

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

КУРСОВАЯ РАБОТА
по дисциплине «Базы данных»
ТЕМА: «СКЛАД ПРОДУКТОВ»

Студент гр. 1335

Максимов Ю.Е.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2024

ЗАДАНИЕ на курсовую работу

Студент Максимов Ю. Е.

Группа 1335

Тема работы: проектирование и разработка базы данных склада продуктов

Исходные данные:

Проектирование и разработка базы данных аптеки. Создание таблиц и ограничений целостности. Заполнение таблиц данным. Разработка объектов промежуточного слоя (представлений, хранимых процедур, UDF-ов).

Разработка стратегии резервного копирования

Содержание пояснительной записки:

Содержание, Аннотация, Цель работы, Введение, Задачи, Формулировка задания, Формулировка задачи, Формализация задания, Описание сущностей, Спецификация, Создание базы данных, Создание таблиц и ограничений целостности, Заполнение таблиц данными, Создание объектов промежуточного слоя, Разработка стратегии резервного копирования, Заключение, Список использованных источников

Предполагаемый объем пояснительной записки:

Не менее 25 страниц.

Дата выдачи задания: 09.09.2024

Дата сдачи курсовой работы: 29.01.2025

Дата защиты курсовой работы: 29.01.2025

Студент

Максимов Ю.Е.

Преподаватель

Новакова Н.Е.

Аннотация

В курсовой работе представлена база данных аптеки. Выделены следующие сущности:

1. Category - категории
2. Customer - клиенты
3. Sale – продажи
4. ReservationDate – дата заказа
5. Reservation – заказы
6. Supplier - поставщик
7. Inventory - запасы
8. Product - продукты
9. StockMovment - транзакции

Была спроектирована база данных аптеки. Были созданы таблицы базы данных, добавлены ограничения целостности, заполнены таблицы.

Были разработаны объекты промежуточного слоя: представления, хранимые процедуры, UDF-ы.

Была разработана стратегия резервного копирования.

ANNOTATION

The course work presents a pharmacy database. The following entities are highlighted:

1. Category - category
2. Customer - customer
3. Sale – sale
4. ReservationDate – reservation date
5. Reservation – reservation
6. Supplier - supplier
7. Inventory - inventory
8. Product - product
9. StockMovment – stock movment

A pharmacy database was designed. Database tables were created, integrity constraints were added, tables were populated.

Middleware objects were developed: views, stored procedures, UDFs.

A backup strategy was developed.

Оглавление

Цель работы.....	6
Введение.....	6
Задачи.....	6
Формулировка задания.....	7
Формулировка задачи.....	7
Краткое описание предметной области.....	7
Запуск БД через Docker.....	8
ER-диаграмма БД.....	9
Спецификация.....	10
Создание БД.....	19
Ограничение целостности.....	24
Разработка объектов промежуточного слоя.....	33
Разработка стратегии резервного копирования.....	41
Вывод.....	45
Список использованных источников.....	46

Цель работы

Целью курсовой работы является закрепление теоретических знаний и получение практических навыков проектирования базы данных.

Введение

В современном мире информационные технологии играют все более значимую роль во всех сферах жизнедеятельности человека. Отрасль фармацевтики не является исключением. Эффективное функционирование аптеки существенно зависит от правильно организованной системы обработки, хранения и обмена данными. В связи с этим, проектирование базы данных аптеки становится актуальной и важной задачей.

Важность этой работы определяется необходимостью использования современных информационных технологий для повышения конкурентоспособности аптечных организаций на рынке.

Задачи

1. Краткое описание предметной области.
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)

Формулировка задания

Спроектировать базу данных склада продуктов и выполнить следующие задания: разработать структуру базы данных, изобразить ее с помощью ER-диаграммы; спроектировать базу данных, относительно созданной диаграммы; создать базу данных и необходимые таблицы; задать ограничения для таблиц и ключи; заполнить базу данных; создать в базе данных объекты промежуточного слоя – хранимые процедуры, представления и UDF; выбрать стратегию резервного копирования.

Формулировка задачи

Необходимо спроектировать базу данных, содержащую таблицы с информацией о работе аптеки, в том числе информацию о работающих в ней сотрудниках и информацию обо всех посетителях.

Краткое описание предметной области

Предметная область "Склад продуктов" в контексте базы данных включает в себя хранение, учет и управление информацией о товарах, которые находятся на складе. Цель такой базы данных – оптимизация операций по управлению запасами и облегчение процессов логистики. Основные объекты, которые обычно моделируются в базе данных склада продуктов, включают:

Продукты – описание товаров, хранящихся на складе (название, категория, уникальный код, единица измерения и прочее). Поставщики – данные о компаниях, которые поставляют продукты на склад (название компании, контакты, договоры и условия поставки). Запасы – информация об объеме и текущем состоянии продуктов на складе (количество, дата поступления, дата истечения срока годности, место хранения). Заказы – сведения о движении товаров, включая заказы от клиентов, внутренние

перемещения и возвраты. Сотрудники – данные о работниках, которые ответственны за управление складом и учёт продуктов. Локации хранения – информация о конкретных местах хранения товаров на складе (регионы, стеллажи, ячейки и т.д.). Использование базы данных для склада продуктов позволяет управлять всеми этими данными, следить за запасами, планировать пополнение, отслеживать сроки годности и контролировать логистику на складе, тем самым улучшая эффективность работы.

