**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра ИС**

**ОТЧЕТ**

**по заданию №2-3**

**по дисциплине «Методы и средства проектирования информационных систем»**

**Тема: Задача 2-3. Приемы моделирования заказов на изделия с вариантами исполнения**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 1374 |  | Архипов Д.А.  Дюков Н.В.  Александр Х. |
| Преподаватель |  | Дубенецкий В.А. |

Санкт-Петербург

2024**Содержание**

[**1. Задание на выполнение работы 3**](#_3znysh7)

[1.1 Функциональные требования 3](#_2et92p0)

[1.2 Содержание работы 3](#_tyjcwt)

[1.3 Исходные данные для задачи 4](#_3dy6vkm)

[1.4 Используемые инструменты 4](#_1t3h5sf)

[1.5 Используемые документы 4](#_4d34og8)

[**2. Описание выполненной работы 5**](#_2s8eyo1)

[2.1 Анализ исходных данных 5](#_17dp8vu)

[2.2 Разработка функциональных требований к подсистеме 6](#_3rdcrjn)

[2.3 Разработка проектной модели классов 7](#_26in1rg)

[2.4 Разработка модели хранения в среде СУБД 7](#_z337ya)

[2.5 Создание базы данных 8](#_lnxbz9)

[2.6 Разработка основных процедур 11](#_35nkun2)

[2.7 Тестирование разработки 20](#_44sinio)

[**Вывод 24**](#_2jxsxqh)

**Задача №3**

**Приемы моделирования заказов на изделия с вариантами исполнения**

# Задание на выполнение работы

Разработать проект каркаса для работы с маршрутной

технологией изготовления изделий

## 1.1 Функциональные требования

Разработать каркас для процесса Ведение заказов на изделия с вариантами исполнения. Обеспечить:

* Настройку справочника изделий для работы с вариантами исполнения;
* Описание параметров конфигурирования изделий;
* Редактирование состава заказа с указанием вариантов исполнения заказываемых изделий;
* Вывод состава заказа с указанием вариантов исполнения изделий.

## 1.2 Содержание работы

|  |  |  |
| --- | --- | --- |
| **№** | **Этап выполнения** | **Результат** |
|  | Разработать функциональные требования к проектируемой подсистеме | Диаграммы вариантов использования. Описание компонентов диаграмм |
|  | Разработать модель классов для выделенных процессов | Диаграммы классов с операциями и атрибутами. Описание компонентов диаграмм. |
|  | Разработать модель хранения в среде СУБД для решения задач управления материальной спецификацией | Диаграммы ER. Скрипты метаданных с комментариями. |
|  | Разработка основных SQL-процедур для работы с классификатором | Скрипты SQL-процедур с комментариями. |
|  | Тестирование процедур | Описание тестов. Скрипты исходных данных для тестов.  Скриншоты результатов тестирования. |

## 1.3 Используемые инструменты

РедакторVisio. МодельDatabase.Entity relationship

IDE Datagrip

СУБДMicrosoft SQL Server

Текстовыйредактор Word

## 1.4 Используемые документы

1. Data Definition Guide : файл DataDef.pdf
2. Language Reference Guide: файл LangRef.pdf
3. Архитектура информационных систем. Учебник. М.,Издательский центр «Академия», 2012
4. Информационное обеспечение жизненного цикла изделий. Учебное пособие: СПб, Издательство Политехнического университета, 2012
5. Приемы объектно-ориентированного проектирования. Паттерны проектирования: Изд-во «ПИТЕР», 2008
6. Проектирование корпоративных информационных систем.: СПб, Изд-во СПбГЭТУ «ЛЭТИ», 2013

