**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**

**по дисциплине «Базы данных»**

**Тема: Проектирование базы данных**

|  |  |  |
| --- | --- | --- |
| Студент |  |  |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2023

**ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**

|  |  |  |
| --- | --- | --- |
| Студент | | |
| Группа | | |
| Тема работы: Проектирование базы данных | | |
| Исходные данные:  Разработать БД для сотрудников склада. | | |
| Содержание пояснительной записки: «Содержание», «Введение», «Задание», «Предметная область», «Проектирование базы данных», «Содержание базы данных», «Создание таблиц», «Заполнение таблиц данными», «Разработка объектов промежуточного слоя», «Разработка стратегии резервного копирования», «Заключение», «Список использованных источников». | | |
| Предполагаемый объем пояснительной записки:  Не менее страниц. | | |
| Дата выдачи задания | | |
| Дата сдачи задания: | | |
| Дата защиты задания | | |
| Студент |  |  |
| Преподаватель |  | Новакова Н.Е. |

**АННОТАЦИЯ**

В данной работе реализована база данных сотрудника склада. Реализованы технологии добавления новых данных, а именно – товаров, контрактов и стеллажей, UDF - функции для проверки правильного добавления данных, представления для вывода всех действительных контрактов на сегодняшний день, заполненности помещения и стеллажей, а также создана возможность резервного копирования базы данных.

**SUMMARY**

In this work, the warehouse employee database is implemented. Technologies for adding new data have been implemented, namely, goods, contracts and racks, UDF functions for checking the correct addition of data, representations for displaying all valid contracts to date, occupancy of the room and racks, and the ability to backup the database has been created.

**СОДЕРЖАНИЕ**

[Введение 5](#_30j0zll)

[Цель работы 6](#_5ua4lycenx7q)

[1. Задание 7](#_1fob9te)

[2. Предметная область 7](#_3znysh7)

[3. Проектирование базы данных 8](#_2et92p0)

[4. Создание базы данных 13](#_tyjcwt)

[5. Создание таблиц 14](#_3dy6vkm)

[6. Заполнение таблиц данными 22](#_1t3h5sf)

[7. Разработка объектов промежуточного слоя 26](#_4d34og8)

[Хранимые процедуры 26](#_2s8eyo1)

[UDF 32](#_17dp8vu)

[Представления 37](#_26in1rg)

[8. Разработка стратегии резервного копирования 40](#_lnxbz9)

[9. Заключение 42](#_1ksv4uv)

[10. Список использованных источников 45](#_44sinio)

# Введение

В ходе выполнения данного индивидуального домашнего задания необходимо спроектировать, создать базу данных в рамках указанной предметной области, разработать объекты промежуточного слоя для неё, а также продемонстрировать полученные результаты.

Выполнение задания позволит лучше понять процесс проектирования баз данных, а также закрепить на практике навыки, полученные в ходе освоения теоретической и практической частей дисциплины «Базы данных».

# Цель работы

Целью данного индивидуального домашнего задания является проектирование и создание базы данных для сотрудника склада. Одной из важных частей проекта является разработка объектов промежуточного слоя (хранимые процедуры, UDF, представления). Также в рамках работы предлагается стратегия резервного копирования базы данных.

# Задание

Разработать БД для сотрудников склада.

1. Разработка включает в себя следующие этапы:
2. Проектирование БД (структура данных)
3. Создание БД
4. Создание таблиц и ограничений целостности
5. Заполнение таблиц данным
6. Разработка объектов промежуточного слоя (представлений, хранимых процедур, UDF-ов)
7. Разработка стратегии резервного копирования

# Предметная область

БД *«Склад»* проектируем при условии, что:

*–* на складе может быть несколько помещений;

*–* в каждом помещении может быть несколько стеллажей с хранимым товаром;

*–* товар на стеллажах может принадлежать разным компаниям и занимать разное место;

*–* у каждой компании хранящей свой товар на данном складе подписан контракт с данным складом;

*–* вместимость помещения принимается, как кол-во стеллажей которые можно разместить в помещении;

*–* вместимость стеллажа принимается, как кол-во товара, хранимого на стеллаже;

# Проектирование базы данных

Проектируем БД с использованием метода ER-диаграмм [1 – 3]. Выделим объекты и их атрибуты:

– Companies (CompanyID\*, Name, Address, PhoneNumber, TIN).

– Contracts (ContractID\*, CompanyID, StartDate, EndDate, Number).

– Rooms (RoomID\*, Capacity, Number).

– Shelves (ShelfID\*, RoomID, Capacity, Number).

Рассмотрим связь между объектами Companies и Contracts представленную на рисунке 1.

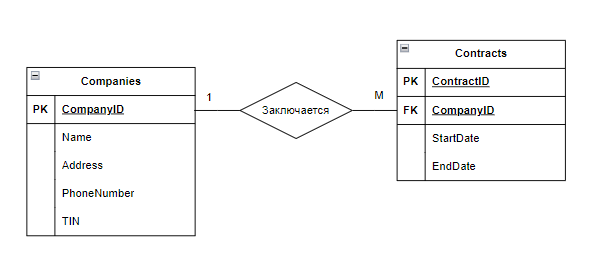


Рисунок 1

У одной компании может быть несколько контрактов, но у каждого контракта одна компания с которой он заключен, следовательно, в соответствии с правилом 4 [1 – 3] «Если степень взаимосвязи 1:M и классы принадлежности обеих сущностей обязательны, то ER-диаграмма преобразуется в общем случае в два, а в частности в одно отношение (таблицу). Схема первого отношения содержит атрибуты первой сущности, схема второго – атрибуты второй сущности и ключ первой сущности. Если схема первого отношения состоит только из ключевого атрибута, то отношение можно исключить как избыточное», получим:

– Companies (CompanyID\*, Name, Address, PhoneNumber, TIN).