Запуск БД через Docker

```
sudo docker run -e "ACCEPT_EULA=Y" -e  
"MSSQL_SA_PASSWORD=Strong!!" -p 1433:1433 --  
name mssql1 -d mcr.microsoft.com/mssql/server:2022-  
latest Логин - sa Пароль - Strong!!
```


ER-диаграмма БД



Спецификация

Таблица 1. Таблица Категория

Описание структуры таблицы БД	Наименование таблицы БД: Категория	Имя таблицы: Category
Дата разработки: 25.10.2024		
Порядковый номер таблицы: 1		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Идентификатор категории	ID	int	P	Not null
2	Имя категории	CategoryName	Nvarchar(50)		Not null
3	Аннотация	Description	Nvarchar(50)		Null

Таблица 2. Таблица Покупатели

Описание структуры таблицы БД	Наименование таблицы БД: Покупатели	Имя таблицы: Customer
Дата разработки: 25.10.2024		
Порядковый номер таблицы: 2		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Идентификатор покупателя	ID	int	P	Not null
2	Имя покупателя	CustomrName	Nvarchar(50)		Not null
3	Контактное имя	ContactName	Nvarchar(50)		Null
4	Адресс	Address	Nvarchar(50)		Null
5	Город	City	Nvarchar(50)		Null
6	Страна	Country	Nvarchar(50)		Null
7	Телефон	Phone	Nvarchar(50)		Null

Таблица 3. Таблица Продажи

Описание структуры таблицы БД	Наименование таблицы БД: Продажи		Имя таблицы: Sale			
Дата разработки: 25.10.2024						
Порядковый номер таблицы: 3						
№ п/п	Наименование поля	Спецификация данных				
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти	
	1	Идентификатор продажи	SAleId	int	P	Not null
	2	Индификатор продажи	ProductId	int	F	Not null
	3	Индификатор покупаетля	CustomerId	int	F	Null
	4	Дата продажи	SaleDate	date		Null
	5	Число	Quantity	int		Null
	6	Цена	UnitPrice	decimal		Null

Таблица 4. Таблица Детали заказа

Описание структуры таблицы БД	Наименование таблицы БД: Детали Заказа	Имя таблицы: ReservationDeta
Дата разработки: 25.10.2024		
Порядковый номер таблицы: 4		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Идентификатор деталей продажи	OrderDetailId	int	P	Not null
2	Индификатор продажи	OrderId	int	F	Not null
3	Индификатор покупаетля	ProductId	int	F	Null
4	Число	Quantity	int		Null
5	Цена	UnitPrice	decimal		Null

Таблица 5. Таблица Детали резервирования

Описание структуры таблицы БД	Наименование таблицы БД: Заказ	Имя таблицы: Reservation
Дата разработки: 25.10.2024		
Порядковый номер таблицы: 5		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Идентификатор продажи	OrderId	int	P	Not null
2	Дата продажи	OrderDate	Date		Not null
3	Индификатор покупателя	SupplierId	int	F	Null
4	Число	TotalAmount	int		Null
5	Статус	Status	Nvarchar(50)		Null
6	Дата отправки	DeliveryDate	Date		

Таблица 6. Таблица Поставщики

Описание структуры таблицы БД	Наименование таблицы БД: Поставщики	Имя таблицы: Supplier
Дата разработки: 25.10.2024		
Порядковый номер таблицы: 6		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Идентификатор поставщиков	OrderId	int	P	Not null
2	Имя поставщиков	OrderDate	Date		Not null
3	Контактное лицо	SupplierId	int	F	Null
4	Адресс	Address	Nvarchar(50)		Null
5	Город	City	Nvarchar(50)		Null
6	Страна	Country	Nvarchar(50)		Null
7	Телефон	Phone	Nvarchar(50)		Null

Таблица 7. Таблица Инвентарь

Описание структуры таблицы БД	Наименование таблицы БД: Инвентарь	Имя таблицы: Invetory
Дата разработки: 25.10.2024		
Порядковый номер таблицы: 7		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Идентификатор инветоря	InvetoryId	int	P	Not null
2	Идентификатор продукта	ProductId	int	F	Not null
3	Сколько есть	QuantityInStoc k	int		Null
4	Последний раз обновлено	LastUpdated	Date		Null