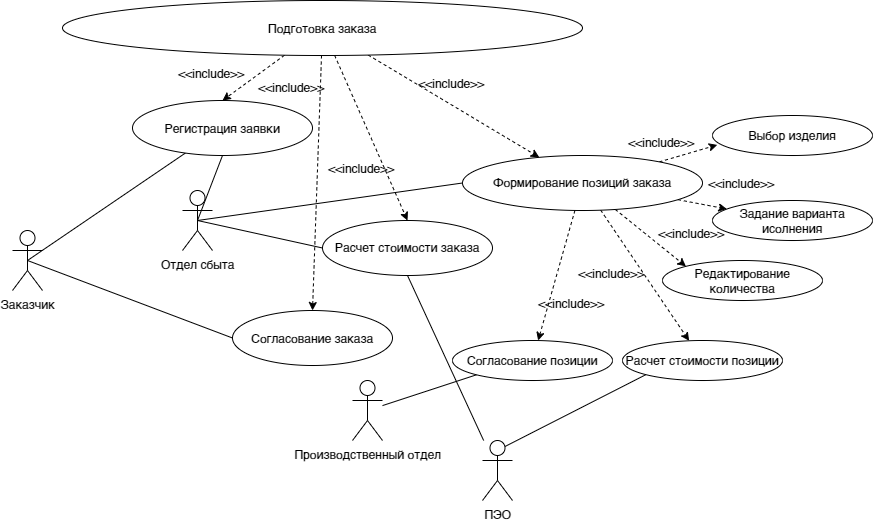
# Описание выполненной работы

## 2.1 Разработка функциональных требований к подсистеме

Для разработки концептуального проекта будем использовать редактор VISIO.

Создадим диаграмму вариантов использования с именем Работа c заказами изделий с вариантами исполнения.

На рис. 1 приведена диаграмма ВИ Работа c заказами изделий с вариантами исполнения.



*Рисунок 1 - Диаграмма ВИ для процесса Работа c заказами изделий с вариантами исполнения*