– Contracts (ContractID\*, CompanyID, StartDate, EndDate, Number).

Рассмотрим связь между объектами Room и Shelves представленную на рисунке 2.

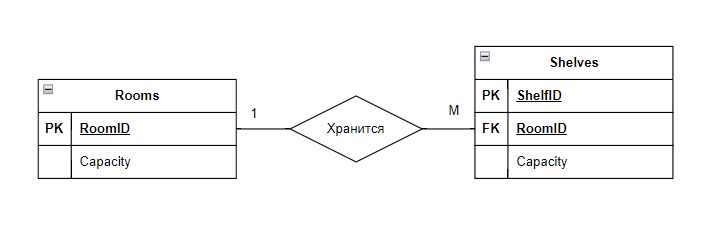


Рисунок 2

В одном помещении может находится несколько стеллажей, следовательно, в соответствии с правилом 4 [1 – 3] «Если степень взаимосвязи 1:M и классы принадлежности обеих сущностей обязательны, то ER-диаграмма преобразуется в общем случае в два, а в частности в одно отношение (таблицу). Схема первого отношения содержит атрибуты первой сущности, схема второго – атрибуты второй сущности и ключ первой сущности. Если схема первого отношения состоит только из ключевого атрибута, то отношение можно исключить как избыточное», получим:

– Rooms (RoomID\*, Capacity, Number).

– Shelves (ShelfID\*, RoomID, Capacity, Number).

Рассмотрим связь между объектами Contracts и Shelves представленную на рисунке 3.

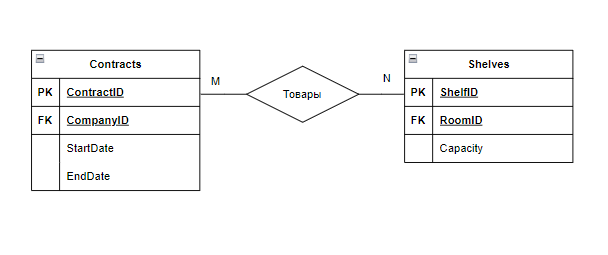


Рисунок 3

По нескольким контрактам на нескольких полках могут хранится разные товары, следовательно, в соответствии с правилом 6 [1 – 3] «Если степень взаимосвязи двух сущностей равна *M*:*N*, то независимо от классов принадлежностей сущностей ER-диаграмма преобразуется в три отношения (три таблицы). Схемы первого и второго отношений содержат атрибуты соответствующих сущностей, а в схему третьего отношения включаются ключи обеих сущностей», получим:

– Companies (CompanyID\*, Name, Address, PhoneNumber, TIN).

– Contracts (ContractID\*, CompanyID, StartDate, EndDate, Number).

– Rooms (RoomID\*, Capacity, Number).

– Shelves (ShelfID\*, RoomID, Capacity, Number).

– Products (ProductID\*, ShelfID, ContractID, Name, Type, Quantity);

В базе данных представлены следующие таблицы:

1. Companies – таблица, хранящая информацию о компаниях с которыми заключен контракт на хранение, а именно: Идентификатор компании, название компании, адрес компании, контактный телефон и ИНН. Первичный ключ — CompanyID.
2. Contracts – таблица, хранящая информацию о номерах контрактах, заключенных со складом, а именно: Идентификатор контракта идентификатор компании, дата подписания контракта, дата окончания контракта. Первичный ключ — ContractID. Вторичный ключ — CompanyID.
3. Rooms – таблица, хранящая информацию о помещениях склада, а именно: Идентификатор помещения и вместимость помещения. Первичный ключ — RoomID.
4. Shelves – таблица, хранящая информацию о стеллажах на которых хранится товар, а именно: Идентификатор стеллажа, идентификатор помещения и вместимость стеллажа. Первичный ключ — ShelfID. Вторичный ключ — RoomID.
5. Products – таблица, хранящая информацию о товарах, хранящихся на складе, а именно: Идентификатор товара, идентификатор помещения, идентификатор контракта, название, тип, количество товара. Первичный ключ — ProductID. Вторичные ключи — ShelfID, RoomID.

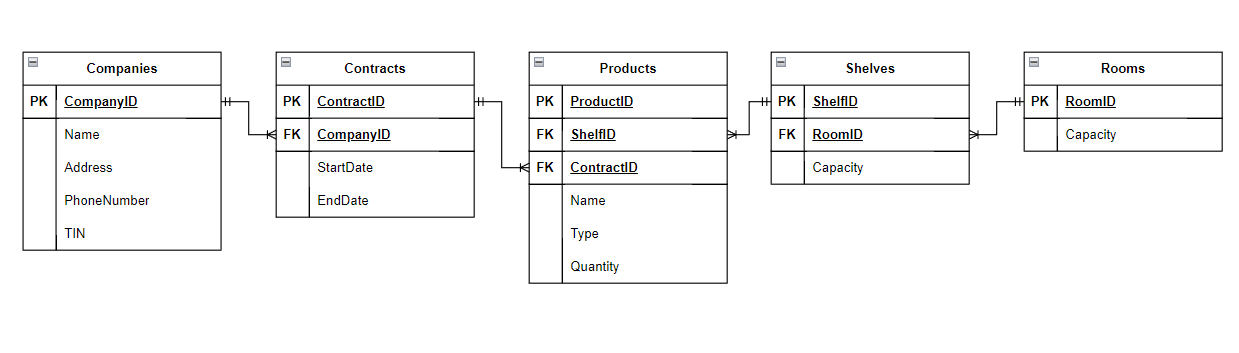


Рисунок 4. Диаграмма сущность—связь.