Таблица 8. Таблица Продукт

Описание структуры таблицы БД	Наименование таблицы БД: Продукт	Имя таблицы: Product
Дата разработки: 25.10.2024		
Порядковый номер таблицы: 8		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Идентификатор продукта	ProductId	int	P	Not null
2	Имя продукта	ProductName	Nvarchar(50)	F	Not null
3	Идентификатор категории	CategoryId	int	F	Null
4	Идентификатор поставщика	SupplierId	int	F	Null
5	Колчиество	QuantityPerUn it	Int		

6	ценна	UnitPrice	Decimal		
7	В наличии на складе	UnitInStock	Int		
8	В налации в заказе	UnitsOnOrder	Int		
9	Уровень заказа	ReorderLevel	Int		
10	Со скидкой	Discontinued	Bool		
11	Срок годности	ExpritationDate	Date		

Создание БД

Структура базы данных "Продуктовый склад"

1. Product (Продукт)

Таблица для хранения информации о каждом продукте.

```
CREATE TABLE Product (  
ProductID INT PRIMARY KEY IDENTITY,  
ProductName NVARCHAR(100) NOT NULL,  
CategoryID INT,  
SupplierID INT,  
QuantityPerUnit NVARCHAR(50),  
UnitPrice DECIMAL(10, 2),  
UnitsInStock INT DEFAULT 0,  
UnitsOnOrder INT DEFAULT 0,  
ReorderLevel INT DEFAULT 0,  
Discontinued BIT DEFAULT 0,  
ExpirationDate DATE  
);
```

2. Category (Категория)

Таблица для хранения категорий продуктов.

```
CREATE TABLE Category (  
CategoryID INT PRIMARY KEY IDENTITY,  
CategoryName NVARCHAR(50) NOT NULL,  
Description  
NVARCHAR(255) );  
  
alter table Product add constraint products_categories_id foreign key  
(CategoryID) references Category(CategoryID)
```

3. Supplier (Поставщик)

Таблица для хранения информации о поставщиках.

```
CREATE TABLE Supplier (  
SupplierID INT PRIMARY KEY IDENTITY,  
SupplierName NVARCHAR(100) NOT NULL,  
ContactName NVARCHAR(100),  
Address NVARCHAR(255),  
City NVARCHAR(50),  
PostalCode NVARCHAR(20),  
Country NVARCHAR(50),  
Phone NVARCHAR(20)  
);
```

4. Reservation (Заказ)

Таблица для хранения заказов на продукты.

```
CREATE TABLE Reservation (  
OrderID INT PRIMARY KEY IDENTITY,  
OrderDate DATE NOT NULL,  
SupplierID INT,  
TotalAmount DECIMAL(10, 2),  
Status NVARCHAR(50),  
DeliveryDate DATE,  
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)  
);
```

5. ReservationDetails (Детали Заказа) Таблица для хранения

подробностей по каждому заказу, включая продукты и их количество.

```
CREATE TABLE ReservationDetails (  
OrderDetailID INT PRIMARY KEY IDENTITY,
```

```

OrderID INT,
ProductID INT,
Quantity INT,
UnitPrice DECIMAL(10, 2),
FOREIGN KEY (OrderID) REFERENCES Reservation(OrderID),
FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);

```

6. Inventory (Запасы) Таблица для хранения информации о запасах продуктов на складе.

```

CREATE TABLE Inventory (
InventoryID INT PRIMARY KEY IDENTITY,
ProductID INT,
QuantityInStock INT,
LastUpdated DATE DEFAULT GETDATE(),
FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);

```

7. StockMovements (Движения Запасов)

Таблица для отслеживания движений запасов (приход, расход).

```

CREATE TABLE StockMovements (
MovementID INT PRIMARY KEY IDENTITY,
ProductID INT,
MovementType NVARCHAR(50) CHECK (MovementType IN (N'Приход',
N'Расход')),
Quantity INT,
MovementDate DATE DEFAULT GETDATE(),
FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);

```

8. Customer (Клиент)

Таблица для хранения информации о клиентах, если склад также обслуживает розничные продажи.

```
CREATE TABLE Customer (  
CustomerID INT PRIMARY KEY IDENTITY,  
CustomerName NVARCHAR(100),  
ContactName NVARCHAR(100),  
Address NVARCHAR(255),  
City NVARCHAR(50),  
PostalCode NVARCHAR(20),  
Country NVARCHAR(50),  
Phone NVARCHAR(20)  
);
```

9. Sale (Продажи)

Таблица для хранения информации о продажах продуктов.

```
CREATE TABLE Sale (  
SaleID INT PRIMARY KEY IDENTITY,  
ProductID INT,  
CustomerID INT,  
SaleDate DATE DEFAULT GETDATE(),  
Quantity INT,  
UnitPrice DECIMAL(10, 2),  
FOREIGN KEY (ProductID) REFERENCES Product(ProductID),  
FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)  
);
```

Связи между таблицами

- Product связан с Category по CategoryID.
- Product связан с Supplier по SupplierID.
- Reservation связан с Supplier по SupplierID.
- ReservationDetails связан с Reservation и Product.
- Inventory связан с Product.
- StockMovements связан с Product.
- Sale связан с Product и Customer.

