**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра ИС**

**ОТЧЕТ**

**по заданию №2-4**

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

**Тема: Задача 2-4. Приемы моделирования Спецификации изделий с вариантами исполнения**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1374 |  | Архипов Д.А.  Дюков Н.В.  Хлебников А. |
| Преподаватель |  | Дубенецкий В.А. |

Санкт-Петербург

2024

**Содержание**

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

[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 5](#_4d34og8)

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

[2.1 Разработка функциональных требований к подсистеме 5](#_17dp8vu)

[2.2 Разработка проектной модели классов 7](#_3rdcrjn)

[2.3 Разработка модели хранения 9](#_26in1rg)

[2.4 Создание базы данных 10](#_lnxbz9)

[2.5 Разработка основных процедур 17](#_35nkun2)

[2.6 Тестирование разработки 32](#_1ksv4uv)

[**Вывод 38**](#_2jxsxqh)

**Задача №4**

**Приемы моделирования Спецификации изделий (Billofmaterials) с вариантами исполнения**

# Задание на выполнение работы

## 1.1 Функциональные требования

Обеспечить:

* Ведение справочника изделий различных типов, используемых в составе изделий;
* Формирование строк спецификаций с указанием норм расхода;
* Формирование состава параметров конфигурации изделия;
* Описание правил управления конфигурацией на основе значений параметров конфигурации;
* Поиск всех строк спецификации изделия на всю глубину вложенности для варианта исполнения;
* Расчет сводных норм расхода компонентов изделия для заданного варианта исполнения по ресурсам заданного класса.

## 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

СУБД Microsoft SQL Server

Текстовый редактор Word

## 1.4 Используемые документы

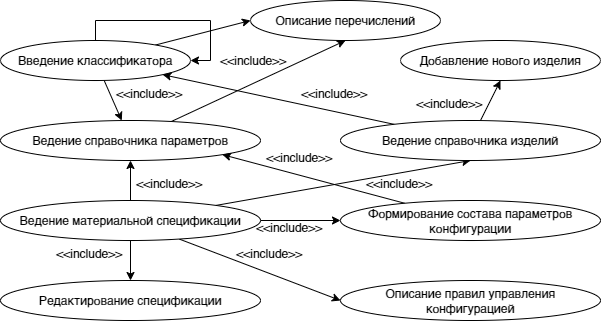
1. Data Definition Guide : файл DataDef.pdf
2. Language Reference Guide: файл LangRef.pdf
3. Архитектура информационных систем. Учебник. М.,Издательский центр «Академия», 2012
4. Информационное обеспечение жизненного цикла изделий. Учебное пособие: СПб, Издательство Политехнического университета, 2012
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

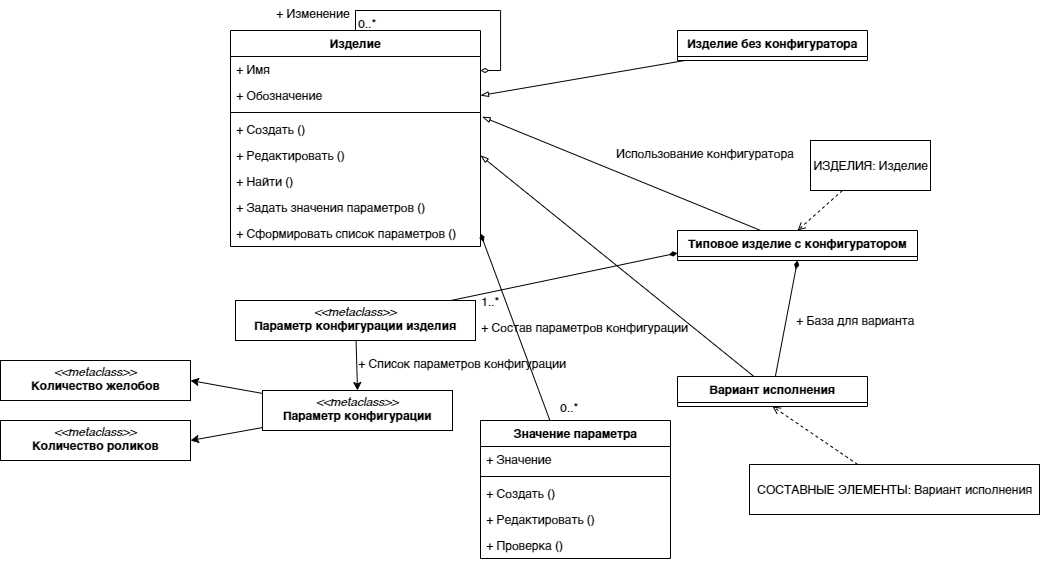
Для разработки концептуального проекта будем использовать редактор VISIO.

Создадим диаграмму вариантов использования с именем *Работа со спецификацией изделий с вариантами исполнения.*



*Рисунок 1 - Расширение модели прецедентов для решения задач по работе с конфигурациями изделий*

Добавлены прецеденты Формирование параметров конфигурации и Описание правил управления конфигурацией.



*Рисунок 2 – Модель классов для описания спецификаций изделий примера*

Рассматривается конкретное типовое изделие ИЗДЕЛИЯ: Изделие, для которого выделены несколько параметров конфигурации. Для этих параметров могут быть введены различные варианты значений, такие как Единица измерения и Класс изделия.