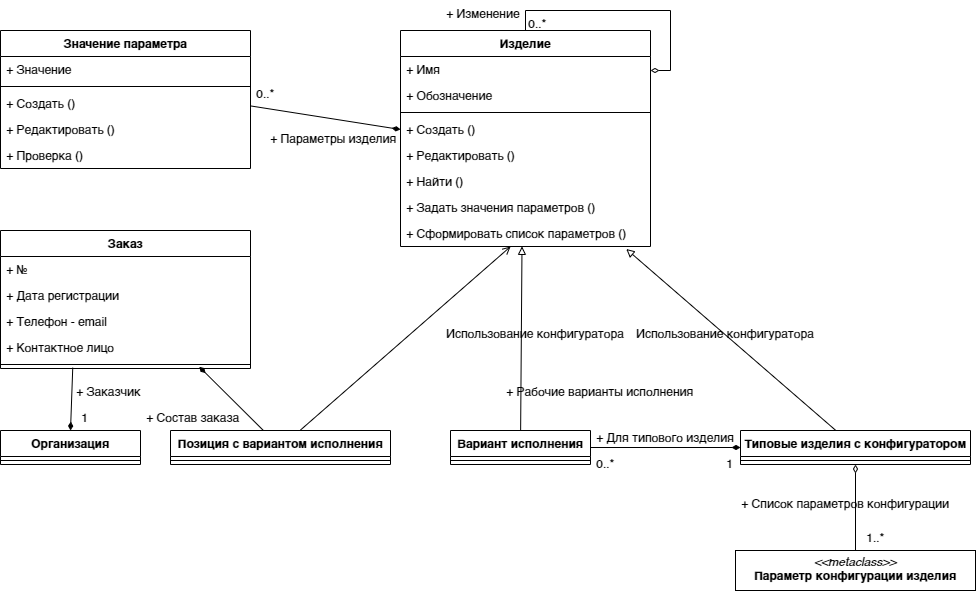
## 

## 

## 

## 2.2 Разработка проектной модели классов

Модель классов для ВИ Задание варианта исполнения

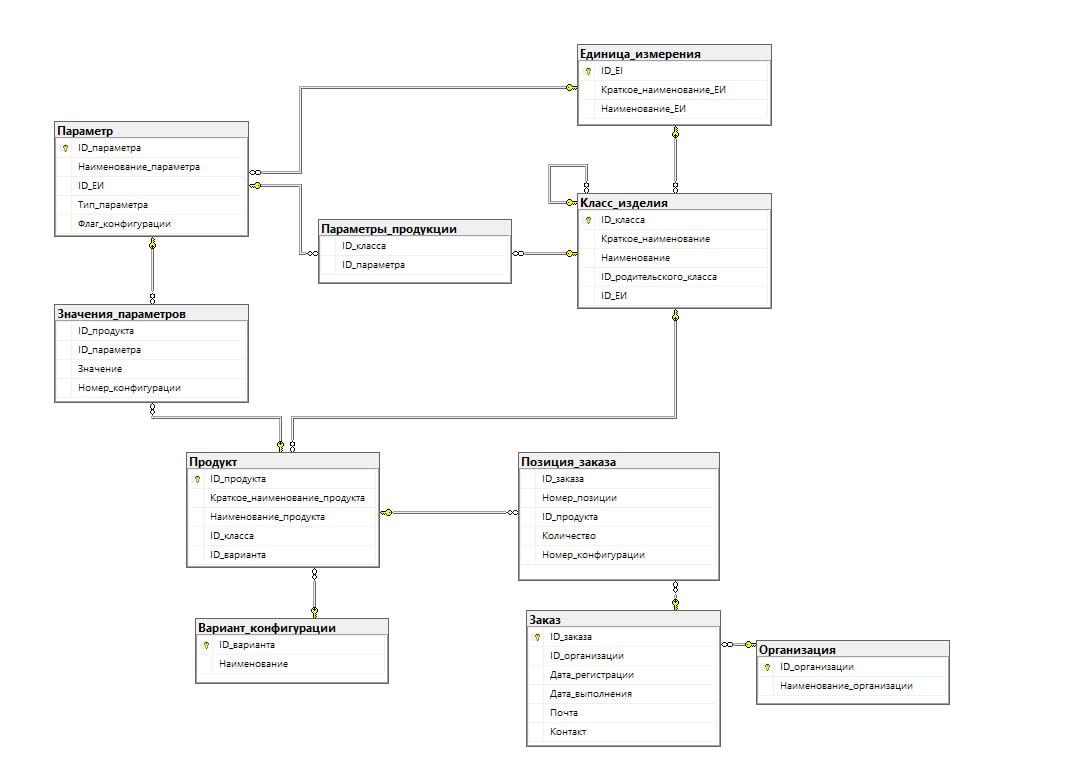


*Рисунок 2 - Модель классов для прецедента Формирование позиций заказа*

В состав заказа могут быть включены как изделия без конфигуратора, так и варианты исполнения. Компоновщик состава заказа должен проверять класс выбранного изделия. При выборе изделия класса *Типовое изделие с конфигуратором* должен создаваться вариант исполнения и указываться значения параметров конфигурации (роль *Список параметров конфигурации*).

## Разработка модели хранения

Для разработки диаграммы ER воспользуемся редактором VISIO. На рис. 4 представлена диаграмма, описывающая сущности и связи локальной модели хранения для прецедента *Задание варианта исполнения*.



*Рисунок 4 - Модель хранения для справочника изделий с вариантами исполнения*

## Создание базы данных

1. **Создание таблицы *Единицы измерения***

CREATE TABLE Единица\_измерения

(

ID\_EI INT IDENTITY(1,1) PRIMARY KEY,

Краткое\_наименование\_ЕИ NVARCHAR(50) NOT NULL,

Наименование\_ЕИ NVARCHAR(100) NOT NULL

);

1. **Создание таблицы *Класс изделия***

CREATE TABLE Класс\_изделия

(

ID\_класса INT IDENTITY(1,1) PRIMARY KEY,

Краткое\_наименование NVARCHAR(50) NOT NULL,

Наименование NVARCHAR(100) NOT NULL,

ID\_родительского\_класса INT NULL,

ID\_ЕИ INT,

CONSTRAINT FK\_ЕИ FOREIGN KEY (ID\_ЕИ)

REFERENCES Единица\_измерения (ID\_EI) ON UPDATE NO ACTION ON DELETE CASCADE,

CONSTRAINT FK\_ID\_родительского\_класса FOREIGN KEY (ID\_родительского\_класса)