# Создание базы данных

CREATE DATABASE Warehouse ON

(

name = Warehouse\_dat,

filename = 'D:\Warehouse.mdf',

size = 5MB,

maxsize = 50MB,

filegrowth = 5MB

)

log ON

(

name = Warehouse\_log,

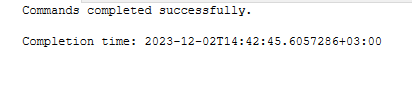
filename = 'D:\Warehouse.ldf',

size = 5MB,

maxsize = 50MB,

filegrowth = 5MB

)



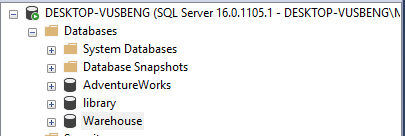


Рисунок 5. Результат создания базы данных.

USE Warehouse

GO

CREATE SCHEMA Wrhs

# Создание таблиц

Таблица 1. Описание таблицы Companies.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица компаний** | | | | Имя таблицы:  Companies | | | |
| Дата разработки:  .12.2023 | | |
| Порядковый номер таблицы:  **1** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор компании | | CompanyID | INT | | P | NOT NULL | |  |
|  | 2 | Название компании | | Name | NVARCHAR(30) | |  | NOT NULL | |  |
|  | 4 | Адрес компании | | Address | NVARCHAR(100) | |  | NOT NULL | |  |
|  | 5 | Контактный телефон | | PhoneNumber | NVARCHAR(20) | |  | NOT NULL | |  |
|  | 6 | ИНН | | TIN | NVARCHAR(12) | |  | NOT NULL | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 2. Описание таблицы Contracts.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица контрактов** | | | | Имя таблицы:  Contracts | | | |
| Дата разработки:  .12.2023 | | |
| Порядковый номер таблицы:  **2** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор контракта | | ContractID | INT | | P | NOT NULL | |  |
|  | 2 | Идентификатор компании | | CompanyID | INT | | F | NOT NULL | |  |
|  | 3 | Дата подписания контракта | | StartDate | DATE | |  | NOT NULL | |  |
|  | 4 | Дата окончания контракта | | EndDate | DATE | |  | NOT NULL | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 3. Описание таблицы Rooms.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица помещений** | | | | Имя таблицы:  Rooms | | | |
| Дата разработки:  .12.2023 | | |
| Порядковый номер таблицы:  **4** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор помещения | | RoomID | INT | | P | NOT NULL | |  |
|  | 2 | Вместимость помещения | | Capacity | INT | |  | CHECK ( Capacity >= 0), NOT NULL | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 4. Описание таблицы Shelves.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица стеллажей** | | | | Имя таблицы:  Shelves | | | |
| Дата разработки:  .12.2023 | | |
| Порядковый номер таблицы:  **4** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор стеллажа | | ShelfID | INT | | P | NOT NULL | |  |
|  | 2 | Идентификатор помещения | | RoomID | INT | | F | NOT NULL | |  |
|  | 3 | Вместимость стеллажа | | Capacity | INT | |  | CHECK ( Capacity >= 0), NOT NULL | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 5. Описание таблицы Products.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица товаров** | | | | Имя таблицы:  Products | | | |
| Дата разработки:  .12.2023 | | |
| Порядковый номер таблицы:  **3** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Идентификатор товара | | ProductID | INT | | P | NOT NULL | |  |
|  | 2 | Идентификатор  помещения | | ShelfID | INT | | F | NOT NULL | |  |
|  | 3 | Идентификатор контракта | | ContractID | INT | | F | NOT NULL | |  |
|  | 4 | Название | | Name | NVARCHAR(20) | |  | NOT NULL | |  |
|  | 5 | Тип | | Type | NVARCHAR(20) | |  | NOT NULL | |  |
|  | 6 | Количество товара | | Quantity | INT | |  | CHECK ( Quantity >= 0), NOT NULL | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

В качестве ограничения целостности в основном используется NOT NULL, однако предусмотрена проверка на неправильный ввод полей с типом данных INT.

CREATE TABLE Wrhs.Companies (

CompanyID INT IDENTITY(1, 1) PRIMARY KEY NOT NULL,

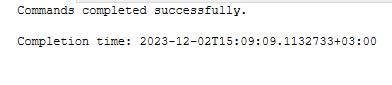
Name NVARCHAR(30) NOT NULL,

Address NVARCHAR(100),

PhoneNumber NVARCHAR(20) NOT NULL,

TIN NVARCHAR(12) NOT NULL

);



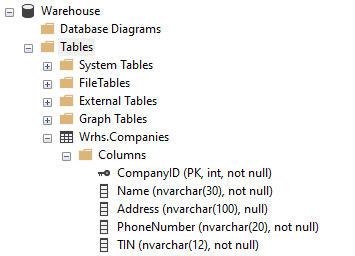


Рисунок 6. Результат создания таблицы Companies.