Конкретный вариант исполнения может быть создан на основе элемента ИЗДЕЛИЯ: Продукт. Для этого продукта указываются параметры конфигурации и значения параметров, такие как ID продукта, Краткое наименование продукта, и Наименование продукта.

Композиция продукта включает в себя различные параметры, которые записываются в таблице Параметр и значения, обозначенные в таблице Значения параметров. Здесь отображаются конкретные параметры, такие как Тип параметра и Флаг конфигурации.

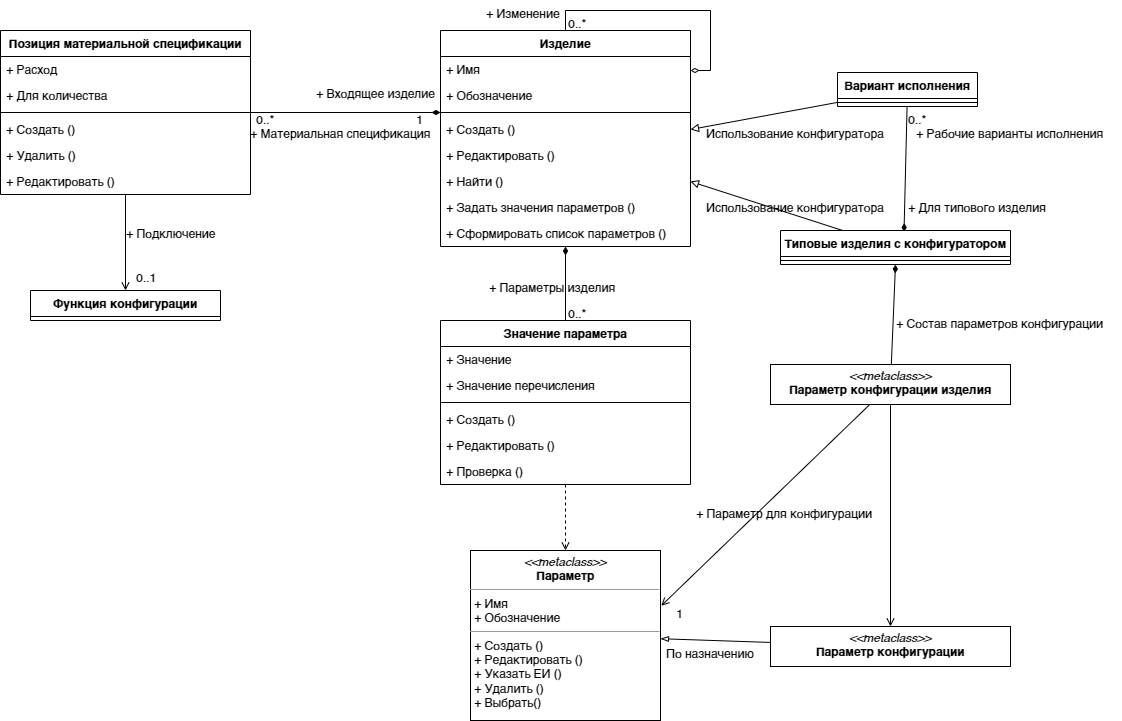
Настройки для этих параметров управляются с помощью таблицы Параметры продукции, указывающей принадлежность параметров к определенному классу изделий и связывающей их с конкретными параметрами конфигурации.

Кроме того, возможно указать конфигурацию на уровне Вариант конфигурации, где определяется конкретный подход к реализации изделия. Организация и выполнение заказов на продукцию управляются с помощью Позиция заказа и Заказ, включая такие важные элементы как Дата регистрации, Дата выполнения, и Контакт.

Наконец, каждый заказ связывается с Организацией, что позволяет управлять коммерческими и техническими аспектами заказов и их выполнением.

## 2.2 Разработка проектной модели классов

Расширенная модель классов для прецедента *Ведение спецификации изделий* представлена на рис. 3.

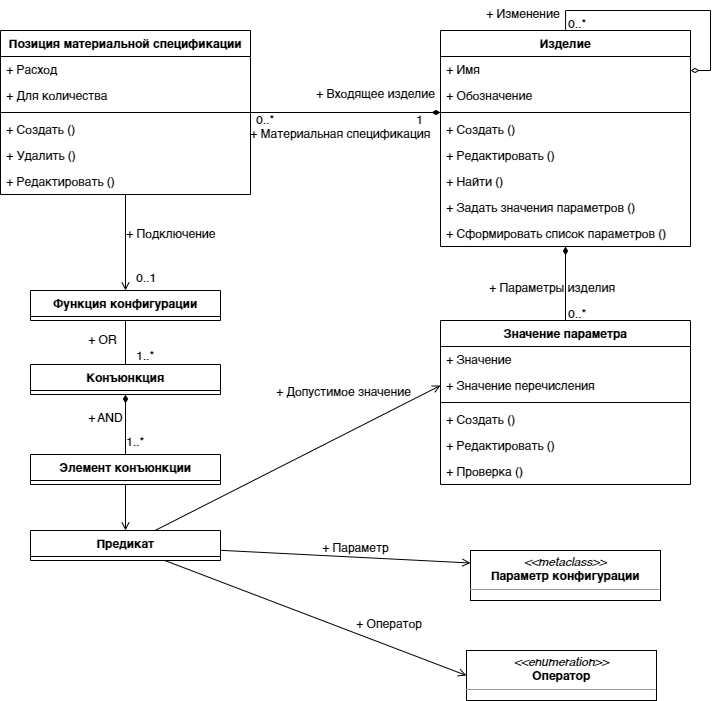
**

*Рисунок 3 - Модель классов для прецедента Ведение спецификации изделий*

В модель включен класс *Функция конфигурации* и ассоциация с ролью *Подключение*.