Эта структура должна покрыть основные операции продуктового склада, такие как управление продуктами, запасами, заказами и продажами.

Ограничение целостности

1. Ограничения внешних ключей (Foreign Key Constraints)

Эти ограничения обеспечат целостность связей между таблицами.

```
ALTER TABLE Product
```

```
ADD CONSTRAINT FK_Products_Categories
```

```
FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID) ON  
DELETE SET NULL;
```

```
ALTER TABLE Product
```

```
ADD CONSTRAINT FK_Products_Suppliers
```

```
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID) ON DELETE  
SET NULL;
```

```
ALTER TABLE Reservation
```

```
ADD CONSTRAINT FK_Orders_Suppliers
```

```
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID) ON DELETE  
SET NULL;
```

```
ALTER TABLE ReservationDetails
```

```
ADD CONSTRAINT FK_OrderDetails_Orders
```

```
FOREIGN KEY (OrderID) REFERENCES Reservation(OrderID) ON DELETE  
CASCADE;
```

```
ALTER TABLE ReservationDetails
```

```
ADD CONSTRAINT FK_OrderDetails_Products
```

```
FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE  
CASCADE;
```

```
ALTER TABLE Inventory
```

```
ADD CONSTRAINT FK_Inventory_Products
```



```
FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE  
CASCADE;
```

```
ALTER TABLE StockMovements
```

```
ADD CONSTRAINT FK_StockMovements_Products
```

```
FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE  
CASCADE;
```

```
ALTER TABLE Sale
```

```
ADD CONSTRAINT FK_Sales_Products
```

```
FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE  
CASCADE;
```

```
ALTER TABLE Sale
```

```
ADD CONSTRAINT FK_Sales_Customers
```

```
FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID) ON  
DELETE SET NULL;
```

2. Ограничения уникальности (Unique Constraints)

Уникальные ограничения помогут предотвратить дублирование в полях, которые должны быть уникальными.

```
ALTER TABLE Category
```

```
ADD CONSTRAINT UQ_Categories_CategoryName UNIQUE (CategoryName);
```

```
ALTER TABLE Supplier
```

```
ADD CONSTRAINT UQ_Suppliers_SupplierName UNIQUE (SupplierName);
```

```
ALTER TABLE Customer
```

```
ADD CONSTRAINT UQ_Customers_CustomerName UNIQUE (CustomerName);
```

3. Ограничения на уровень данных (Check Constraints)

Ограничения CHECK обеспечат, что данные соответствуют определенным условиям.

```
ALTER TABLE Product
```

```
ADD CONSTRAINT CHK_Products_UnitPrice CHECK (UnitPrice >= 0);
```

```
ALTER TABLE Product
```

```
ADD CONSTRAINT CHK_Products_UnitsInStock CHECK (UnitsInStock >= 0);
```

```
ALTER TABLE Product
```

```
ADD CONSTRAINT CHK_Products_UnitsOnOrder CHECK (UnitsOnOrder >= 0);
```

```
ALTER TABLE Product
```

```
ADD CONSTRAINT CHK_Products_ReorderLevel CHECK (ReorderLevel >= 0);
```

```
ALTER TABLE ReservationDetails
```

```
ADD CONSTRAINT CHK_OrderDetails_Quantity CHECK (Quantity > 0);
```

```
ALTER TABLE ReservationDetails
```

```
ADD CONSTRAINT CHK_OrderDetails_UnitPrice CHECK (UnitPrice >= 0);
```

```
ALTER TABLE Inventory
```

```
ADD CONSTRAINT CHK_Inventory_QuantityInStock CHECK  
(QuantityInStock >= 0);
```

```
ALTER TABLE StockMovements
```

```
ADD CONSTRAINT CHK_StockMovements_Quantity CHECK (Quantity > 0);
```

```
ALTER TABLE Sale
```

```
ADD CONSTRAINT CHK_Sales_Quantity CHECK (Quantity > 0);
```

```
ALTER TABLE Sale
```

```
ADD CONSTRAINT CHK_Sales_UnitPrice CHECK (UnitPrice >= 0);
```

4. Ограничения NOT NULL (Not Null Constraints)