REFERENCES Класс\_изделия (ID\_класса) ON UPDATE NO ACTION ON DELETE NO ACTION

);

1. **Создание таблицы *Вариант конфигурации***

CREATE TABLE Вариант\_конфигурации

(

ID\_варианта INT IDENTITY(1,1) PRIMARY KEY,

Наименование NVARCHAR(100) NOT NULL

);

1. **Создание таблицы *Организация***

CREATE TABLE Организация

(

ID\_организации INT IDENTITY(1,1) PRIMARY KEY,

Наименование\_организации NVARCHAR(255) NOT NULL

);

1. **Создание таблицы *Заказ***

CREATE TABLE Заказ

(

ID\_заказа INT IDENTITY(1,1) PRIMARY KEY,

ID\_организации INT,

Дата\_регистрации DATE,

Дата\_выполнения DATE,

Почта NVARCHAR(100),

Контакт NVARCHAR(100),

CONSTRAINT FK\_ID\_организации FOREIGN KEY (ID\_организации)

REFERENCES Организация (ID\_организации) ON UPDATE NO ACTION ON DELETE CASCADE

);

1. **Создание таблицы *Позиции заказа***

CREATE TABLE Позиция\_заказа

(

ID\_заказа INT,

Номер\_позиции INT,

ID\_продукта INT,

Количество INT,

Номер\_конфигурации INT,

CONSTRAINT FK\_ID\_заказа FOREIGN KEY (ID\_заказа)

REFERENCES Заказ (ID\_заказа) ON UPDATE NO ACTION ON DELETE CASCADE,

CONSTRAINT FK\_ID\_продукта FOREIGN KEY (ID\_продукта)

REFERENCES Продукт (ID\_продукта) ON UPDATE NO ACTION ON DELETE CASCADE

);

1. **Создание таблицы *Параметр***

CREATE TABLE Параметр

(

ID\_параметра INT IDENTITY(1,1) PRIMARY KEY,

Наименование\_параметра NVARCHAR(100),

ID\_ЕИ INT,

Тип\_параметра NVARCHAR(50),

Флаг\_конфигурации BIT,

CONSTRAINT FK\_ID\_ЕИ FOREIGN KEY (ID\_ЕИ)

REFERENCES Единица\_измерения (ID\_EI) ON UPDATE NO ACTION ON DELETE CASCADE

);

1. **Создание таблицы *Параметры продукции***

CREATE TABLE Параметры\_продукции

(

ID\_класса INT,

ID\_параметра INT,

CONSTRAINT FK\_Параметры\_класса FOREIGN KEY (ID\_класса)

REFERENCES Класс\_изделия (ID\_класса) ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT FK\_Параметры\_параметра FOREIGN KEY (ID\_параметра)

REFERENCES Параметр (ID\_параметра) ON UPDATE NO ACTION ON DELETE NO ACTION

);

1. **Создание таблицы *Продукт***

CREATE TABLE Продукт

(

ID\_продукта INT IDENTITY(1,1) PRIMARY KEY,

Краткое\_наименование\_продукта NVARCHAR(100) NOT NULL,

Наименование\_продукта NVARCHAR(255) NOT NULL,

ID\_класса INT NOT NULL,

ID\_варианта INT,

CONSTRAINT FK\_ID\_класса FOREIGN KEY (ID\_класса)

REFERENCES Класс\_изделия (ID\_класса) ON UPDATE NO ACTION ON DELETE CASCADE,

CONSTRAINT FK\_ID\_варианта FOREIGN KEY (ID\_варианта)

REFERENCES Вариант\_конфигурации (ID\_варианта) ON UPDATE NO ACTION ON DELETE CASCADE

);

1. **Создание таблицы *Значения параметров***

CREATE TABLE Значения\_параметров

(

ID\_продукта INT,

ID\_параметра INT,

Значение FLOAT,

Номер\_конфигурации INT,

CONSTRAINT FK\_Продукта\_Значения FOREIGN KEY (ID\_продукта)