При наличии функции у позиции спецификации эта позиция подключается только, если соответствующая функция принимает значение *true* для заданных значений параметров конфигурации.

Модель классов для прецедента *Описание правил управления конфигурации* представлена на рис. 4.



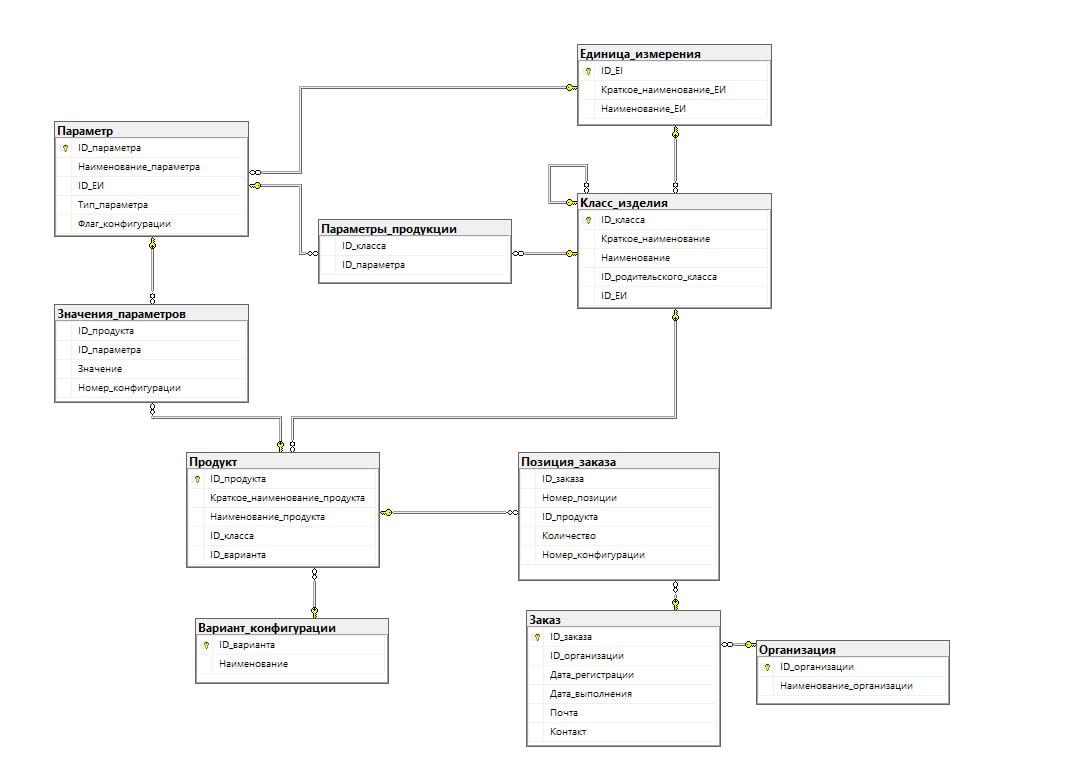
*Рисунок 4 - Модель классов для описания функций конфигурирования*

Функция конфигурации представляется в СДНФ. Класс *Предикат* позволяет описывать простые предикаты с указанием тройки (*Параметр, Оператор, Допустимое значение*).

Каждая конъюнкция описывается как список предикатов (класс *Конъюнкция*, ассоциативная сущность *Элемент конъюнкции* и ассоциация *AND*). Каждая функция описывается в виде списка конъюнкций (класс *Конъюнкция*, ассоциация *OR*).

## Разработкамодели хранения

Для разработки диаграммы ER воспользуемся редактором VISIO. На рис. 5 представлена диаграмма, описывающая сущности и связи модели хранения.



*Рисунок 5 - ERD для прецедента Формирование правил управления конфигурацией*

## Создание базы данных

1. **Создание таблицы единицы измерения**

CREATE TABLE Единицы\_измерения

(

ID\_ЕИ INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Краткое\_название\_ЕИ NVARCHAR(MAX) NOT NULL,

Название\_ЕИ NVARCHAR(MAX) NOT NULL

);

1. **Создание таблицы Класс изделия**

CREATE TABLE Класс\_изделия

(

ID\_класса INT IDENTITY(1,1) NOT NULL 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 Единицы\_измерения (ID\_ЕИ) 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 );

**3. Создание таблицы Перечисления**

CREATE TABLE Перечисления

(

ID\_перечисления INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Название\_перечисления NVARCHAR(MAX)

);

**4. Создание таблицы Продукция**

CREATE TABLE Продукция

(

ID\_продукции INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Название\_продукции NVARCHAR(MAX) NOT NULL,