CREATE TABLE Wrhs.Contracts (

ContractID INT IDENTITY(1, 1) PRIMARY KEY NOT NULL,

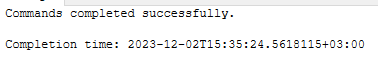
CompanyID INT NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

FOREIGN KEY (CompanyID) REFERENCES Wrhs.Companies(CompanyID)

);



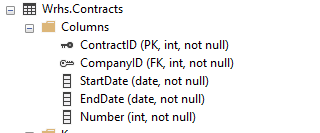


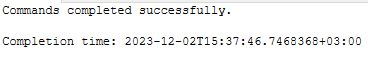
Рисунок 7. Результат создания таблицы Contracts.

CREATE TABLE Rooms (

RoomID INT IDENTITY(1, 1) PRIMARY KEY NOT NULL,

Capacity INT CHECK (Capacity >= 0) NOT NULL,

);



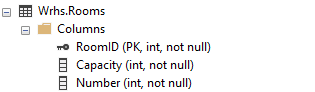


Рисунок 8. Результат создания таблицы Rooms.

CREATE TABLE Shelves (

ShelfID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

RoomID INT NOT NULL,

Capacity INT CHECK (Capacity >= 0) NOT NULL,

FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)

);



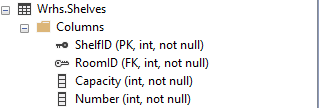


Рисунок 9. Результат создания таблицы Shelves.

CREATE TABLE Wrhs.Products (

ProductID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

ShelfID INT NOT NULL,

ContractID INT NOT NULL,

Name NVARCHAR(20) NOT NULL,

Type NVARCHAR(20) NOT NULL,

Quantity INT CHECK (Quantity >= 0) NOT NULL,

FOREIGN KEY (ShelfID) REFERENCES Wrhs.Shelves(ShelfID),

FOREIGN KEY (ContractID) REFERENCES Wrhs.Contracts(ContractID)

);



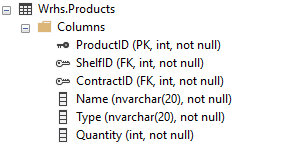


Рисунок 8. Результат создания таблицы Rooms.

# Заполнение таблиц данными

INSERT INTO Wrhs.Companies (Name, Address, PhoneNumber, TIN)

VALUES

('Альфа Технолоджис', 'Улица Пушкина, д. 10', '111-111-111', '123456789012'),

('Бета Инновейшн', 'Проезд Гагарина, д. 5', '222-222-222', '234567890123'),

('Гамма Сервисы', 'Проспект Ленина, д. 15', '333-333-333', '345678901234'),

('Дельта Продукты', 'Микрорайон Солнечный, д. 8', '444-444-444', '456789012345'),

('Эпсилон Консалтинг', 'Переулок Петровского, д. 12', '555-555-555', '567890123456'),

('Зета Системс', 'Бульвар Кирова, д. 7', '666-666-666', '678901234567'),

('Иота Строительство', 'Площадь Революции, д. 20', '777-777-777', '789012345678'),

('Каппа Фарма', 'Проспект Гагарина, д. 3', '888-888-888', '890123456789'),

('Лямбда Логистика', 'Проезд Ленина, д. 6', '999-999-999', '901234567890'),

('Мю Электроникс', 'Улица Пушкина, д. 25', '000-000-000', '012345678901');

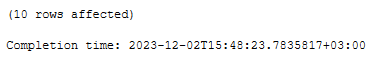
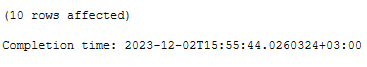




Рисунок 9. Результат заполнения таблицы Companies.

INSERT INTO Wrhs.Contracts (CompanyID, StartDate, EndDate)

VALUES (1, '2022-08-01', '2022-08-31'), (2, '2022-07-15', '2022-10-15'), (3, '2022-09-01', '2022-12-31'), (4, '2022-06-01', '2022-12-01'), (5, '2022-10-01', '2022-11-30'), (6, '2022-08-15', '2022-12-15'), (7, '2022-07-01', '2022-09-30'), (8, '2022-09-15', '2022-11-15'), (9, '2022-06-15', '2022-10-15'), (10, '2022-07-15', '2022-08-31');



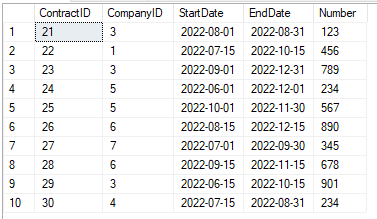


Рисунок 10. Результат заполнения таблицы Contracts.

INSERT INTO Wrhs.Rooms (Capacity)

VALUES (50),(40),(30),(20),(50),(40),(30),(20),(50),(40);



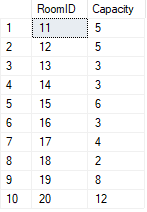
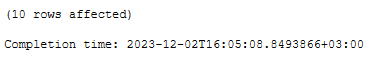


Рисунок 11. Результат заполнения таблицы Rooms.

