

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

ОТЧЕТ

по индивидуальному домашнему заданию
по дисциплине «Базы данных»

Тема: Проектирование и создание базы данных «Управление торговли»

Студента гр. 9308

Яловега Н.В.

Преподаватель

Жирнова О.А.

Санкт-Петербург

2021

АННОТАЦИЯ

В процессе выполнения индивидуального домашнего задания была спроектирована и создана база данных «Управление торговли». При проектировании и создании были использованы знания, полученные в процессе изучения курса «Базы данных». Были созданы и использованы объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы), построена диаграмма базы данных и выбрана стратегия резервного копирования.

SUMMARY

In the process of completing the individual homework, the Trade Management database was designed. When designing, the knowledge gained in the course of studying the course "Databases" was used. Middleware objects (views, stored procedures, UDFs) were created and used, a database diagram was built, and a backup strategy was selected.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Цель работы.....	5
Задание.....	5
Формализация задания.....	5
Формализация задачи.....	5
Краткое описание предметной области.....	6
Проектирование базы данных с использованием метода ER-диаграмм.....	7
Физическая модель данных.....	14
Создание базы данных.....	21
Создание таблиц базы данных.....	22
Разработка объектов промежуточного слоя.....	32
Представления.....	32
Хранимые процедуры.....	34
UDF.....	38
Резервное копирование.....	40
Вывод.....	41

ВВЕДЕНИЕ

В данной работе была спроектирована и создана база данных (БД) «Управление торговли», предназначенная для хранения и управления сведений о магазинах, отделах, товарах, торговых базах.

В БД предусмотрено, что отдел имеет состояние работы (открытый или закрытый), каждый товар имеет дату поступления в магазин, каждый магазин закреплен за одной торговой базой, одинаковые товары могут храниться на разных базах, одинаковые товары могут продаваться в разных отделах.

Также в процессе выполнения работы были созданы объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы), а также выбрана стратегия резервного копирования.

Цель работы

Закрепить теоретические знания, полученные на курсе «Базы данных», и получить практические навыки в проектировании и создании базы данных.

Задание

Для работы была выбрана база данных «Управление торговли». Необходимо выполнить следующие задания:

1. Краткое описание предметной области;
2. Спроектировать БД (структура данных);
3. Создать БД;
4. Создать таблицы и ограничения целостности;
5. Заполнить таблицы данными;
6. Создать объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы);
7. Разработка стратегии резервного копирования;

Формализация задания

Спроектировать базу данных для управления торговлей, используя метод ER-диаграмм, создать базу данных и необходимые таблицы, задать ограничения для таблиц и ключи, заполнить базу данных и создать в базе данных объекты промежуточного слоя.

Формализация задачи

Необходимо спроектировать базу данных, содержащую таблицы со сведениями о магазинах, отделах, товарах и торговых базах.

Краткое описание предметной области

В БД должны храниться сведения о магазинах города; товарах, имеющихся в магазинах; о торговых базах и товарах, хранящихся на базах. Каждый магазин закреплён за одной торговой базой. Магазин характеризуется классом, номером и имеет несколько отделов. Каждый товар в каждом магазине продаётся, по крайней мере, в одном отделе. Каждый отдел имеет одного заведующего отделом. Товары, имеющиеся в магазине и хранящиеся на базах, характеризуются ценой, сортом и количеством. Розничные цены в магазине зависят от класса магазина и сорта товара и могут изменяться. Магазин может открыть новый отдел или закрыть старый. В этом случае товар передаётся в другие отделы.

БД «Управление торговли» проектируем при условии, что:

- каждый магазин закреплён за одной торговой базой;
- одинаковые товары могут храниться на разных базах;
- одинаковые товары могут продаваться в разных отделах;
- каждый отдел имеет состояние (закрыт или открыт);
- у каждого товара есть дата поступления в магазин.

Проектирование базы данных с использованием метода ER-диаграмм

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

- БАЗА (НомерБ*, Название, ФИОЗаведующего, Примечание);
- МАГАЗИН (НомерМ*, Класс, Название, Адрес, ФИОДиректора, Примечание);
- ОТДЕЛ (НомерО*, ФИОЗаведующего, Состояние, Примечание);
- ТОВАР (Шифр*, Наименование, Сорт, Примечание).

Рассмотрим связь между объектами БАЗА и МАГАЗИН (рисунок 1).

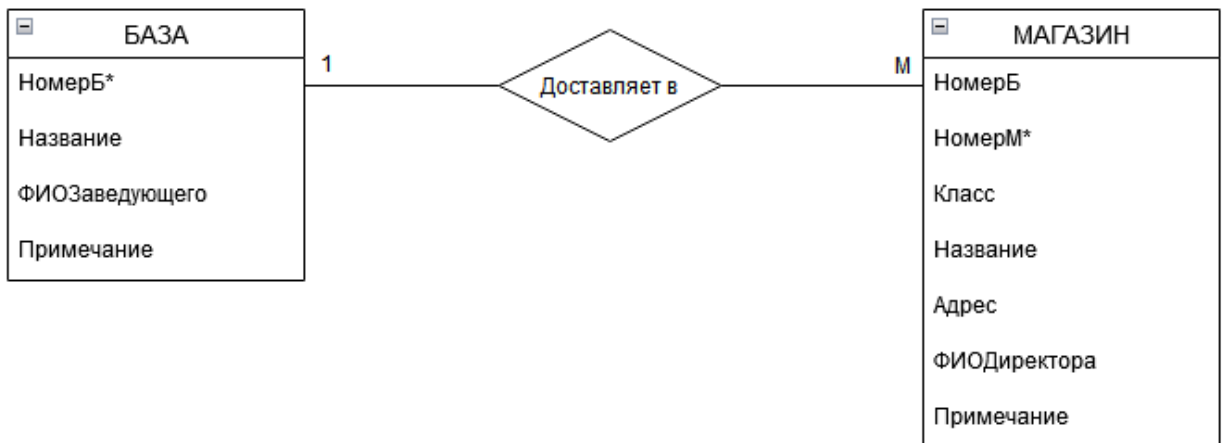


Рисунок 1

База доставляет товары в магазины, магазин получает товары с одной базы, следовательно, в соответствии с правилом 4 [1 – 3] «Если степень взаимосвязи 1:М 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:

- БАЗА (НомерБ*, Название, ФИОЗаведующего, Примечание);
- МАГАЗИН (НомерМ*, НомерБ, Класс, Название, Адрес, ФИОДиректора, Примечание).