ID\_класса INT NOT NULL,

CONSTRAINT FK\_ID\_класса FOREIGN KEY (ID\_класса) REFERENCES Класс\_изделия (ID\_класса) ON UPDATE NO ACTION ON DELETE CASCADE

);

**5. Создание таблицы Параметр**

CREATE TABLE Параметры

(

ID\_параметра INT PRIMARY KEY,

Название\_параметра NVARCHAR(MAX),

ЕИ\_параметра INT NULL,

Тип\_параметра NVARCHAR(MAX),

Флаг\_конфигурации BIT,

CONSTRAINT FK\_ЕИ\_параметра FOREIGN KEY (ЕИ\_параметра) REFERENCES Единицы\_измерения (ID\_ЕИ) ON UPDATE NO ACTION ON DELETE CASCADE

);

**6. Создание таблицы Значения перечислений**

CREATE TABLE Значения\_перечислений

(

ID\_значения INT PRIMARY KEY,

Номер\_значения INT,

ID\_перечисления INT,

Название\_значения NVARCHAR(MAX),

Значение INT,

CONSTRAINT FK\_ID\_перечисления\_Значения FOREIGN KEY (ID\_перечисления) REFERENCES Перечисления (ID\_перечисления) ON UPDATE NO ACTION ON DELETE CASCADE );

**7. Создание таблицы Параметры перечисления**

CREATE TABLE Параметры\_перечисления

(

ID\_параметра INT,

ID\_перечисления INT,

CONSTRAINT FK\_ID\_перечисления\_Параметры FOREIGN KEY (ID\_перечисления) REFERENCES Перечисления (ID\_перечисления) ON UPDATE NO ACTION ON DELETE CASCADE,

CONSTRAINT FK\_ID\_параметра FOREIGN KEY (ID\_параметра) REFERENCES Параметры (ID\_параметра) ON UPDATE NO ACTION ON DELETE CASCADE

);

**8. Создание таблицы Значения параметров**

CREATE TABLE Значения\_параметров

(

ID\_продукции INT,

ID\_параметра INT,

Значение FLOAT,

Номер\_конфигурации INT,

CONSTRAINT FK\_ID\_продукции FOREIGN KEY (ID\_продукции) REFERENCES Продукция (ID\_продукции) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT FK\_ID\_параметра\_Значения FOREIGN KEY (ID\_параметра) REFERENCES Параметры (ID\_параметра) ON DELETE NO ACTION ON UPDATE NO ACTION

);

**9. Создание таблицы Функция**

CREATE TABLE Функции

(

ID\_функции INT PRIMARY KEY,

Название\_функции NVARCHAR(MAX)

);

**10. Создание таблицы Предикат**

CREATE TABLE Предикаты

(

ID\_предиката INT PRIMARY KEY,

Оператор NVARCHAR(MAX),

ID\_параметра INT,

ID\_значения INT,

CONSTRAINT FK\_ID\_параметра\_Предикаты FOREIGN KEY (ID\_параметра) REFERENCES Параметры (ID\_параметра) ON UPDATE NO ACTION ON DELETE CASCADE,

CONSTRAINT FK\_ID\_значения\_Предикаты FOREIGN KEY (ID\_значения) REFERENCES Значения\_перечислений (ID\_значения) ON UPDATE NO ACTION ON DELETE CASCADE

);

**11. Создание таблицы Тело функции**

CREATE TABLE Тело\_функции

(

ID\_функции INT,

Номер\_устройства INT,

Номер\_связи INT,

ID\_предиката INT,

CONSTRAINT FK\_ID\_функции\_Тело FOREIGN KEY (ID\_функции) REFERENCES Функции (ID\_функции) ON UPDATE NO ACTION ON DELETE CASCADE,

CONSTRAINT FK\_ID\_предиката\_Тело FOREIGN KEY (ID\_предиката) REFERENCES Предикаты (ID\_предиката) ON UPDATE NO ACTION ON DELETE CASCADE

);

**12. Создание таблицы Спецификация продукции**

CREATE TABLE Спецификация\_продукции

(

ID\_продукции\_общей INT,

Номер\_позиции INT,

ID\_продукции\_части INT,

Количество INT,

ID\_функции INT,

CONSTRAINT FK\_ID\_продукции\_общей FOREIGN KEY (ID\_продукции\_общей)

REFERENCES Продукция (ID\_продукции) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT FK\_ID\_продукции\_части FOREIGN KEY (ID\_продукции\_части)

REFERENCES Продукция (ID\_продукции) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT FK\_ID\_функции\_Спецификация FOREIGN KEY (ID\_функции)

REFERENCES Функции (ID\_функции) ON DELETE NO ACTION ON UPDATE NO ACTION

);

## Разработка основных процедур

1. **Процедура добавления продукта**

CREATE PROCEDURE [dbo].[Добавить\_продукт]

*-- Функция: Добавление нового продукта в базу данных*

*-- Вход:*

-- @short\_prod\_name NVARCHAR(MAX) - Краткое название продукта

-- @full\_prod\_name NVARCHAR(MAX) - Полное название продукта

-- @id\_class INT - Идентификатор класса продукта

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Продукция*

AS

BEGIN

INSERT INTO Продукция (Название\_продукции, ID\_класса)

VALUES (@short\_prod\_name, @id\_class);

END;