INSERT INTO Shelves (RoomID, Capacity)

VALUES (1, 50), (2, 40), (1, 30), (3, 20), (2, 50), (3, 40), (1, 30), (2, 20), (3, 50), (1, 40);



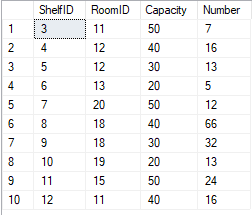


Рисунок 12. Результат заполнения таблицы Shelves.

INSERT INTO Wrhs.Products (ProductID, ShelfID, ContractID, Name, Type, Quantity)

VALUES (1, 3, 21, 'Чашка', 'Чашка', 20),

(2, 3, 21, 'Кружка', 'Чашка', 20),

(3, 4, 25, 'Спиннинг', 'Удочка или спиннинг', 5),

(4, 4, 25, 'Удочка', 'Удочка или спиннинг', 10),

(5, 11, 30, 'Кукла', 'Игрушка', 30),

(6, 11, 30, 'Машинка', 'Игрушка', 20),

(7, 8, 23, 'Набор ложек', 'Столовые', 10),

(8, 6, 27, 'Сковородка', 'Сковородка', 5),

(9, 6, 27, 'Сковородка 20см', 'Сковородка', 5),

(10, 12, 30, 'Спиннер', 'Игрушка', 20);



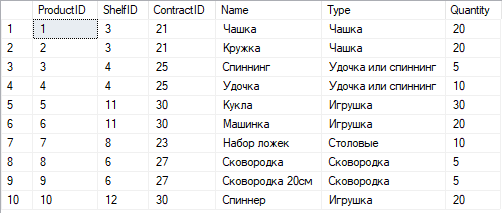


Рисунок 13. Результат заполнения таблицы Products.

Таким образом было произведено заполнение таблиц базы данных.

# Разработка объектов промежуточного слоя

## Хранимые процедуры

1. Добавление новых товаров.

CREATE PROCEDURE Wrhs.AddProductToShelf

@ShelfID INT,

@ContractID INT,

@Name NVARCHAR(20),

@Type NVARCHAR(20),

@Quantity INT

AS

BEGIN

DECLARE @IsAvailable BIT;

SET @IsAvailable = Wrhs.CheckShelf(@ShelfID, @Quantity)

IF (@IsAvailable = 1)

BEGIN

INSERT INTO Wrhs.Products (ShelfID, ContractID, Name, Type, Quantity)

VALUES (@ShelfID, @ContractID, @Name, @Type, @Quantity);

SELECT 'Товар успешно добавлен на полку.' AS [Результат];

END

ELSE

BEGIN

SELECT 'Недостаточно места на полке для добавления товара.' AS [Результат];

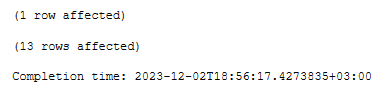
END

END

Проверим корректность добавления нового товара:

EXEC Wrhs.AddProductToShelf @ShelfID = 3, @ContractID = 21, @Name = 'Чашкааа', @Type = 'Чашка', @Quantity = 10

SELECT \* FROM Wrhs.Products



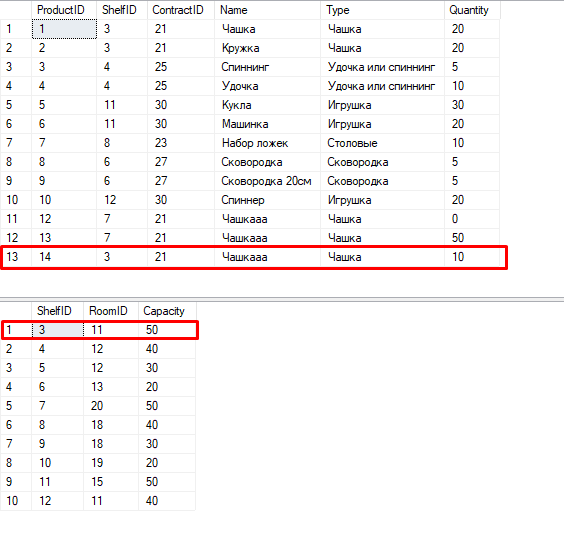


Рисунок 14. Результат вызова хранимой процедуры AddProductToShelf.

1. Добавление новых контрактов.

CREATE PROCEDURE Wrhs.AddContract

(

@CompanyID INT,

@StartDateStr VARCHAR(10),

@EndDateStr VARCHAR(10)

)

AS

BEGIN

DECLARE @StartDate DATE;

DECLARE @EndDate DATE;

IF (Wrhs.ValidateDates(@StartDateStr, @EndDateStr) = 1)

BEGIN

SET @StartDate = TRY\_CAST(@StartDateStr AS DATE);

SET @EndDate = TRY\_CAST(@EndDateStr AS DATE);

INSERT INTO Wrhs.Contracts (CompanyID, StartDate, EndDate)

VALUES (@CompanyID, @StartDate, @EndDate);

SELECT 'Запись успешно добавлена.' AS [Результат];

END

ELSE

BEGIN

SELECT 'Невалидные даты. Запись не добавлена.' AS [Результат];

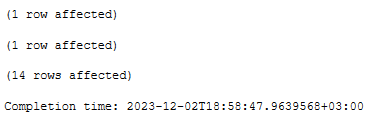
END

END;

Проверим корректность добавления нового контракта:

EXEC Wrhs.AddContract @CompanyID = 1, @StartDateStr = '2023-11-10', @EndDateStr = '2024-01-19'

SELECT \* FROM Wrhs.Contracts



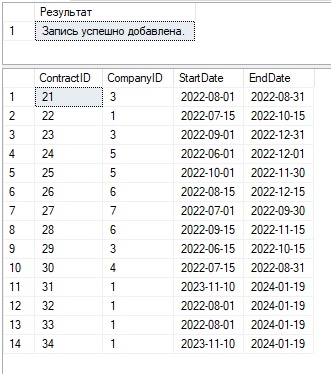


Рисунок 15. Результат вызова хранимой процедуры AddContract.