Эти ограничения предотвратят вставку NULL значений в обязательные поля.

```
ALTER TABLE Product
```

```
ALTER COLUMN ProductName NVARCHAR(100) NOT NULL;
```

```
ALTER TABLE Category
```

```
ALTER COLUMN CategoryName NVARCHAR(50) NOT NULL;
```

```
ALTER TABLE Supplier
```

```
ALTER COLUMN SupplierName NVARCHAR(100) NOT NULL;
```

```
ALTER TABLE Reservation
```

```
ALTER COLUMN OrderDate DATE NOT NULL;
```

```
ALTER TABLE Reservation
```

```
ALTER COLUMN TotalAmount DECIMAL(10, 2) NOT NULL;
```

```
ALTER TABLE ReservationDetails
```

```
ALTER COLUMN Quantity INT NOT NULL;
```

```
ALTER TABLE ReservationDetails
```

```
ALTER COLUMN UnitPrice DECIMAL(10, 2) NOT NULL;
```

```
ALTER TABLE Inventory
```

```
ALTER COLUMN QuantityInStock INT NOT NULL;
```

```
ALTER TABLE StockMovements
```

```
ALTER COLUMN MovementType NVARCHAR(50) NOT NULL;
```

```
ALTER TABLE StockMovements
```

```
ALTER COLUMN Quantity INT NOT NULL;
```

```
ALTER TABLE StockMovements
```

```
ALTER COLUMN MovementDate DATE NOT NULL;
```

```
ALTER TABLE Sale
```

```
ALTER COLUMN SaleDate DATE NOT NULL;
```

```
ALTER TABLE Sale
```

```
ALTER COLUMN Quantity INT NOT NULL;
```

```
ALTER TABLE Sale
```

```
ALTER COLUMN UnitPrice DECIMAL(10, 2) NOT NULL;
```

5. Ограничения значения MovementType в StockMovements Это ограничение гарантирует, что тип движения (MovementType) может быть только "Приход" или "Расход".

```
ALTER TABLE StockMovements
```

```
ADD CONSTRAINT CHK_StockMovements_MovementType CHECK
```

```
(MovementType IN (N'Приход',
```

```
N'Расход'));
```

Заполнение базы данных данными

1. Вставка данных в таблицу Category

```
INSERT INTO Category (CategoryName, Description)
VALUES
('Молочные продукты', 'Все виды молочной продукции'),
('Фрукты', 'Свежие фрукты'),
('Овощи', 'Свежие овощи'),
('Мясо', 'Мясные продукты'),
('Снеки', 'Закуски и снеки');
```

2. Вставка данных в таблицу Supplier

```
INSERT INTO Supplier (SupplierName, ContactName, Address, City, PostalCode,
Country, Phone)
VALUES
('Поставщик Молока', 'Иван Иванов', 'Улица Мира, 1', 'Москва', '123456',
'Россия',
'123-456-7890'),
('Фрукты и Овощи', 'Петр Петров', 'Улица Ленина, 5', 'Санкт-Петербург',
'654321',
'Россия', '987-654-3210'),
('Мясной Двор', 'Сергей Сергеев', 'Улица Центральная, 10', 'Казань',
'223344',
'Россия', '555-666-7777');
```

3. Вставка данных в таблицу Product

```
INSERT INTO Product (ProductName, CategoryID, SupplierID, QuantityPerUnit,
UnitPrice,
```

UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued, ExpirationDate)
VALUES

(N'Молоко', 1, 1, N'1 литр', 50.00, 100, 20, 10, 0, '2024-12-31'),
(N'Яблоки', 2, 2, N'1 кг', 120.00, 50, 10, 5, 0, '2024-01-15'),
(N'Морковь', 3, 2, N'1 кг', 80.00, 70, 15, 10, 0, '2024-02-01'),
(N'Говядина', 4, 3, N'1 кг', 500.00, 30, 5, 5, 0, '2024-12-25'),
(N'Чипсы', 5, 1, N'100 г', 30.00, 200, 50, 20, 0, '2024-05-10');

4. Вставка данных в таблицу Reservation

INSERT INTO Reservation (OrderDate, SupplierID, TotalAmount, Status,
DeliveryDate)
VALUES

('2023-11-01', 1, 5000.00, N'Доставлен', '2023-11-02'),
('2023-11-05', 2, 3000.00, N'В пути', '2023-11-06'),
('2023-11-10', 3, 7500.00, N'Ожидается', '2023-11-15');

5. Вставка данных в таблицу ReservationDetails

INSERT INTO ReservationDetails (OrderID, ProductID, Quantity, UnitPrice)
VALUES

(1, 1, 100, 50.00),
(1, 5, 50, 30.00),
(2, 2, 25, 120.00),
(2, 3, 30, 80.00),
(3, 4, 15, 500.00);

6. Вставка данных в таблицу Inventory

INSERT INTO Inventory (ProductID, QuantityInStock, LastUpdated)
VALUES

```
(1, 100, GETDATE()),  
(2, 50, GETDATE()),  
(3, 70, GETDATE()),  
(4, 30, GETDATE()),  
(5, 200, GETDATE());
```

