**ИТОГОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Название программы | **«Инженер-тестировщик»** |
| Группа обучения | **«ИТ-6-Р»** |
| Срок обучения | **«23.09.2025- 27.10.2025»** |
| **«‎Толстокорова Алёна Дмитриевна»** | |
| Номер/Название Кейса | **«Проверка основных ограничений в базе данных магазина»** |

Москва 2025г.

**Проверка основных ограничений в базе данных магазина**

**Цель работы** — разработать и протестировать набор SQL-запросов для проверки корректности работы ограничений целостности в базе данных PostgreSQL на примере модели магазина.

- Для достижения поставленной цели необходимо решить следующие задачи:  
изучить основные типы ограничений в PostgreSQL (CHECK, NOT NULL, FOREIGN KEY, ON DELETE);

- Разработать структуру базы данных, включающую таблицы «Категории» и «Товары»;

- Реализовать ограничения целостности на уровне таблиц;  
- Выполнить тестовые операции вставки, обновления и удаления данных для проверки корректности работы ограничений;

- Проанализировать возникающие ошибки, определить их причины и предложить способы устранения;

- Сделать вывод о надёжности механизмов целостности данных в PostgreSQL.

**Описание набора данных (датасета)**

В данной работе используется тестовая база данных магазина, состоящая из двух таблиц: categories (категории) и products (товары). Таблица categories хранит список категорий товаров, а таблица products — сведения о товарах и их связи с категориями.

Структура таблиц:

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | SERIAL | Уникальный идентификатор категории |
| name | VARCHAR(100) | Название категории (обязательно к заполнению) |

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | SERIAL | Уникальный идентификатор товара |
| name | VARCHAR(100) | Название товара (обязательно к заполнению) |
| price | NUMERIC(10,2) | Цена товара (должна быть положительной) |
| category\_id | INT | Внешний ключ на таблицу categories |

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

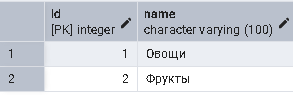
**Ссылки на файлы с выполненной работой**

* *SQL-скрипт создания базы и таблиц* — create\_tables.sql(<https://github.com/kusteg-69/Final_certification/blob/main/create_tables.sql>)
* *SQL-скрипт с тестированием ограничений* — test\_constraints.sql(<https://github.com/kusteg-69/Final_certification/blob/main/test_constraints.sql>)
* SQL-скрипт создания корректной таблицы products — correct\_table\_products.sql(*<https://github.com/kusteg-69/Final_certification/blob/main/correct_table_products.sql>*)
* Отчёт в формате Word — lab\_report\_postgresql.docx  
    
  Файлы могут быть сохранены в локальной директории проекта или опубликованы в репозитории GitHub.

**Этап 4. Добавление тестовых данных**

В качестве примера были добавлены две категории такие как: «Овощи», «Фрукты»





**Этап 5. Тестирование ограничений**

1. **Добавление товара с отрицательной ценой**



A close-up of a text

AI-generated content may be incorrect.

Операция «**выполнилась успешно»**, поскольку изначально ограничение CHECK отсутствовало, и база данных не проверяла корректность значения поля price. В данном случае это нарушает логику бизнесс-процесса.

*Решение:*

Для начала «обнуляем» тот товар, который уже был добавлен в бд. После чего обновляем атрибут price на ограничение.

A white background with purple text

AI-generated content may be incorrect.

После внесения данного изменения повторная вставка отрицательной цены вызывает ошибку:

A close up of text

AI-generated content may be incorrect.

Это подтверждает корректную работу ограничения **CHECK**.

1. **Добавление товара с несуществующей категорией**





*Пояснение:* сработало ограничение ссылочной целостности FOREIGN KEY, запрещающее добавление записи, если значение category\_id не существует в родительской таблице categories

*Решение:* предварительно добавить нужную категорию в таблицу categories или выбрать корректный идентификатор.

1. **Добавление товара без обязательных полей**



A close up of a text

AI-generated content may be incorrect.

*Пояснение:* сработало ограничение **NOT NULL**, которое не допускает вставку пустых значений в обязательное поле name

*Решение*: перед вставкой убедиться, что все обязательные поля (name, price, category\_id) заполнены

1. **Удаление категории с товарами**



A close-up of a text

AI-generated content may be incorrect.





*Пояснение:* PostgreSQL по умолчанию применяет поведение **ON DELETE NO ACTION**, которое предотвращает удаление записи из родительской таблицы, если на неё ссылаются строки из дочерней.

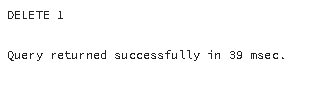
*Решение:* при необходимости можно изменить поведение связи

Для начала удаляем старый внешний ключ, после чего обновляем таблицу на каскадное удаление. В этом случае при удалении категории автоматически удаляются все связанные с ней товары.

A screenshot of a computer code

AI-generated content may be incorrect.

Результат:



A close-up of a computer screen

AI-generated content may be incorrect.

A close-up of a grey box

AI-generated content may be incorrect.

Была изменена структуру таблицы products, чтобы не допускались подобные ошибки.

Окончальный вариант таблицы products:

A computer code with text

AI-generated content may be incorrect.

**Вывод:**

В ходе работы была создана и протестирована простая база данных магазина в PostgreSQL, включающая таблицы *«Категории»* и *«Товары»*. Проведён набор практических тестов, направленных на проверку корректности работы базовых ограничений целостности данных.

Проверка показала корректное срабатывание ключевых механизмов PostgreSQL: обязательность заполнения полей (NOT NULL), обеспечение ссылочной целостности (FOREIGN KEY) и проверку корректности значений (CHECK). Все ошибки, возникшие при тестировании, соответствовали ожидаемому поведению системы и подтвердили надёжность встроенных механизмов защиты данных.

Внесённые изменения в структуру таблиц — добавление ограничения CHECK и удаление избыточной опции ON DELETE RESTRICT — позволили улучшить качество модели данных и сделать её более логичной и устойчивой к ошибкам ввода.