1. **Процедура добавления единицы измерения**

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

-- Функция: Добавление новой единицы измерения в базу данных

-- Вход:

-- @short\_name NVARCHAR(MAX) - Краткое название единицы измерения

-- @full\_name NVARCHAR(MAX) - Полное название единицы измерения

-- Выход: Нет

-- Эффекты: Добавляет новую запись в таблицу Единицы\_измерения

@short\_name NVARCHAR(MAX),

@full\_name NVARCHAR(MAX)

AS

BEGIN

INSERT INTO Единицы\_измерения (Краткое\_название\_ЕИ, Название\_ЕИ)

VALUES (@short\_name, @full\_name);

END;

1. **Процедура добавления значения нового параметра**

CREATE PROCEDURE [dbo].[Добавить\_значение\_параметра]

*-- Функция: Добавление нового значения параметра для продукта в определенной конфигурации*

*-- Вход:*

*-- @prod\_id INT - Идентификатор продукции*

*-- @parameter\_id INT - Идентификатор параметра*

*-- @val FLOAT - Значение параметра*

*-- @config INT - Номер конфигурации*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Значения\_параметров*

AS

BEGIN

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

VALUES (@prod\_id, @parameter\_id, @val, @config);

END;

1. **Процедура добавление значения перечисления**

CREATE PROCEDURE [dbo].[Добавить\_значение\_перечисления]

*-- Функция: Добавление нового значения в перечисление*

*-- Вход:*

*-- @id\_val INT - Идентификатор значения*

*-- @id\_enum INT - Идентификатор перечисления*

*-- @number\_val INT - Номер значения в перечислении*

*-- @name\_val NVARCHAR(MAX) - Название значения*

*-- @value\_val INT - Само значение*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Значения\_перечислений*

AS

BEGIN

INSERT INTO Значения\_перечислений (ID\_значения, Номер\_значения, ID\_перечисления, Название\_значения, Значение)

VALUES (@id\_val, @number\_val, @id\_enum, @name\_val, @value\_val);

END;

1. **Процедура добавления нового заказа**

CREATE PROCEDURE [dbo].[Добавить\_класс]

*-- Функция: Добавление нового класса изделий в базу данных*

*-- Вход:*

*-- @class\_short\_name NVARCHAR(50) - Краткое наименование класса*

*-- @full\_name NVARCHAR(100) - Полное наименование класса*

*-- @parent INT - Идентификатор родительского класса (NULL, если родитель отсутствует)*

*-- @ei INT - Идентификатор единицы измерения*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Класс\_изделия*

AS

BEGIN

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

VALUES (@class\_short\_name, @full\_name, @parent, @ei);

END;

1. **Процедура добавления параметра**

CREATE PROCEDURE [dbo].[Добавить\_параметр]

*-- Функция: Добавление нового параметра*

*-- Вход:*

*-- @id\_parameter INT - Идентификатор параметра*

*-- @name\_parameter NVARCHAR(MAX) - Название параметра*

*-- @ei\_parameter INT - Идентификатор единицы измерения для параметра*

*-- @type\_parameter NVARCHAR(MAX) - Тип параметра*

*-- @flag\_config BIT - Флаг наличия конфигурации (0 или 1)*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Параметры*

AS

BEGIN

INSERT INTO Параметры (ID\_параметра, Название\_параметра, ЕИ\_параметра, Тип\_параметра, Флаг\_конфигурации)

VALUES (@id\_parameter, @name\_parameter, @ei\_parameter, @type\_parameter, @flag\_config);

END;

1. **Процедура добавления перечисления**

CREATE PROCEDURE [dbo].[Добавить\_перечисление]

*-- Функция: Добавление нового перечисления в базу данных*

*-- Вход:*

*-- @name\_enum NVARCHAR(MAX) - Название перечисления*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Перечисления*

AS

BEGIN

INSERT INTO Перечисления (Название\_перечисления)

VALUES (@name\_enum);

END;

1. **Процедура добавления предиката**

CREATE PROCEDURE [dbo].[Добавить\_предикат]

*-- Функция: Добавление нового предиката в таблицу*

*-- Вход:*

*-- @id\_pred INT - Идентификатор предиката*

*-- @operation\_pred NVARCHAR(MAX) - Оператор предиката (например, '=', '<', '>')*

*-- @id\_parametr INT - Идентификатор параметра*

*-- @id\_val INT - Идентификатор значения*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Предикаты*

AS

BEGIN

INSERT INTO Предикаты (ID\_предиката, Оператор, ID\_параметра, ID\_значения)

VALUES (@id\_pred, @operation\_pred, @id\_parametr, @id\_val);

END;

1. **Процедура добавления спецификации**

CREATE PROCEDURE [dbo].[Добавить\_спецификацию]

*-- Функция: Добавление новой спецификации в таблицу*

*-- Вход:*

*-- @id\_prod\_general INT - Идентификатор общей продукции*

*-- @position\_number INT - Номер позиции в спецификации*

*-- @id\_prod\_part INT - Идентификатор части продукции*

*-- @quantity INT - Количество частей продукции*

*-- @id\_func INT - Идентификатор функции*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Спецификация\_продукции*

AS

BEGIN

INSERT INTO Спецификация\_продукции (ID\_продукции\_общей, Номер\_позиции, ID\_продукции\_части, Количество, ID\_функции)