REFERENCES Продукт (ID\_продукта) ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT FK\_Параметра\_Значения FOREIGN KEY (ID\_параметра)

REFERENCES Параметры (ID\_параметра) ON UPDATE NO ACTION ON DELETE NO ACTION

);

## Разработка основных процедур

1. **Добавление единицы измерения**

CREATE PROCEDURE [dbo].[Добавить\_единицу\_измерения]

*-- Функция: Добавление новой единицы измерения*

@краткое\_наименование NVARCHAR(50)

@полное\_наименование NVARCHAR(100)

*-- Выход: Нет*

*-- Эффекты: Вставляет новую запись в таблицу Единица\_измерения с указанными наименованиями*

AS

BEGIN

INSERT INTO Единица\_измерения (Краткое\_наименование\_ЕИ, Наименование\_ЕИ)

VALUES (@краткое\_наименование, @полное\_наименование);

END;

1. **Добавление классов в классификатор**

CREATE PROCEDURE [dbo].[Добавить\_класс]

*-- Функция: Добавление нового класса изделия*

@краткое\_наименование NVARCHAR(50),

@полное\_наименование NVARCHAR(100),

@родительский\_класс INT,

@единица\_измерения INT

*-- Выход: Нет*

*-- Эффекты: Вставляет новую запись в таблицу Класс\_изделия с указанными данными*

AS

BEGIN

INSERT INTO Класс\_изделия (Краткое\_наименование, Наименование, ID\_родительского\_класса, ID\_ЕИ)

VALUES (@краткое\_наименование, @полное\_наименование, @родительский\_класс, @единица\_измерения);

END;

1. **Добавление продукта**

CREATE PROCEDURE [dbo].[Добавить\_продукт]

*-- Функция: Добавление нового продукта*

@краткое\_наименование\_продукта NVARCHAR(50),

@полное\_наименование\_продукта NVARCHAR(100),

@id\_класса INT,

@id\_варианта INT

*-- Выход: Нет*

*-- Эффекты: Вставляет новую запись в таблицу Продукт с указанными данными*

AS

BEGIN

INSERT INTO Продукт (Краткое\_наименование\_продукта, Наименование\_продукта, ID\_класса, ID\_варианта)

VALUES (@краткое\_наименование\_продукта, @полное\_наименование\_продукта, @id\_класса, @id\_варианта);

END;

1. **Настройка справочника изделий для работы с вариантами исполнения**

CREATE PROCEDURE [dbo].[Добавить\_параметры\_и\_значения]

*-- Функция: Добавление параметров и их значений для указанного продукта*

@id\_продукта INT,

@номер\_конфигурации INT,

@id\_параметра1 INT, @значение1 FLOAT,

@id\_параметра2 INT, @значение2 FLOAT,

@id\_параметра3 INT, @значение3 FLOAT

AS

BEGIN

*-- Выход: Нет*

*-- Эффекты: Добавляет записи в таблицу Значения\_параметров для указанного продукта и конфигурации,*

*-- При условии, что продукт имеет конфигуратор и конфигурация с данным номером не существует*

IF (SELECT ID\_варианта FROM Продукт WHERE ID\_продукта = @id\_продукта) = 1

BEGIN

PRINT 'Невозможно выполнить операцию. Продукт без конфигуратора.';

RETURN;

END

*-- Проверка, существует ли уже конфигурация с данным номером*

IF EXISTS (SELECT Номер\_конфигурации FROM Значения\_параметров WHERE ID\_продукта = @id\_продукта AND Номер\_конфигурации = @номер\_конфигурации)

BEGIN

PRINT 'Невозможно выполнить операцию. Конфигурация с данным номером уже создана.';

RETURN;

END

*-- Добавление параметров и значений*

INSERT INTO Значения\_параметров (ID\_продукта, ID\_параметра, Номер\_конфигурации, Значение)

VALUES (@id\_продукта, @id\_параметра1, @номер\_конфигурации, @значение1),

(@id\_продукта, @id\_параметра2, @номер\_конфигурации, @значение2),

(@id\_продукта, @id\_параметра3, @номер\_конфигурации, @значение3);