7. Вставка данных в таблицу StockMovements

```
INSERT INTO StockMovements (ProductID, MovementType, Quantity,  
MovementDate)
```

```
VALUES
```

```
(1, N'Приход', 50, '2023-10-01'),  
(1, N'Расход', 10, '2023-10-05'),  
(2, N'Приход', 30, '2023-10-10'),  
(2, N'Расход', 5, '2023-10-12'),  
(3, N'Приход', 20, '2023-10-15'),  
(4, N'Расход', 3, '2023-10-18'),  
(5, N'Приход', 100, '2023-10-20');
```

8. Вставка данных в таблицу Customer

```
INSERT INTO Customer (CustomerName, ContactName, Address, City,  
PostalCode, Country, Phone)
```

```
VALUES
```

```
(N'Магазин 1', N'Анна Андреева', N'Улица Победы, 25', N'Москва', '125123',  
N'Россия', '789-  
123-4567'),  
(N'Магазин 2', N'Виктор Викторович', N'Улица Заречная, 7', N'Санкт-  
Петербург', '121212',
```

```
N'Россия', '456-789-1230');
```

9. Вставка данных в таблицу Sale

```
INSERT INTO Sale (ProductID, CustomerID, SaleDate, Quantity, UnitPrice)
VALUES
(1, 1, '2023-11-12', 10, 50.00),
(2, 2, '2023-11-13', 5, 120.00),
(3, 1, '2023-11-14', 7, 80.00),
(4, 2, '2023-11-15', 3, 500.00),
(5, 1, '2023-11-16', 20, 30.00);
```

Эти скрипты добавят тестовые данные в каждую из таблиц базы данных "Продуктовый склад".

Разработка объектов промежуточного слоя (представлений, хранимых процедур, UDF-ов)

Для разработки промежуточного слоя базы данных на SQL Server можно использовать представления (views), хранимые процедуры (stored procedures) и пользовательские функции (userdefined functions, UDF). Вот примеры SQL-скриптов для их создания в базе данных "Продуктовый склад".

1. Представления (Views)

- Представление для отображения всей информации о продуктах вместе с категориями и поставщиками

```
CREATE VIEW
vw_ProductInfo AS SELECT
p.ProductID,
p.ProductName,
c.CategoryName,
s.SupplierName,
p.QuantityPerUnit,
p.UnitPrice,
p.UnitsInStock,
p.UnitsOnOrder,
p.ReorderLevel,
p.Discontinued,
p.ExpirationDate
FROM
Product p
JOIN
Category c ON p.CategoryID = c.CategoryID
```

JOIN

Supplier s ON p.SupplierID = s.SupplierID;

- Представление для отображения заказов и их деталей

CREATE VIEW vw_OrderDetails AS

SELECT

o.OrderID,

o.OrderDate,

o.DeliveryDate,

o.Status,

s.SupplierName,

d.ProductID,

p.ProductName,

d.Quantity,

d.UnitPrice,

(d.Quantity * d.UnitPrice) AS TotalPrice

FROM

Reservation o

JOIN

ReservationDetails d ON o.OrderID =

```
d.OrderID JOIN
Product p ON d.ProductID =
p.ProductID JOIN
Supplier s ON o.SupplierID = s.SupplierID;
```

2.Хранимые процедуры (Stored Procedures)

- Процедура для добавления нового продукта

```
CREATE PROCEDURE sp_AddProduct
@ProductName NVARCHAR(100),
@CategoryID INT,
@SupplierID INT,
@QuantityPerUnit NVARCHAR(50),
@UnitPrice DECIMAL(10, 2),
@UnitsInStock INT,
@UnitsOnOrder INT,
@ReorderLevel INT,
@ExpirationDate DATE
AS
BEGIN
INSERT INTO Product (ProductName, CategoryID, SupplierID, QuantityPerUnit,
UnitPrice,
UnitsInStock, UnitsOnOrder, ReorderLevel, ExpirationDate)
VALUES (@ProductName, @CategoryID, @SupplierID, @QuantityPerUnit,
@UnitPrice,
@UnitsInStock, @UnitsOnOrder, @ReorderLevel, @ExpirationDate);
END;
```

- Процедура для обновления количества запасов для продукта

```
CREATE PROCEDURE sp_UpdateInventory
@ProductID INT,
@QuantityChange INT,
@MovementType NVARCHAR(50)
AS
BEGIN
IF @MovementType = N'Приход'
BEGIN
UPDATE Inventory
SET QuantityInStock = QuantityInStock + @QuantityChange, LastUpdated =
GETDATE()
WHERE ProductID = @ProductID;
END
ELSE IF @MovementType = N'Расход'
BEGIN
UPDATE Inventory
SET QuantityInStock = QuantityInStock - @QuantityChange, LastUpdated =
GETDATE()
WHERE ProductID = @ProductID;
END
ELSE
BEGIN
RAISERROR (N'Неверный тип движения. Используйте "Приход" или
"Расход".', 16, 1);
END;
```