Рассмотрим связь между объектами МАГАЗИН и ОТДЕЛ (рисунок 2).

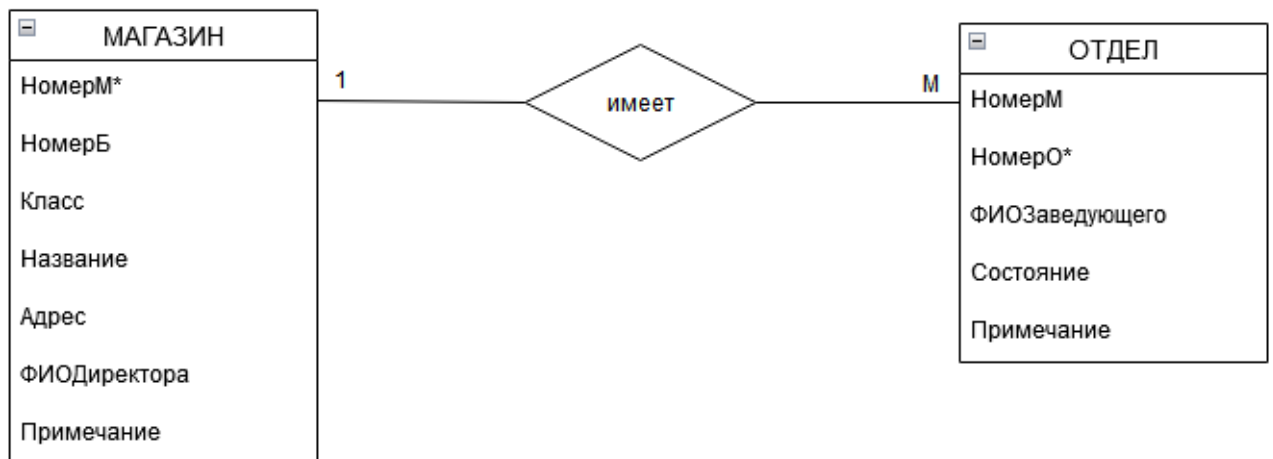


Рисунок 2

В магазине есть как минимум один отдел, и каждый отдел обязательно существует в каком-нибудь одном магазине, следовательно, в соответствии с правилом 4 [1 – 3] «Если степень взаимосвязи 1:М 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:

- МАГАЗИН (НомерМ*, НомерБ, Класс, Название, Адрес, ФИОДиректора, Примечание);
- ОТДЕЛ (НомерО*, НомерМ, ФИОЗаведующего, Примечание).

Рассмотрим связь между объектами ТОВАР и ОТДЕЛ (рисунок 3).

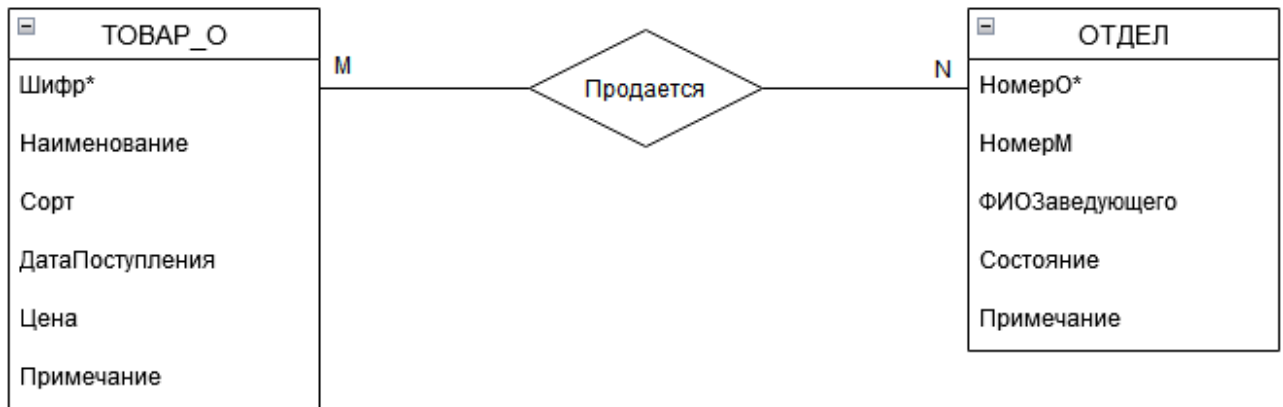


Рисунок 3

Товары могут продаваться во многих отделах, отделы могут содержать множество товаров, следовательно, в соответствии с правилом 6 «Если степень взаимосвязи двух сущностей равна $M:N$, то независимо от классов принадлежности 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:

- ТОВАР_О (Шифр*, Наименование, Сорт, ДатаПоступления, Цена, Примечание);
- ОТДЕЛ (НомерО*, НомерМ, ФИОЗаведующего, Состояние, Примечание);
- ТОВАРВОТДЕЛЕ (НомерО*, Шифр*, Количество).

Рассмотрим связь между объектами БАЗА и ТОВАР (рисунок 4).