1. Добавление новых стеллажей.

CREATE PROCEDURE Wrhs.AddShelfToRoom(

@RoomID INT,

@Capacity INT

)

AS

BEGIN

IF (Wrhs.CheckRoom(@RoomID) = 1)

BEGIN

INSERT INTO Shelves (RoomID, Capacity)

VALUES (@RoomID, @Capacity)

SELECT 'Запись успешно добавлена.' AS [Результат];

END

ELSE

BEGIN

SELECT 'Ошибка! Запись не добавлена. Помещение может быть переполнено' AS [Результат];

END

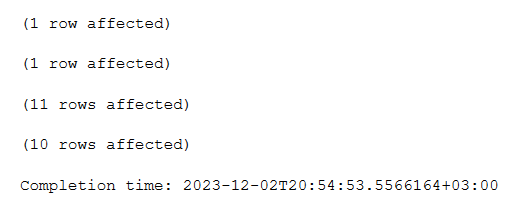
END

Проверим корректность добавления нового стеллажа:

EXEC Wrhs.AddShelfToRoom @RoomID = 11, @Capacity = 50

SELECT \* FROM Wrhs.Shelves

SELECT \* FROM Wrhs.Rooms



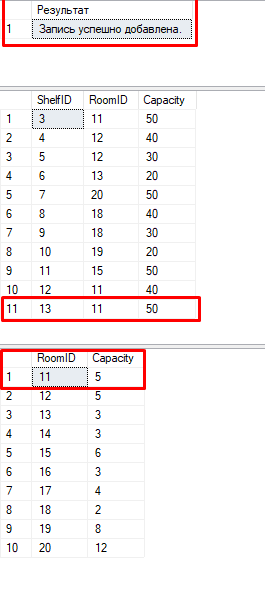
****

Рисунок 16. Результат вызова хранимой процедуры AddShelfToRoom.

## UDF

1. Проверка вместимости полки при добавлении новых товаров.

CREATE FUNCTION Wrhs.CheckShelf

(@ShelfID INT, @ProductSize INT)

RETURNS BIT

AS

BEGIN

DECLARE @IsAvailable BIT

DECLARE @CurrentCapacity INT

DECLARE @MaxCapacity INT

SET @CurrentCapacity = (SELECT SUM(Quantity) FROM Wrhs.Products WHERE ShelfID = @ShelfID)

SET @MaxCapacity = (SELECT Capacity FROM Wrhs.Shelves WHERE ShelfID = @ShelfID)

IF (@CurrentCapacity IS NULL)

SET @CurrentCapacity = 0

DECLARE @AvailableSpace INT

SET @AvailableSpace = @MaxCapacity - @CurrentCapacity

IF (@AvailableSpace >= @ProductSize)

SET @IsAvailable = 1

ELSE

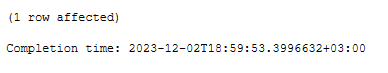
SET @IsAvailable = 0

RETURN @IsAvailable

END

Проверка:

SELECT Wrhs.CheckShelf(7,50)



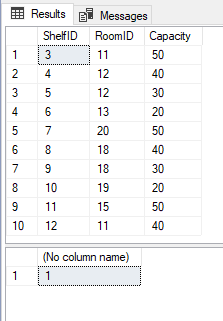


Рисунок 17. Результат вызова CheckShelf.

2. Проверка дат при добавлении нового контракта.

CREATE FUNCTION Wrhs.ValidateDates

(

@StartDateStr VARCHAR(10),

@EndDateStr VARCHAR(10)

)

RETURNS BIT

AS

BEGIN

DECLARE @IsValid BIT = 1;

DECLARE @StartDate DATE;

DECLARE @EndDate DATE;

SET @StartDate = TRY\_CAST(@StartDateStr AS DATE);

SET @EndDate = TRY\_CAST(@EndDateStr AS DATE);

IF (@StartDate IS NULL)

SET @IsValid = 0;

IF (@EndDate IS NULL)

SET @IsValid = 0;

ELSE IF (@EndDate < @StartDate)

SET @IsValid = 0;

RETURN @IsValid;

END;

Проверка:

SELECT Wrhs.ValidateDates('аыфафыафы','афафаф')

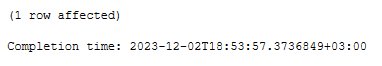




Рисунок 18. Результат вызова CheckShelf.

3. Проверка заполненности помещения стеллажами.