-- Запись в таблицу движения запасов

```
INSERT INTO StockMovements (ProductID, MovementType, Quantity,  
MovementDate)  
VALUES (@ProductID, @MovementType, @QuantityChange, GETDATE());  
END;
```

- Процедура для получения информации о запасах для конкретного продукта

```
CREATE PROCEDURE sp_GetProductInventory  
@ProductID INT  
AS  
BEGIN  
SELECT  
p.ProductName,  
i.QuantityInStock,  
i.LastUpdated  
FROM  
Inventory i  
JOIN  
Product p ON i.ProductID =  
p.ProductID WHERE  
i.ProductID = @ProductID;  
END;
```

3. Пользовательские функции (User-Defined Functions, UDF)

- Функция для расчета стоимости заказа по его ID

```
CREATE FUNCTION fn_CalculateOrderTotal (@OrderID INT)  
RETURNS DECIMAL(10, 2)  
AS  
BEGIN
```

```

DECLARE @Total DECIMAL(10, 2);

SELECT @Total = SUM(d.Quantity * d.UnitPrice)
FROM ReservationDetails d
WHERE d.OrderID = @OrderID;

RETURN ISNULL(@Total, 0);

END;

```

- Функция для проверки доступности продукта на складе

```

CREATE FUNCTION fn_CheckProductAvailability (@ProductID INT)
RETURNS BIT
AS
BEGIN
DECLARE @Available BIT;

IF EXISTS (
SELECT 1
FROM Inventory
WHERE ProductID = @ProductID AND QuantityInStock > 0
)
SET @Available = 1;
ELSE
SET @Available = 0;

RETURN @Available;

END;

```

4.Пример использования хранимых процедур и функций

- Использование `sp_AddProduct` для добавления нового продукта:

```
EXEC sp_AddProduct N'Сыр', 1, 1, N'200 г', 150.00, 100, 10, 5, '2024-06-01';
```

ProductID	ProductName	CategoryID	SupplierID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued	ExpirationDate
1	Сыр	1		1 200 г	150.00	100	10	5	false	2024-06-01

- Использование `sp_UpdateInventory` для добавления поступления продукта:

```
EXEC sp_UpdateInventory @ProductID = 1, @QuantityChange = 50, @MovementType = N'Приход';
```

ProductID	QuantityInStock	LastUpdated
1	150	2025-01-24

- Использование функции `fn_CalculateOrderTotal` для получения общей стоимости заказа:

```
SELECT dbo.fn_CalculateOrderTotal(1) AS OrderTotal;
```

OrderTotal
1 6500.00

- Использование функции `fn_CheckProductAvailability` для проверки доступности продукта:

```
SELECT dbo.fn_CheckProductAvailability(2) AS IsAvailable;
```

IsAvailable
1 true

- Использование представления `vw_OrderDetails` для отображения заказов и их деталей:

```
SELECT * FROM vw_OrderDetails;
```

	OrderID	OrderDate	DeliveryDate	Status	SupplierName	ProductID	ProductName	Quantity	UnitPrice	TotalPrice
1	1	2023-11-01	2023-11-02	Доставлен	Поставщик Молока	1	Молоко	100	50.00	5000.00
2	1	2023-11-01	2023-11-02	Доставлен	Поставщик Молока	5	Чипсы	50	30.00	1500.00
3	2	2023-11-05	2023-11-06	В пути	Фрукты и Овощи	2	Яблоки	25	120.00	3000.00
4	2	2023-11-05	2023-11-06	В пути	Фрукты и Овощи	3	Морковь	30	80.00	2400.00
5	3	2023-11-10	2023-11-15	Ожидается	Мясной Двор	4	Говядина	15	500.00	7500.00

Эти объекты промежуточного слоя помогают организовать работу с данными, обеспечивая централизованный доступ к бизнес-логике и обработке данных для приложения "Продуктовый склад".

Разработка стратегии резервного копирования

Для базы данных "Продуктовый склад" на SQL Server можно разработать стратегию резервного копирования, которая обеспечит защиту данных от потерь, минимизирует простой и позволит быстро восстановить информацию. Учитывая тип данных и сценарий использования, стратегия может включать полные, дифференциальные и журнальные (транзакционные) резервные копии.

1.Цели резервного копирования

Основные цели резервного копирования:

- Обеспечение ежедневного полного резервного копирования базы данных.
- Обеспечение быстрого восстановления данных в случае сбоя или потери данных.
- Защита от потерь данных с минимальными потерями транзакций, даже при серьезных сбоях.