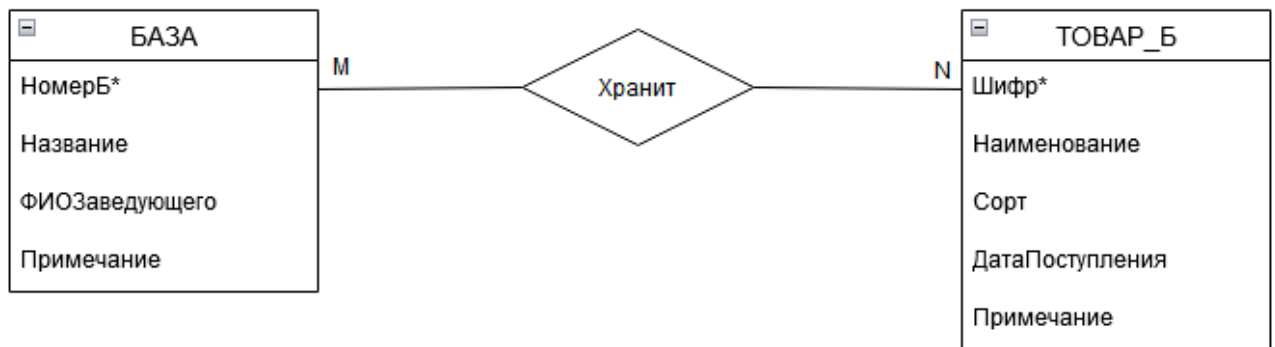


Рисунок 4

Товары могут храниться на многих базах, база может содержать множество товаров, следовательно, в соответствии с правилом 6 «Если степень взаимосвязи двух сущностей равна $M:N$, то независимо от классов принадлежности 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:

- БАЗА (НомерБ*, Название, Примечание);
- ТОВАР_Б (Шифр*, Наименование, Сорт, ДатаПоступления, Примечание);
- ТОВАРНАБАЗЕ (НомерБ*, Шифр*, Количество).

Таким образом, БД «Управление торговли» включает:

- БАЗА (НомерБ*, Название, ФИОЗаведующего, Примечание);
- МАГАЗИН (НомерМ*, НомерБ, Класс, Название, Адрес, ФИОДиректора, Примечание);
- ОТДЕЛ (НомерО*, НомерМ, ФИОЗаведующего, Состояние, Примечание);
- ТОВАР_О (Шифр*, Наименование, Сорт, ДатаПоступления, Цена, Примечание);
- ТОВАР_Б (Шифр*, Наименование, Сорт, ДатаПоступления, Примечание);
- ТОВАРВОТДЕЛЕ (НомерО*, Шифр*, Количество);
- ТОВАРНАБАЗЕ (НомерБ*, Шифр*, Количество).

На основе всех данных была построена ER-диаграмма сущностей и связей между ними. (см рис 5.)