VALUES (@id\_prod\_general, @position\_number, @id\_prod\_part, @quantity, @id\_func);

END;

1. **Процедура добавления тела функции**

CREATE PROCEDURE [dbo].[Добавить\_тело\_функции]

*-- Функция: Добавление нового тела функции в таблицу*

*-- Вход:*

*-- @id\_func INT - Идентификатор функции*

*-- @num\_diz INT - Номер устройства (дизайна)*

*-- @num\_con INT - Номер связи*

*-- @id\_pred INT - Идентификатор предиката*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Тело\_функции*

AS

BEGIN

INSERT INTO Тело\_функции (ID\_функции, Номер\_устройства, Номер\_связи, ID\_предиката)

VALUES (@id\_func, @num\_diz, @num\_con, @id\_pred);

END;

1. **Процедура добавления функции**

CREATE PROCEDURE [dbo].[Добавить\_функцию]

*-- Функция: Добавление новой функции в таблицу*

*-- Вход:*

*-- @id\_func INT - Идентификатор функции*

*-- @name\_func NVARCHAR(MAX) - Название функции*

*-- Выход: Нет*

*-- Эффекты: Добавляет новую запись в таблицу Функции*

AS

BEGIN

INSERT INTO Функции (ID\_функции, Название\_функции)

VALUES (@id\_func, @name\_func);

END;

1. **Процедура поиска по конфигурации**

CREATE PROCEDURE [dbo].[Поиск\_параметров\_по\_конфигурации]

*-- Функция: Извлечение параметров продукции по заданной продукции и конфигурации*

*-- Вход:*

*-- @prod INT - Идентификатор продукции*

*-- @config INT - Номер конфигурации*

*-- Выход: Нет (возвращает набор данных)*

*-- Эффекты: Возвращает список параметров для указанной продукции и конфигурации*

AS

BEGIN

SELECT

p.Название\_продукции AS product\_name,

pr.Название\_параметра AS parameter\_name,

vp.Значение AS parameter\_value,

EI.Краткое\_название\_ЕИ AS ei

FROM Значения\_параметров vp

JOIN Продукция p ON vp.ID\_продукции = p.ID\_продукции

JOIN Параметры pr ON vp.ID\_параметра = pr.ID\_параметра

LEFT JOIN Единицы\_измерения EI ON pr.ЕИ\_параметра = EI.ID\_ЕИ

WHERE vp.ID\_продукции = @prod AND vp.Номер\_конфигурации = @config;

END;

1. **Процедура поиска конфигурации**

CREATE PROCEDURE [dbo].[Поиск\_спецификации]

*-- Функция: Извлечение полной спецификации для указанного изделия с учетом иерархии частей*

*-- Вход:*

*-- @id\_prod INT - Идентификатор продукции (изделия)*

*-- Выход: Нет (возвращает набор данных)*

*-- Эффекты: Возвращает иерархический список частей и количества для указанного изделия*

AS

BEGIN

*-- Определение рекурсивного CTE*

WITH tab1 (ID\_продукции\_общей, Номер\_позиции, ID\_продукции\_части, Количество, Название\_продукции, path) AS

(

*-- Базовый случай: выбираем записи для конкретного изделия*

SELECT s.ID\_продукции\_общей, s.Номер\_позиции, s.ID\_продукции\_части, s.Количество, p.Название\_продукции,

CAST(s.ID\_продукции\_части AS NVARCHAR(255)) AS path

FROM Спецификация\_продукции s

JOIN Продукция p ON s.ID\_продукции\_части = p.ID\_продукции

WHERE s.ID\_продукции\_общей = @id\_prod

UNION ALL

*-- Рекурсивный случай: выбираем дочерние элементы*

SELECT s2.ID\_продукции\_общей, s2.Номер\_позиции, s2.ID\_продукции\_части, s2.Количество, p.Название\_продукции,

CAST(tab1.path + '->' + CAST(s2.ID\_продукции\_части AS NVARCHAR(255)) AS NVARCHAR(255))

FROM Спецификация\_продукции s2

JOIN Продукция p ON s2.ID\_продукции\_части = p.ID\_продукции

JOIN tab1 ON tab1.ID\_продукции\_части = s2.ID\_продукции\_общей

)

*-- Вывод результата: все строки спецификации для изделия*

SELECT

(SELECT p.Название\_продукции FROM Продукция p WHERE tab3.ID\_продукции\_общей = p.ID\_продукции) AS name\_product,

tab3.Номер\_позиции,

p.Название\_продукции AS name\_product\_part,

tab3.Количество

FROM tab1 AS tab3

JOIN Продукция AS p ON tab3.ID\_продукции\_части = p.ID\_продукции

ORDER BY path, p.Название\_продукции;

END;

1. **Процедура проверки спецификации**

CREATE PROCEDURE [dbo].[Проверка\_спецификации]

*-- Функция: Проверка возможности добавления позиции в спецификацию на основе заданных правил*

*-- Вход:*

*-- @id\_prod\_general INT - Идентификатор общей продукции (изделия)*

*-- @position\_number INT - Номер позиции спецификации*

*-- @id\_prod\_part INT - Идентификатор части продукции*

*-- @quantity INT - Количество части продукции*

*-- @idFunc INT - Идентификатор функции для проверки предикатов*

*-- Выход: Сообщение о результате операции*

*-- Эффекты: Добавляет позицию в спецификацию при соблюдении правил; возвращает сообщение об успехе или неудаче*