CREATE FUNCTION Wrhs.CheckRoom

(@RoomID INT)

RETURNS BIT

AS

BEGIN

DECLARE @IsAvailable BIT

DECLARE @CurrentCapacity INT

DECLARE @MaxCapacity INT

SET @CurrentCapacity = ISNULL((SELECT COUNT(\*) FROM Wrhs.Shelves WHERE RoomID = @RoomID), 0)

SET @MaxCapacity = (SELECT Capacity FROM Wrhs.Rooms WHERE RoomID = @RoomID)

IF (@CurrentCapacity IS NULL)

SET @CurrentCapacity = 0

DECLARE @AvailableSpace INT

SET @AvailableSpace = @MaxCapacity - @CurrentCapacity

IF (@AvailableSpace >= 1)

SET @IsAvailable = 1

ELSE

SET @IsAvailable = 0

RETURN @IsAvailable

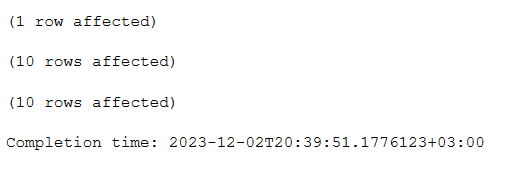
END

Проверка:

SELECT Wrhs.CheckRoom(18)

SELECT \* FROM Wrhs.Shelves

SELECT \* FROM Wrhs.Rooms



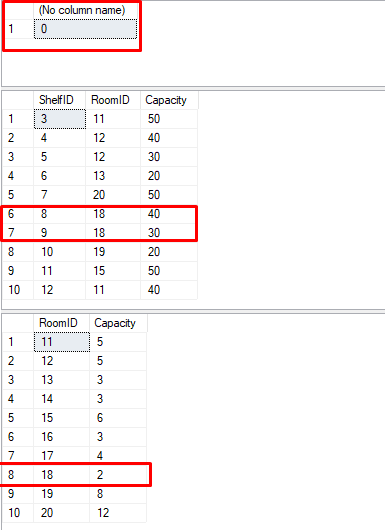


Рисунок 19. Результат вызова CheckRoom.

## 

## Представления

1. Заполненность всех помещений.

CREATE VIEW OccupancyView AS

SELECT

R.RoomID,

R.Capacity AS RoomCapacity,

COUNT(S.ShelfID) AS OccupiedShelves,

R.Capacity - COUNT(S.ShelfID) AS FreeShelves

FROM

Wrhs.Rooms R

LEFT JOIN

Wrhs.Shelves S ON R.RoomID = S.RoomID

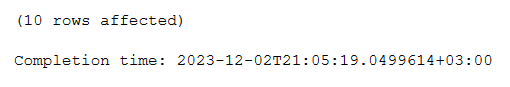
GROUP BY

R.RoomID,

R.Capacity;

Проверка:

SELECT \* FROM OccupancyView;



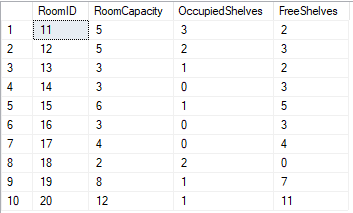


Рисунок 20. Результат вызова OccupancyView.

2. Заполненность стеллажей:

CREATE VIEW ShelfOccupancy AS

SELECT

S.ShelfID,

S.Capacity,

(S.Capacity - COALESCE(SUM(P.Quantity), 0)) AS FreeSpace

FROM

Wrhs.Shelves S

LEFT JOIN

Wrhs.Products P ON S.ShelfID = P.ShelfID

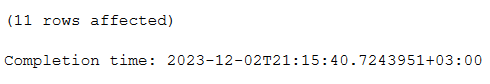
GROUP BY

S.ShelfID,

S.Capacity;

Проверка:

SELECT \* FROM ShelfOccupancy;



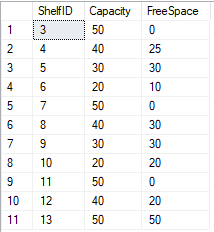


Рисунок 21. Результат вызова ShelfOccupancy.

3. Список всех контрактов, действительных на сегодняшний день.

CREATE VIEW Wrhs.ContractsToday AS

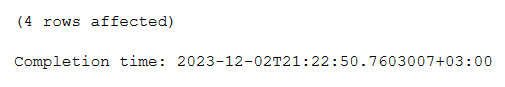
SELECT ContractID, CompanyID, StartDate, EndDate

FROM Wrhs.Contracts

WHERE GETDATE() BETWEEN StartDate AND EndDate;

Проверка:

SELECT \* FROM Wrhs.ContractsToday;



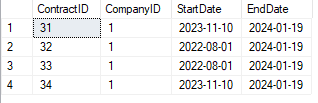


Рисунок 22. Результат вызова ContractsToday.

# Разработка стратегии резервного копирования

Для базы данных с небольшим количеством записей, можно предложить выполнять полное резервное копирование базы данных ежедневно в определённое время.

BACKUP DATABASE Warehouse

TO DISK = 'D:\IDZ\_BD\Warehouse.bak'

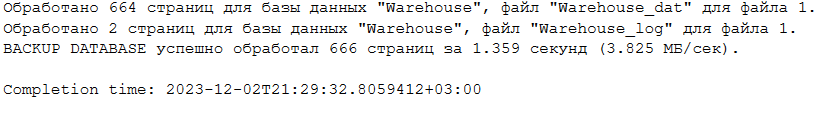


Рисунок 23.

В случае большого объема данных, можно выполнять полное резервное копирование еженедельно, а ежедневное полное резервное копирование заменить разностным резервным копированием.