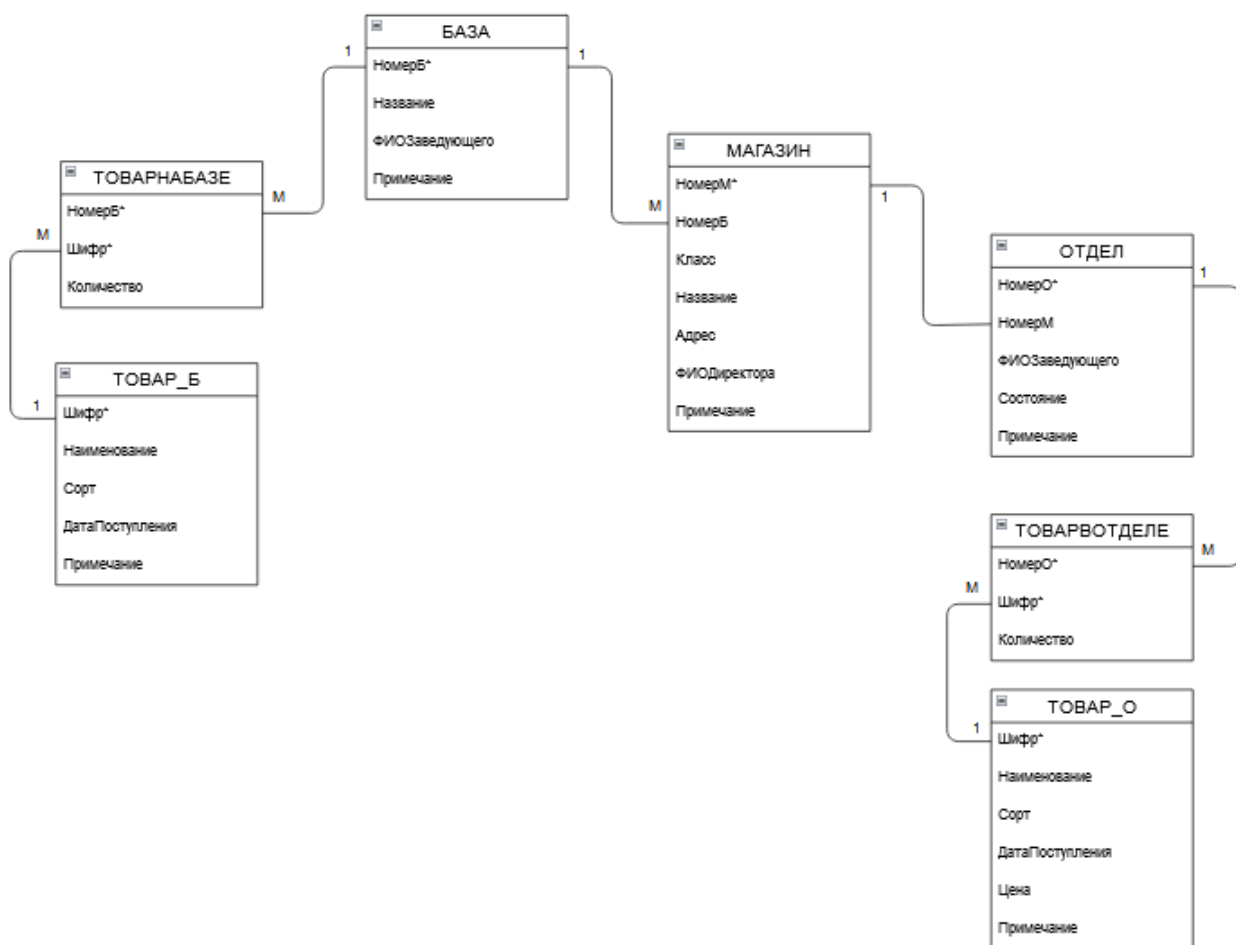


Рисунок 5

Физическая модель данных

Опишем данные в терминах T-SQL.

Таблица 1 – Спецификация таблицы bases

Описание структуры таблицы БД	Наименование таблицы БД: Таблица торговых баз	Имя таблицы: bases
Дата разработки: 17.11.2021		
Порядковый номер таблицы: 1		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Номер базы	base_number	int	PK	notNull
2	Название	name	nchar(15)		notNull
3	ФИО Управляющего	manager_name	nchar(50)		notNull
4	Примечание	note	nchar(30)		

Таблица 2 – Спецификация таблицы shops

Описание структуры таблицы БД	Наименование таблицы БД: Таблица магазинов	Имя таблицы: shops
Дата разработки: 17.11.2021		
Порядковый номер таблицы: 2		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Номер магазина	shop_number	int	PK	notNull
2	Номер базы	base_number	int	FK	notNull
3	Класс	class	nchar(50)		notNull
4	Название	name	nchar(30)		notNull
5	Адрес	address	nchar(50)		notNull
6	ФИО директора	director_name	nchar(50)		notNull
7	Примечание	note	nchar(30)		

Таблица 3 – Спецификация таблицы departments

Описание структуры таблицы БД	Наименование таблицы БД: Таблица отделов	Имя таблицы: departments
Дата разработки: 17.11.2021		
Порядковый номер таблицы: 3		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Номер отдела	department_number	int	PK	notNull
2	Номер магазина	shop_number	int	FK	notNull
3	ФИО управляющего	manager_name	nchar(50)		notNull
4	Состояние	state	bit		notNull
5	Примечание	note	nchar(30)		

Таблица 4 – Спецификация таблицы products_b

Описание структуры таблицы БД	Наименование таблицы БД: Таблица товаров базы	Имя таблицы: products_b
Дата разработки: 17.11.2021		
Порядковый номер таблицы: 4		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Шифр	code	int	P	notNull
2	Название	name	nchar(30)		notNull
3	Сорт	sort	nchar(30)		notNull
4	Дата поступления	date	datetime		notNull
5	Примечание	note	nchar(30)		

Таблица 5 – Спецификация таблицы products_d

Описание структуры таблицы БД	Наименование таблицы БД: Таблица товаров отдела	Имя таблицы: products_d
Дата разработки: 17.11.2021		
Порядковый номер таблицы: 5		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничен ия целостнос ти
1	Шифр	code	int	P	notNull
2	Название	name	nchar(30)		notNull
3	Сорт	sort	nchar(30)		notNull
4	Дата поступления	date	datetime		notNull
5	Цена	price	real		price > 0
6	Примечание	note	nchar(30)		

Таблица 6 – Спецификация таблицы products_bases

Описание структуры таблицы БД	Наименование таблицы БД: Таблица связи продуктов и торговых баз	Имя таблицы: products_bases
Дата разработки: 17.11.2021		
Порядковый номер таблицы: 6		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Номер базы	base_number	int		notNull
2	Шифр	code	int		notNull
3	Количество	amount	int		amount >0

Таблица 7 – Спецификация таблицы products_departments

Описание структуры таблицы БД	Наименование таблицы БД: Таблица связи продуктов и магазинов	Имя таблицы: products_departments
Дата разработки: 17.11.2021		
Порядковый номер таблицы: 7		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Номер отдела	department_number	int		notNull
2	Шифр	code	int		notNull
3	Количество	amount	int		amount >0

Создание базы данных

Создание базы данных «Управление торговли»:

```
CREATE DATABASE [Trade]
  CONTAINMENT = NONE
  ON PRIMARY
( NAME = N'Trade', FILENAME = N'D:\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Trade.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
  LOG ON
( NAME = N'Trade_log', FILENAME = N'D:\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Trade_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
  WITH CATALOG_COLLATION = DATABASE_DEFAULT
GO
```

На рис. 6 представлена созданная база данных Trade.

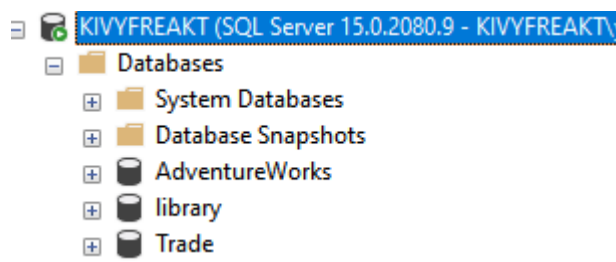


Рисунок 6

Создание таблиц базы данных

Создание таблицы bases:

```
CREATE TABLE [dbo].[bases](
    [base_number] [int] NOT NULL,
    [name] [nvarchar](15) NOT NULL,
    [manager_name] [nvarchar](50) NOT NULL,
    [note] [nvarchar](30) NULL,
    CONSTRAINT [PK_bases] PRIMARY KEY CLUSTERED
(
    [base_number] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Создание таблицы shops:

```
CREATE TABLE [dbo].[shops](
    [shop_number] [int] NOT NULL,
    [base_number] [int] NOT NULL,
    [class] [nvarchar](50) NOT NULL,
    [name] [nvarchar](30) NOT NULL,
    [address] [nvarchar](50) NOT NULL,
    [director_name] [nvarchar](50) NOT NULL,
    [note] [nvarchar](30) NULL,
    CONSTRAINT [PK_shops] PRIMARY KEY CLUSTERED
(
    [shop_number] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[shops] WITH CHECK ADD CONSTRAINT [FK_shops_bases] FOREIGN
KEY([base_number])
REFERENCES [dbo].[bases] ([base_number])
GO

ALTER TABLE [dbo].[shops] CHECK CONSTRAINT [FK_shops_bases]
GO
```

Создание таблицы departments:

```
CREATE TABLE [dbo].[departments](  
  
    [department_number] [int] NOT NULL,  
    [shop_number] [int] NOT NULL,  
    [manager_name] [nvarchar](50) NOT NULL,  
    [state] [bit] NOT NULL,  
    [note] [nvarchar](30) NULL,  
    CONSTRAINT [PK_departments] PRIMARY KEY CLUSTERED  
(  
        [department_number] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
) ON [PRIMARY]  
GO  
  
ALTER TABLE [dbo].[departments] WITH CHECK ADD CONSTRAINT [FK_departments_departments]  
FOREIGN KEY([shop_number])  
REFERENCES [dbo].[shops] ([shop_number])  
GO  
  
ALTER TABLE [dbo].[departments] CHECK CONSTRAINT [FK_departments_departments]  
GO
```

Создание таблицы products_b:

```
CREATE TABLE [dbo].[products_b](  
  
    [code] [int] NOT NULL,  
    [name] [nvarchar](30) NOT NULL,  
    [sort] [nvarchar](30) NOT NULL,  
    [date] [datetime] NOT NULL,  
    [note] [nvarchar](30) NULL,  
    CONSTRAINT [PK_products_b] PRIMARY KEY CLUSTERED  
(  
        [code] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
) ON [PRIMARY]  
GO
```

Создание таблицы products_d:

```
CREATE TABLE [dbo].[products_d](
    [code] [int] NOT NULL,
    [name] [nchar](30) NOT NULL,
    [sort] [nchar](30) NOT NULL,
    [date] [datetime] NOT NULL,
    [price] [real] NOT NULL,
    [note] [nchar](30) NULL,
    CONSTRAINT [PK_products_d] PRIMARY KEY CLUSTERED
(
    [code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[products_d] WITH CHECK ADD CONSTRAINT [CK_products_d] CHECK
(((price)>(0)))
GO

ALTER TABLE [dbo].[products_d] CHECK CONSTRAINT [CK_products_d]
GO
```


Создание таблицы products_departments:

```
CREATE TABLE [dbo].[products_departments](
    [department_number] [int] NOT NULL,
    [code] [int] NOT NULL,
    [amount] [int] NOT NULL,
    CONSTRAINT [PK_products_departments] PRIMARY KEY CLUSTERED
(
    [department_number] ASC,
    [code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[products_departments] WITH CHECK ADD CONSTRAINT
[FK_products_departments_departments] FOREIGN KEY([department_number])
REFERENCES [dbo].[departments] ([department_number])
GO

ALTER TABLE [dbo].[products_departments] CHECK CONSTRAINT
[FK_products_departments_departments]
GO

ALTER TABLE [dbo].[products_departments] WITH CHECK ADD CONSTRAINT
[FK_products_departments_products_d] FOREIGN KEY([code])
REFERENCES [dbo].[products_d] ([code])
GO

ALTER TABLE [dbo].[products_departments] CHECK CONSTRAINT
[FK_products_departments_products_d]
GO

ALTER TABLE [dbo].[products_departments] WITH CHECK ADD CONSTRAINT
[CK_products_departments] CHECK (([amount]>(0)))
GO

ALTER TABLE [dbo].[products_departments] CHECK CONSTRAINT [CK_products_departments]
GO
```

Создание таблицы products_bases:

```
CREATE TABLE [dbo].[products_bases](
    [base_number] [int] NOT NULL,
    [code] [int] NOT NULL,
    [amount] [int] NOT NULL,
    CONSTRAINT [PK_products_bases] PRIMARY KEY CLUSTERED
(
    [base_number] ASC,
    [code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[products_bases] WITH CHECK ADD CONSTRAINT [FK_products_bases_bases]
FOREIGN KEY([base_number])
REFERENCES [dbo].[bases] ([base_number])
GO

ALTER TABLE [dbo].[products_bases] CHECK CONSTRAINT [FK_products_bases_bases]
GO

ALTER TABLE [dbo].[products_bases] WITH CHECK ADD CONSTRAINT
[FK_products_bases_products_b] FOREIGN KEY([code])
REFERENCES [dbo].[products_b] ([code])
GO