PRINT 'Конфигурация успешно добавлена';

END;

1. **Описание параметров конфигурирования изделий**

CREATE PROCEDURE [dbo].[Получить\_параметры\_для\_конфигурации]

*-- Функция: Получение параметров и их значений для указанной конфигурации продукта*

@id\_продукта INT,

@номер\_конфигурации INT

*-- Выход: Набор данных, содержащий наименование продукта, параметры, их значения и единицы измерения*

*-- Эффекты: Возвращает параметры и их значения для заданного продукта и конфигурации*

AS

BEGIN

SELECT p.Наименование\_продукта AS Наименование\_продукта,

par.Наименование\_параметра AS Наименование\_параметра,

pv.Значение AS Значение\_параметра,

ei.Краткое\_наименование\_ЕИ AS Единица\_измерения

FROM Значения\_параметров pv

JOIN Продукт p ON pv.ID\_продукта = p.ID\_продукта

JOIN Параметр par ON pv.ID\_параметра = par.ID\_параметра

JOIN Единица\_измерения ei ON par.ID\_ЕИ = ei.ID\_EI

WHERE pv.ID\_продукта = @id\_продукта

AND pv.Номер\_конфигурации = @номер\_конфигурации;

END;

1. **Редактирование состава заказа**

CREATE PROCEDURE [dbo].[Обновить\_заказ]

*-- Функция: Обновление позиции заказа или добавление новой, если она не существует*

@id\_заказа INT,

@номер\_позиции INT,

@prod\_старый INT, @config\_старый INT,

@prod\_новый INT, @config\_новый INT, @количество\_новое INT

*-- Выход: Нет*

*-- Эффекты: Обновляет существующую позицию заказа или добавляет новую позицию*

AS

BEGIN

*-- Проверка существования заказа*

IF NOT EXISTS (SELECT id\_заказа FROM Заказ WHERE id\_заказа = @id\_заказа)

BEGIN

PRINT 'Невозможно выполнить операцию. Заказа с таким номером не существует.';

RETURN;

END

*-- Проверка существования позиции в заказе*

IF NOT EXISTS (SELECT Номер\_позиции FROM Позиция\_заказа WHERE Номер\_позиции = @номер\_позиции AND ID\_заказа = @id\_заказа)

BEGIN

INSERT INTO Позиция\_заказа (ID\_заказа, Номер\_позиции, ID\_продукта, Количество, Номер\_конфигурации)

VALUES (@id\_заказа, @номер\_позиции, @prod\_новый, @количество\_новое, @config\_новый);

PRINT 'В заказе создана новая позиция.';

RETURN;

END

*-- Обновление существующей позиции заказа*

UPDATE Позиция\_заказа

SET ID\_продукта = @prod\_новый, Количество = @количество\_новое, Номер\_конфигурации = @config\_новый

WHERE ID\_продукта = @prod\_старый AND Номер\_конфигурации = @config\_старый AND ID\_заказа = @id\_заказа AND Номер\_позиции = @номер\_позиции;

PRINT 'Позиция заказа успешно обновлена.';

END;

1. **Вывод состава заказа с указанием вариантов исполнения изделий**

CREATE PROCEDURE [dbo].[Показать\_состав\_заказа]

*-- Функция: Получение состава заказа с детализированной информацией о позициях и параметрах*

@id\_заказа INT

*-- Выход: Набор данных с информацией о номерах позиций, продуктах, конфигурациях, параметрах и их значениях*

*-- Эффекты: Возвращает детализированную информацию о составе заказа*

AS

BEGIN

SELECT op.Номер\_позиции AS номер\_позиции,

p.Наименование\_продукта AS имя\_продукта,

op.Номер\_конфигурации AS номер\_конфигурации,

pr.Наименование\_параметра AS имя\_параметра,

zp.Значение AS значение\_параметра,

ei.Краткое\_наименование\_ЕИ AS единица\_измерения,

op.Количество AS количество

FROM Позиция\_заказа op

JOIN Продукт p ON op.ID\_продукта = p.ID\_продукта

LEFT JOIN Значения\_параметров zp ON op.ID\_продукта = zp.ID\_продукта AND op.Номер\_конфигурации = zp.Номер\_конфигурации