AS

BEGIN

IF EXISTS (

SELECT 1

FROM Тело\_функции tf

LEFT JOIN Предикаты p ON tf.ID\_предиката = p.ID\_предиката

LEFT JOIN Значения\_перечислений EV ON EV.ID\_значения = p.ID\_значения

WHERE tf.ID\_функции = @idFunc AND EV.Значение = @quantity

)

BEGIN

*-- Если правило проходит проверку, добавляем позицию в спецификацию*

EXEC Добавить\_спецификацию @id\_prod\_general, @position\_number, @id\_prod\_part, @quantity, @idFunc;

RETURN 'Позиция спецификации успешно добавлена';

END

ELSE

BEGIN

*-- Если проверка не пройдена, выводим ошибку*

RETURN 'Позиция не может быть добавлена, так как количество входного продукта нарушает правила конфигурации';

END;

END;

1. **Процедура просмотра подграфа**

CREATE PROCEDURE [dbo].[Просмотр\_подграфа]

*-- Функция: Извлечение иерархического подграфа классов и связанных продуктов*

*-- Вход:*

*-- @id\_class INT - Идентификатор начального класса*

*-- Выход: Нет (возвращает набор данных)*

*-- Эффекты: Возвращает все дочерние классы для заданного класса, включая их наименования и связанные имена продуктов*

AS

BEGIN

*-- Определение рекурсивного CTE*

WITH tab1 (ID\_класса, ID\_родительского\_класса, Наименование, path) AS

(

*-- Базовый случай (выбираем начальный класс)*

SELECT t1.ID\_класса, t1.ID\_родительского\_класса, t1.Наименование, CAST(t1.Наименование AS NVARCHAR(255)) AS path

FROM Класс\_изделия t1

WHERE t1.ID\_класса = @id\_class

UNION ALL

*-- Рекурсивный случай (поиск всех дочерних классов)*

SELECT t2.ID\_класса, t2.ID\_родительского\_класса, t2.Наименование,

CAST(tab1.path + ' => ' + t2.Наименование AS NVARCHAR(255))

FROM Класс\_изделия t2

INNER JOIN tab1 ON tab1.ID\_класса = t2.ID\_родительского\_класса

)

*-- Результат запроса: выбираем все классы, их наименования и привязанные продукты*

SELECT c.ID\_класса, c.path, p.Название\_продукции AS name\_product

FROM tab1 AS c

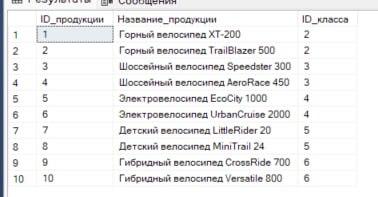
LEFT JOIN Продукция AS p ON c.ID\_класса = p.ID\_класса

ORDER BY c.path, p.Название\_продукции;

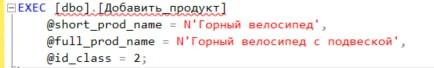
END;

