost = base_repair_cost * 1.05

WHERE category = 'смартфоны';

-- Завершение ремонта, у которого не была указана end_date

UPDATE repairs

SET end_date = '2024-02-15'

WHERE repair_id = 3;

-- Удаление специализации (например, мастерская 4 больше не ремонтирует
холодильники)

DELETE FROM shop_specializations
```

```
WHERE shop_id = 4 AND product_id = 3;

-- Удаление записи о ремонте (если она была создана ошибочно)

DELETE FROM repairs

WHERE repair_id = 4;
```

Вывод: изучили основные возможности языка SQL для создания структуры базы данных. Научились создавать базы данных, таблицы, связи, ограничения, а также создавать, изменять и удалять данные.