LEFT JOIN Параметр pr ON zp.ID\_параметра = pr.ID\_параметра

LEFT JOIN Единица\_измерения ei ON pr.ID\_ЕИ = ei.ID\_EI

WHERE op.ID\_заказа = @id\_заказа

UNION

SELECT op2.Номер\_позиции, p2.Наименование\_продукта, NULL, NULL, NULL, ei2.Краткое\_наименование\_ЕИ, op2.Количество

FROM Позиция\_заказа op2

JOIN Продукт p2 ON op2.ID\_продукта = p2.ID\_продукта

JOIN Класс\_изделия ci ON p2.ID\_класса = ci.ID\_класса

JOIN Единица\_измерения ei2 ON ci.ID\_ЕИ = ei2.ID\_EI

WHERE op2.ID\_заказа = @id\_заказа AND op2.Номер\_конфигурации IS NULL;

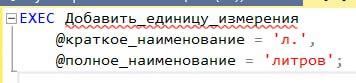
END;

## Тестирование разработки

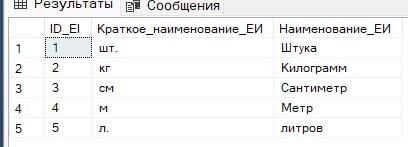
1. Добавление данных в таблицы



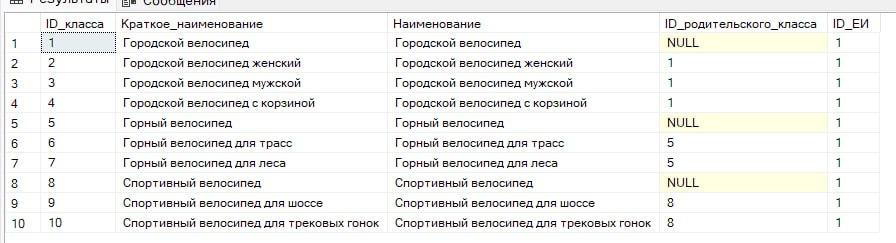
*Таблица 1 – Единицы измерения*

Добавление единицы измерения.  


*Команда 1 – Функция добавления единицы измерений*

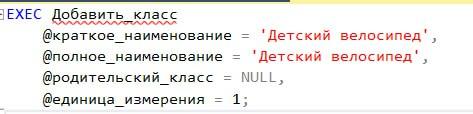


*Таблица 2 – Новые единицы измерения*



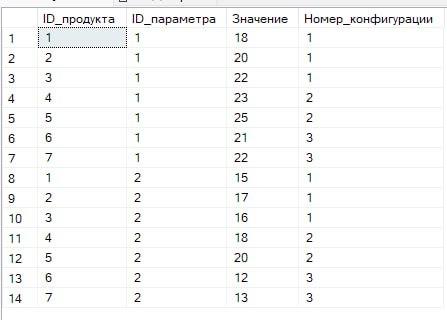
*Таблица 3 – Классы изделий*

Добавление класса изделия



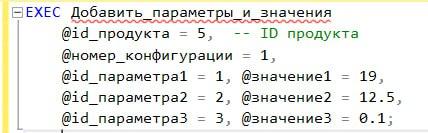
*Команда 2 – Функция добавления класса изделия*