USE Warehouse

BACKUP DATABASE Warehouse

TO DISK = 'D:\Warehouse\_diff.bak'

WITH DIFFERENTIAL

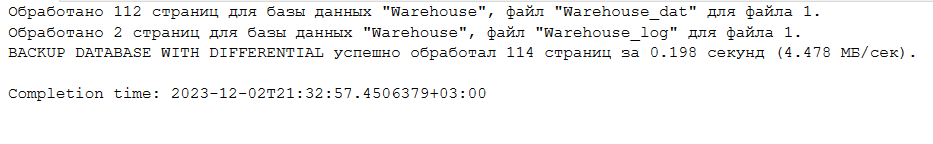


Рисунок 24.

Также необходимо выполнить резервное копирование журнала транзакций.

BACKUP LOG Warehouse

TO DISK = 'D:\Warehouse\_log.bak'

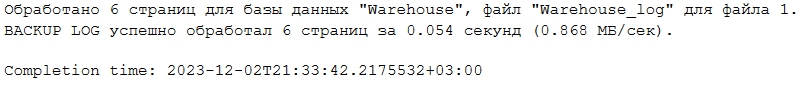


Рисунок 57.

Для восстановления резервной копии базы данных и одного или нескольких журналов транзакций или в других случаях, когда необходимо применить несколько инструкций RESTORE, необходимо указать параметр WITH NORECOVERY для всех инструкций, кроме последней.

RESTORE DATABASE Warehouse

FROM DISK = 'D:\IDZ\_BD\Warehouse.bak'

WITH NORECOVERY

RESTORE DATABASE Warehouse

FROM DISK = 'D:\Warehouse\_diff.bak'

WITH NORECOVERY

RESTORE LOG Warehouse

FROM DISK = 'D:\Warehouse\_log.bak'

WITH RECOVERY

# Заключение

В ходе выполнения данного индивидуального задания была спроектирована база данных для сотрудников склада, состоящая из пяти таблиц.

В ходе работы было разработано несколько объектов промежуточного слоя:

1. Хранимые процедуры для товаров, контрактов и стеллажей.
2. Функции для проверки вместимости помещений и стеллажей, а также для проверки правильности ввода даты при добавлении новых контрактов.
3. Представления для подсчета и отображения заполненности помещений и стеллажей, а также список всех контрактов, которые актуальны на сегодняшний день.

Также была предложена стратегия резервного копирования базы данных.

База данных является важным инструментом для менеджера склада, поскольку она позволяет эффективно управлять информацией о товарах, контрактах, клиентах и других аспектах складской деятельности. Вот несколько способов использования базы данных в контексте менеджера склада:

1. Управление товарами: База данных с информацией о количестве и расположении товаров на складе помогает менеджеру контролировать товары и предотвращать нехватку или избыток товаров. Он может использовать базу данных для проверки доступного запаса, отслеживания поступления и отгрузки товаров, а также для определения оптимального уровня запасов.

2. Управление контрактами: База данных может быть использована для отслеживания контрактов с компаниями, упрощая связь с компанией, хранящей свой товар на данном складе. Менеджер склада может использовать базу данных для обработки и отслеживания заказов, управления сроками хранения и оповещения компаний о статусе их контрактов.

3. Анализ и отчетность: База данных может быть использована для генерации различных отчетов и аналитической информации. Менеджер склада может анализировать данные о поставках и других параметрах, чтобы принимать более информированные решения по управлению складской деятельностью. Например, он может определить популярность определенных товаров, оценить эффективность тех или иных компаний или выявить потенциальные проблемы в процессе управления товарами.

Перспективы развития и модернизации базы данных для менеджера склада огромны. Некоторые из возможных улучшений включают следующее:

- Расширение функциональности: Базу данных можно расширить, чтобы включить дополнительные модули или таблицы для учета специфических потребностей складской деятельности, таких как учет серийных номеров или управление возвратами.

- Интеграция с другими системами: Базу данных можно интегрировать с другими системами, такими как система учета, чтобы обеспечить более единый и автоматизированный подход к управлению складской деятельностью.

- Использование продвинутых алгоритмов: Базу данных можно модернизировать, чтобы включить алгоритмы оптимизации, машинное обучение или искусственный интеллект, которые помогут прогнозировать спрос, оптимизировать размещение товаров или автоматически распределять задачи на складе.

База данных для менеджера склада имеет большой потенциал для совершенствования и развития. С соответствующими улучшениями и модернизацией, она может значительно улучшить эффективность и точность работы на складе, а также обеспечить более высокий уровень обслуживания клиентов.

Выполнение задания позволило отточить навыки работы с базами данных, закрепить навыки, полученные на лекционной и практической частях курса «Базы данных», так как все применяемые инструменты и знания уже были изучены с преподавателями.

# Список использованных источников

1. Распределенные базы данных: Методические указания к лабораторным работам / Сост.: А. В. Горячев, Н. Е. Новакова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 32 с
2. Курс «Базы данных» / Moodle — Open-source learning platform [Электронный ресурс] Режим доступа: https://vec.etu.ru/moodle/course/view.php?id=14314 (дата обращения 02.12.2023).
3. Руководство по SQL. Транзакции. / Proselyte [Электронный ресурс] Режим доступа: https://proselyte.net/tutorials/sql/sql-transactions/ (дата обращения 02.12.2023)
4. Справочник по Transact-SQL / Microsoft Learn [Электронный ресурс] Режим доступа: https://learn.microsoft.com/ru-ru/sql/t-sql/ (дата обращения 02.12.2023).
5. SQL Transactions Tutorial / Database.Guide [Электронный ресурс] Режим доступа: https://database.guide/sql-transactions-tutorial/ (дата обращения 02.12.2023).