## Тестирование разработки

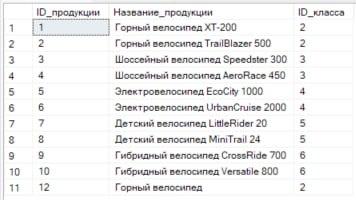
1. **Добавления продукта**



*Таблица 1 – Продукты*



*Процедура 1 – Добавление продукта*



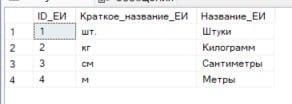
*Таблица 2 – Новый продукт*

1. **Добавление единицы измерения**

*Таблица 3 – Единицы измерения*

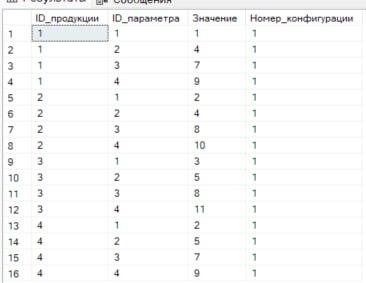


*Процедура 2 – Добавление единицы измерения*

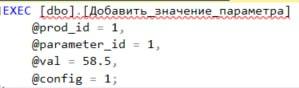


*Таблица 4 – Новая единица измерения*

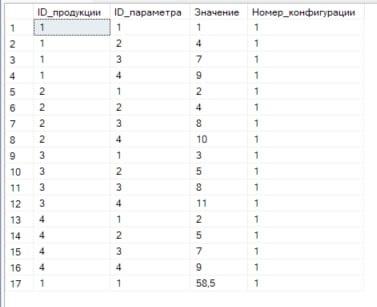
1. **Добавление значений и параметров**



*Таблица 5 – Значения и параметры*



*Процедура 3 – Добавление значения параметра*

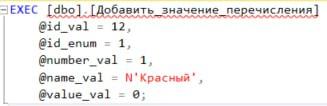


*Таблица 8 – Новые значения и параметры*

1. **Добавление значений перечислений**



*Таблица 9 – Значения перечисления*

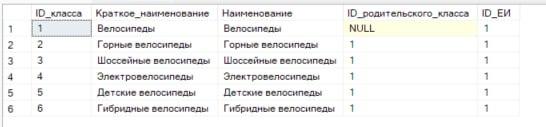


*Процедура 4 – Добавление значения перечисления*

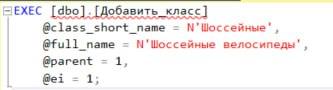


*Таблица 10 –Новое значение перечисления*

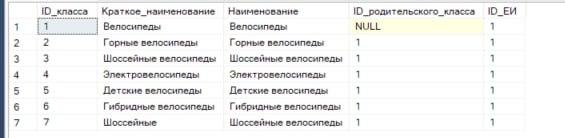
1. **Добавление классов изделий**



*Таблица 11 – Классы изделий*

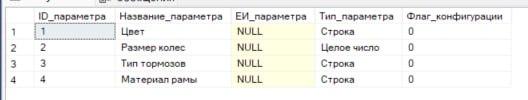


*Процедура 5 – Добавление класса продукта*

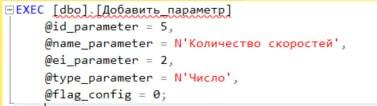


*Таблица 12 – Новый класс изделий*

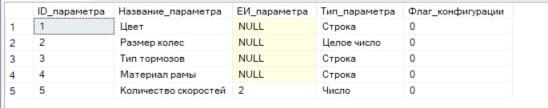
1. **Добавление параметров**



*Таблица 13 – Параметры*

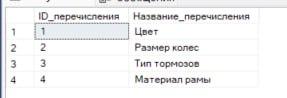


*Процедура 6 – Добавление параметра*



*Таблица 14 – Новый параметр*

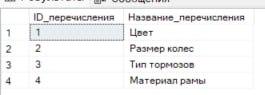
1. **Добавление перечислений**



*Таблица 15 – Перечисления*

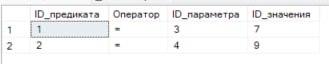


*Процедура 7 – Добавление перечисления*

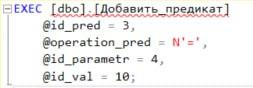


*Таблица 16 – Новое перечисление*

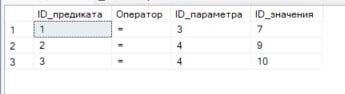
1. **Добавление предикатов**



*Таблица 17 –Предикаты*

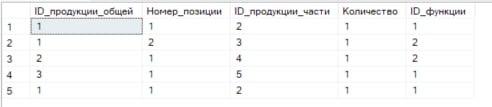


*Процедура 8 – Добавление предиката*

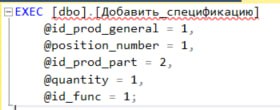


*Таблица 18 – Новый предикат*

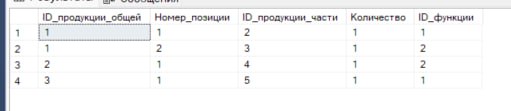
1. **Добавление спецификации**



*Таблица 19 – Спецификации*

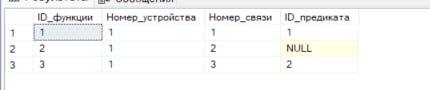


*Процедура 9 – Добавление спецификации*



*Таблица 20 – Новая спецификация*

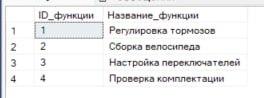
1. **Добавление фукнции**



*Таблица 21 – Функции*



*Процедура 10 – Добавление функции*

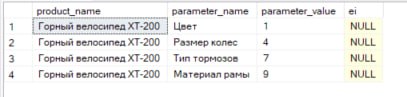


*Таблица 22 – Новая функция*

1. **Поиск параметров по конфигурации**



*Процедура 11 – Поиск параметров по конфигурации*

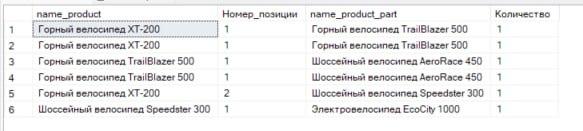


*Таблица 23 – Параметры по конфигурации*

1. **Поиск спецификации**

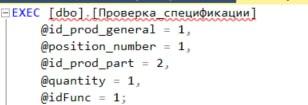


*Процедура 12 – Поиск спецификации*



*Таблица 24 – спецификации*

1. **Проверка спецификации**



*Процедура 13 – Проверка спецификации*

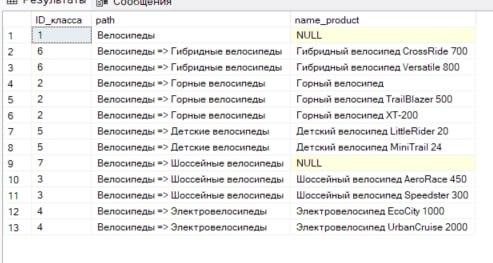


*Процедура 14 – Ошибка проверки спецификации*

1. **Просмотр подграфа**



*Процедура 15 – Просмотр подграфа*



*Таблица 25 – Подграф*

# Вывод

В данной работе был разработан проект каркаса для Спецификации изделий с вариантами исполнения.

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