*Таблица 4 – Новые классы изделий*

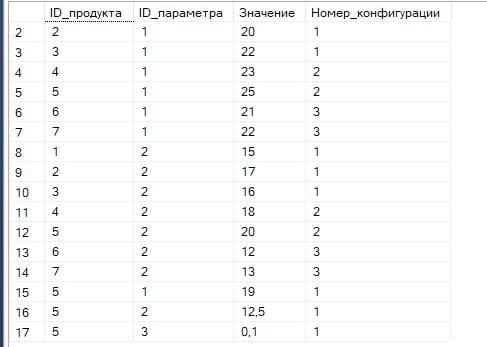


*Таблица 5 – Параметры и значения*

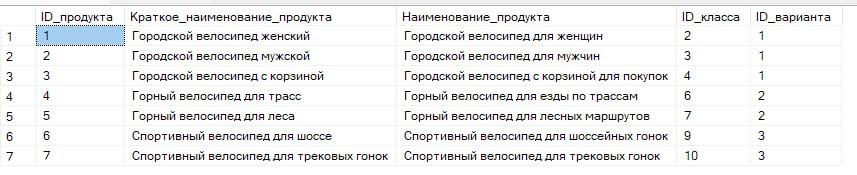
Добавление параметров и значений



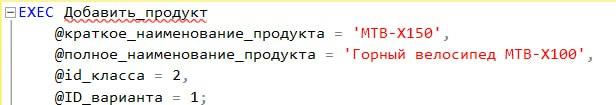
*Команда 3 –Функция добавления параметров и значенией*



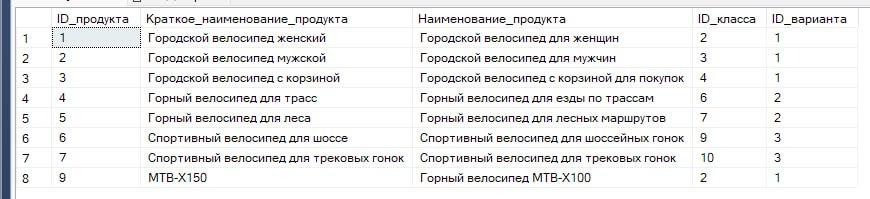
*Таблица 6 – Новые параметры и значниея*



*Таблица 7 – Продукты*

Добавление продукта  


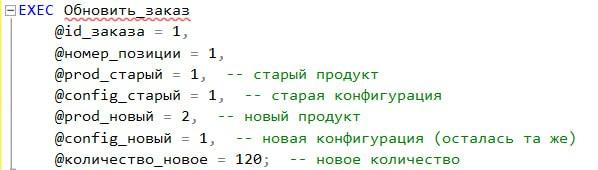
*Команда 4 –Функция добавления продукта*



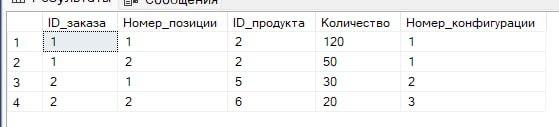
*Таблица 8 – Новые продукты*



*Таблица 9 – Заказы*

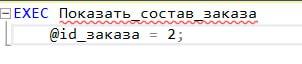
Обновление заказа  


*Команда 5 – Функция обновления заказа*

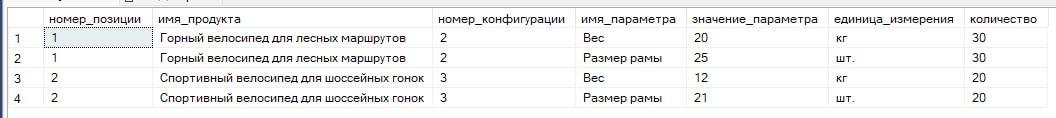


*Таблица 10 –Новые заказы*

Показать состав заказа

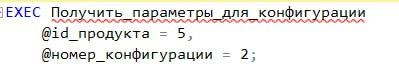


*Команда 6 – Функция показа состава заказа*

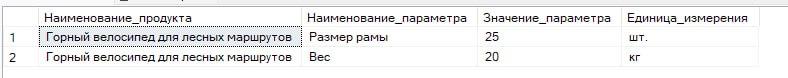


*Таблица 11 – Состав заказа*

Показать параметры для конфигурации



*Команда 7 – Функция показа параметров для конфигурации*



*Таблица 12 - Параметры для конфигурации*

# Вывод

В данной работе был разработан проект каркаса для процесса Ведение заказов на изделия с вариантами исполнения.

Реализация произведена в базе данных PostgreSQL. Проведено тестирование, показывающее корректную работу реализованных функций для работы с данными.