2.Типы резервных копий и частота выполнения

Для базы данных "Продуктовый склад" предлагаем следующую стратегию резервного копирования:

- Полное резервное копирование (Full Backup): выполняется один раз в неделю, например, в воскресенье ночью (в период минимальной нагрузки).
- Дифференциальное резервное копирование (Differential Backup): выполняется ежедневно, за исключением дня, когда выполняется полное резервное копирование.
Дифференциальное копирование содержит все изменения, произошедшие с момента последнего полного резервного копирования.

- Журнал транзакций (Transaction Log Backup): выполняется каждые 4 часа в рабочие часы (например, с 8:00 до 20:00) для минимизации потерь данных при сбое.

3. Параметры хранения и ротации

- Хранение резервных копий на удаленном сервере или облаке для предотвращения потерь в случае физического сбоя основного сервера.
- Ротация резервных копий: хранить полные резервные копии за последние 4 недели, дифференциальные копии за последнюю неделю и журналы транзакций за последние 2 дня. Более старые копии можно удалять.

4. SQL-скрипты для настройки резервного копирования

- Полное резервное копирование

```
BACKUP DATABASE [idz]
TO DISK = '/tmp/ProductWarehouse_Full.bak'
WITH FORMAT,
NAME = N'Полное резервное копирование ProductWarehouse',
DESCRIPTION = N'Еженедельное полное резервное копирование';
```

- Дифференциальное резервное копирование

```
BACKUP DATABASE [idz]
TO DISK = '/tmp/ProductWarehouse_Diff.bak'
WITH DIFFERENTIAL,
NAME = N'Дифференциальное резервное копирование ProductWarehouse';
```

- Резервное копирование журнала транзакций

```
BACKUP LOG [idz]
TO DISK = '/tmp/ProductWarehouse_Log.trn'
WITH NAME = N'Резервное копирование журнала транзакций
ProductWarehouse';
```

5. Настройка автоматического расписания Для автоматизации можно использовать SQL Server Agent. Шаги настройки расписания через SQL Server Management Studio (SSMS):

- Полное резервное копирование:
 - Создайте задание (Job) в SQL Server Agent и установите расписание на еженедельное выполнение в воскресенье ночью.
- Дифференциальное резервное копирование:
 - Добавьте новое задание для ежедневного выполнения дифференциального резервного копирования (кроме воскресенья).
 - Журналы транзакций:
 - Создайте задание для резервного копирования журналов транзакций с расписанием каждые 4 часа.

6. Тестирование и проверка восстановления

- Регулярно проверяйте резервные копии на возможность восстановления данных, чтобы убедиться, что копии пригодны для использования:

```
RESTORE VERIFYONLY
```

```
FROM DISK = '/tmp/ProductWarehouse_Full.bak';
```

- Для восстановления базы данных

```
RESTORE DATABASE idz FROM DISK = '/tmp/ProductWarehouse_Full.bak'  
WITH MOVE 'idz' TO
```

```
'/var/opt/mssql/data/idz.mdf', MOVE 'idz_log' TO '/var/opt/mssql/data/idz_log.ldf';
```

- Узнать имя бд и логов

```
RESTORE FILELISTONLY FROM DISK = '/tmp/ProductWarehouse_Full.bak';
```

7. Мониторинг и оповещения

- Мониторинг выполнения задач резервного копирования: настроить уведомления в SQL Server Agent для отправки сообщений при успешном завершении или ошибках.
- Оповещения: в случае ошибки задачи SQL Server Agent может отправлять уведомления администраторам базы данных.

8.Дополнительные меры

- Архивирование: резервные копии за месяц можно перемещать на внешние носители или в облачное хранилище для длительного хранения.
- Шифрование и сжатие резервных копий: использовать опции WITH ENCRYPTION и WITH COMPRESSION для повышения безопасности и снижения объема данных

Вывод

В ходе выполнения работы были изучены методы проектирования и механизмы создания баз данных: создание БД, создание таблиц и ограничений целостности, добавление данных в таблицы, разработка объектов промежуточного слоя, разработка стратегии промежуточного копирования.

Список использованных источников

1. Горячев А. В., Новакова Н. Е. Распределенные базы данных. Мет. указания к лаб. работам., СПб. Изд-во СПбГЭТУ «ЛЭТИ», 2008
2. Горячев А.В, Новакова Н.Е. Особенности разработки и администрирования приложений баз данных: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2016. 68 с.
3. Дейт К. Введение в системы баз данных. : Пер. с англ. – 6-е изд. - К.:Диалектика, 1998.
4. Базы данных: модели, разработка, реализация: учеб./ Т.С.Карпова. - СПб. : Питер, 2001. - 303 с.
5. Сайт о программировании и создании БД
<https://metanit.com/sql/sqlserver>
6. Документация MSSQL
<https://learn.microsoft.com/ru-ru/sql/t-sql>
7. Редактор Visio online
<https://products.aspose.app/diagram/ru/editor>