ALTER TABLE [dbo].[products_bases] CHECK CONSTRAINT [FK_products_bases_products_b]
GO
```

На рис. 7 представлены созданные таблицы в базе данных Trade.

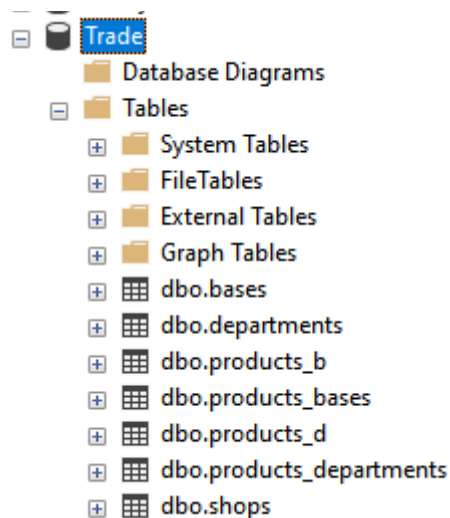


Рисунок 7

На рис.8 изображена диаграмма связей

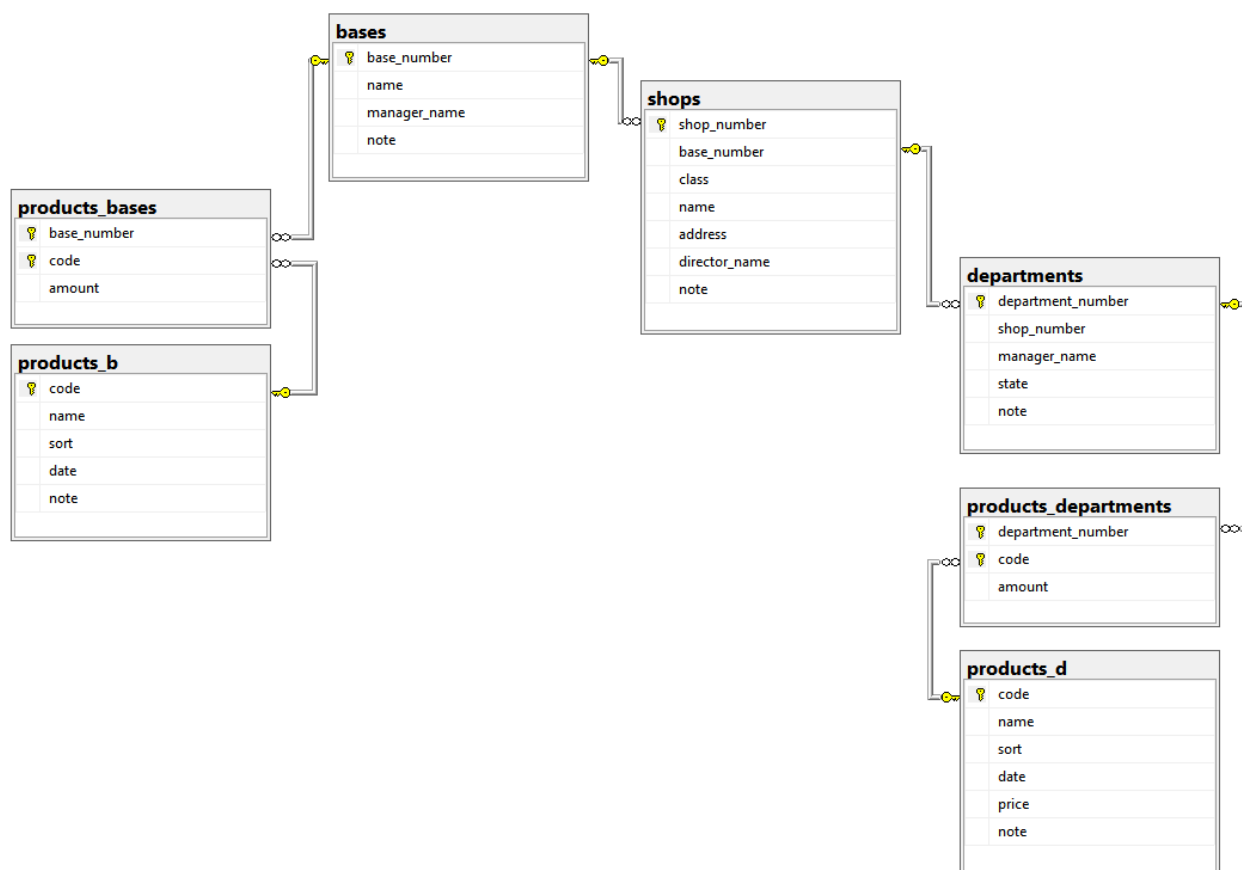


Рисунок 8

Заполнение таблиц данными

Заполнение таблицы bases:

```
INSERT INTO bases
VALUES
```

```
(189023, 'Сиверский', 'Е. Т. Мартынова', NULL),
(178023, 'Октябрьский', 'К. Н. Никулина', NULL),
(183623, 'Автопластик', 'А. А. Вавилова', NULL),
(185023, 'Воднев', 'А. А. Потапов', NULL),
(123423, 'Новотроицкая', 'П. Д. Антонова', 'В ремонте');
```

Results Messages

	base_number	name	manager_name	note
1	123423	Новотроицкая	П. Д. Антонова	В ремонте
2	178023	Октябрьский	К. Н. Никулина	NULL
3	183623	Автопластик	А. А. Вавилова	NULL
4	185023	Воднев	А. А. Потапов	NULL
5	189023	Сиверский	Е. Т. Мартынова	NULL

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 5 rows

Рисунок 9

Заполнение таблицы shops:

```
INSERT INTO shops
VALUES
```

```
(289023, 185023, 'Игрушки', 'Растишка', 'пл. Гоголя, 95', 'А. Г. Гущин', 'В ремонте'),
(278023, 189023, 'Продукты', 'Микси', 'спуск Домодедовская, 17', 'А. В. Игнатова', NULL),
(283623, 189023, 'Продукты', 'Переулоч', 'ул. Ладыгина, 31', 'А. И. Смирнов', NULL),
(285023, 178023, 'Мебель', 'Елена', 'пер. Балканская, 49', 'Д. П. Балашова', NULL),
(223423, 178023, 'Двери', 'Запилил', 'пл. Будапештская, 99', 'Д. В. Крючкова', NULL);
```

Results

Messages

	shop_number	base_number	class	name	address	director_name	note
1	223423	178023	Двери	Запилил	пл. Будапештская, 99	Д. В. Крючкова	NULL
2	278023	189023	Продукты	Микси	спуск Домодедовская, 17	А. В. Игнатова	NULL
3	283623	189023	Продукты	Переулоч	ул. Ладыгина, 31	А. И. Смирнов	NULL
4	285023	178023	Мебель	Елена	пер. Балканская, 49	Д. П. Балашова	NULL
5	289023	185023	Игрушки	Растишка	пл. Гоголя, 95	А. Г. Гущин	В ремонте

Query executed successfully.

KIVYFREAKT (15.0 RTM)

KIVYFREAKT\yalov (55)

Trade

00:00:00

5 rows

Рисунок 10

Заполнение таблицы departments:

```
INSERT INTO departments
VALUES
```

```
(1, 278023, 'А. В. Игнатова', 1, NULL),
(2, 278023, 'К. П. Овчинников', 0, 'Перестановка'),
(3, 278023, 'К. П. Овчинников', 1, NULL),
(4, 283623, 'А. И. Смирнов', 1, NULL),
(5, 285023, 'Д. П. Балашова', 1, NULL),
(6, 223423, 'Д. В. Крючкова', 1, NULL);
```

	department_number	shop_number	manager_name	state	note
1	1	278023	А. В. Игнатова	1	NULL
2	2	278023	К. П. Овчинников	0	Перестановка
3	3	278023	К. П. Овчинников	1	NULL
4	4	283623	А. И. Смирнов	1	NULL
5	5	285023	Д. П. Балашова	1	NULL
6	6	223423	Д. В. Крючкова	1	NULL

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 6 rows

Рисунок 11

Заполнение таблицы products_b:

```
INSERT INTO products_b
```

```
VALUES
```

```
(111111, 'Сырок', 'Б.У.АЛЕХИН', GETDATE(), NULL),
(111112, 'Молоко', 'Давлеканово', GETDATE(), NULL),
(111114, 'Хлеб', 'Бородинский', GETDATE(), NULL),
(222221, 'Кровать', '160x200', GETDATE(), NULL),
(222222, 'Стол', '160 x 90', GETDATE(), NULL),
(444441, 'Вода', 'Газированная', GETDATE(), NULL);
```

	code	name	sort	date	note
1	111111	Сырок	Б.У.АЛЕХИН	2021-12-24 11:48:20.950	NULL
2	111112	Молоко	Давлеканово	2021-12-24 11:48:20.950	NULL
3	111114	Хлеб	Бородинский	2021-12-24 11:48:20.950	NULL
4	222221	Кровать	160x200	2021-12-24 11:48:20.950	NULL
5	222222	Стол	160 x 90	2021-12-24 11:48:20.950	NULL
6	444441	Вода	Газированная	2021-12-24 11:48:20.950	NULL

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 6 rows

Рисунок 12

Заполнение таблицы products_bases:

```
INSERT INTO products_bases
```

```
VALUES
```

```
(189023,111111, 1000),
(189023,111112, 1234234),
(178023,111114, 23423),
(183623,222221, 7756),
(183623,222222, 45),
(123423,444441, 4643);
```

Results		Messages	
	base_number	code	amount
1	123423	444441	4643
2	178023	111114	23423
3	183623	222221	7756
4	183623	222222	45
5	189023	111111	1000
6	189023	111112	1234234

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 6 rows

Рисунок 13

Заполнение таблицы products_departments:

```
INSERT INTO products_d
```

```
VALUES
```

```
(111112, 'Молоко', 'Давлеканово', GETDATE(), 65.34, NULL),
(111114, 'Хлеб', 'Бородинский', GETDATE(), 45.32, NULL),
(111113, 'Вода', 'Газированная', GETDATE(), 45.11, NULL),
(323322, 'Матрас', 'мягкий)', GETDATE(), 23999.99, NULL),
(323324, 'Ковер', '300x300', GETDATE(), 39999.32, NULL),
(323323, 'Наушники', 'Проводные', GETDATE(), 3400.00, NULL);
```

Results

Messages

	code	name	sort	date	price	note
1	111112	Молоко	Давлеканово	2021-12-24 11:56:16.057	65,34	NULL
2	111113	Вода	Газированная	2021-12-24 11:56:16.057	45,11	NULL
3	111114	Хлеб	Бородинский	2021-12-24 11:56:16.057	45,32	NULL
4	323322	Матрас	мягкий)	2021-12-24 11:56:16.057	23999,99	NULL
5	323323	Наушники	Проводные	2021-12-24 11:56:16.057	3400	NULL
6	323324	Ковер	300x300	2021-12-24 11:56:16.057	39999,32	NULL

Query executed successfully.

KIVYFREACT (15.0 RTM)

KIVYFREACT\yalov (55)

Trade

00:00:00

6 rows

Рисунок 14

Заполнение таблицы products_b:

```
INSERT INTO products_departments
```

```
VALUES
```

```
(1,111112, 1000),  
(1,111114, 1234234),  
(3,111113, 23423),  
(4,323322, 7756),  
(4,323324, 45),  
(6,323323, 4643);
```

Results		Messages	
	department_number	code	amount
1	1	111112	1000
2	1	111114	1234234
3	3	111113	23423
4	4	323322	7756
5	4	323324	45
6	6	323323	4643

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 6 rows

Рисунок 15

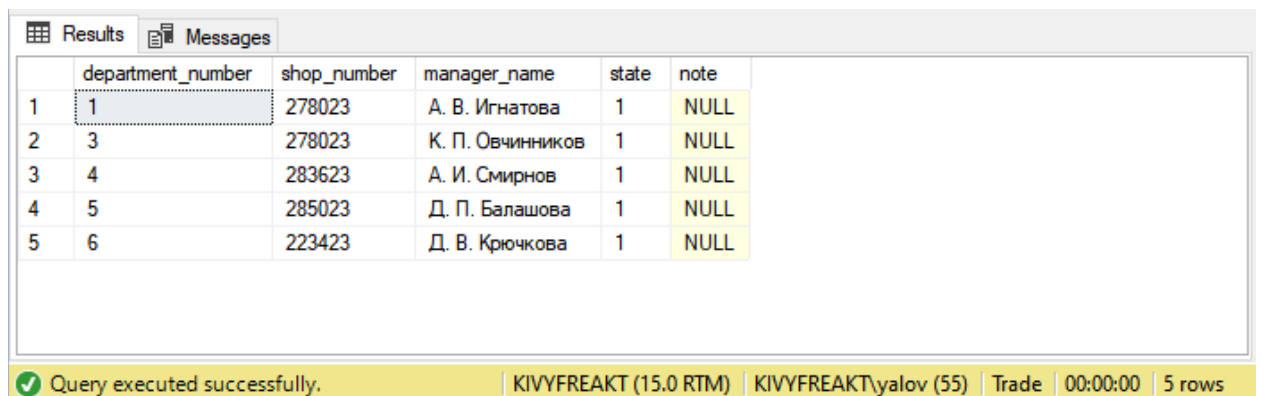
Разработка объектов промежуточного слоя

Представления

Создание представления WorkedDepartments:

```
CREATE VIEW WorkedDepartments AS
SELECT *
FROM departments
WHERE state = 1
```

Данное представление выводит информацию о отделах, которые были закрыты.



	department_number	shop_number	manager_name	state	note
1	1	278023	А. В. Игнатова	1	NULL
2	3	278023	К. П. Овчинников	1	NULL
3	4	283623	А. И. Смирнов	1	NULL
4	5	285023	Д. П. Балашова	1	NULL
5	6	223423	Д. В. Крючкова	1	NULL

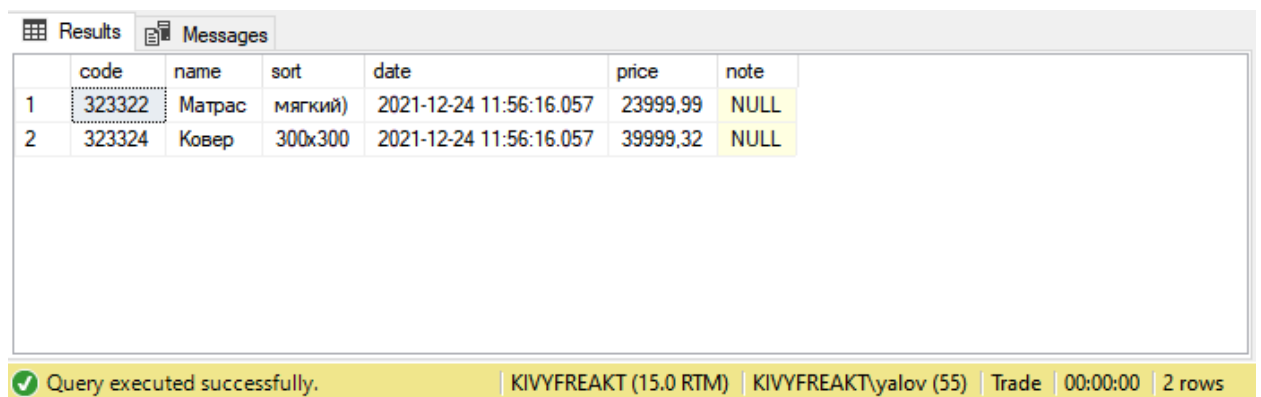
Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 5 rows

Рисунок 16

Создание представления ValuableProducts:

```
CREATE VIEW ValuableProducts AS
SELECT *
FROM products_d
WHERE price > 10000
```

Данное представление выводит продукты, цены которых выше 10000 в отделах магазинов.



	code	name	sort	date	price	note
1	323322	Матрас	мягкий)	2021-12-24 11:56:16.057	23999,99	NULL
2	323324	Ковер	300x300	2021-12-24 11:56:16.057	39999,32	NULL

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 2 rows

Рисунок 17

Создание представления FoodShops:

```
CREATE VIEW FoodShops AS  
SELECT *  
FROM shops  
WHERE class = 'Продукты'
```

Данное представление выводит продуктовые магазины.

Results

Messages

	shop_number	base_number	class	name	address	director_name	note
1	278023	189023	Продукты	Микси	спуск Домодедовская, 17	А. В. Игнатова	NULL
2	283623	189023	Продукты	Переулк	ул. Падыгина, 31	А. И. Смирнов	NULL

Query executed successfully.

KIVYFREAKT (15.0 RTM)

KIVYFREAKT\yakov (55)

Trade

00:00:00

2 rows

Рисунок 18

Хранимые процедуры

Создание хранимой процедуры CloseDepartment. Эта процедура закрывает отдел магазина, при этом переносит товары из этого отдела в открытый. При попытке поставить id, которое отсутствует в таблицах, произойдет исключение.

```
GO
CREATE PROCEDURE CloseDepartment
@department int,
@new_department int,
@shop int
AS
BEGIN

IF (NOT EXISTS (SELECT department_number FROM departments Where shop_number = @shop and
department_number = @department))
THROW 51000, 'Thes department dont exist in the database', 1;

IF (NOT EXISTS (SELECT department_number FROM departments Where shop_number = @shop and
department_number = @new_department and state = 1))
THROW 51000, 'Thes new_department dont exist in the database', 1;

UPDATE departments
SET state = 0
WHERE department_number = @department AND shop_number = @shop

UPDATE products_departments
SET department_number = @new_department
WHERE department_number = @department

END
```

Проверим:

```
EXEC CloseDepartment 1, 3, 238023
```

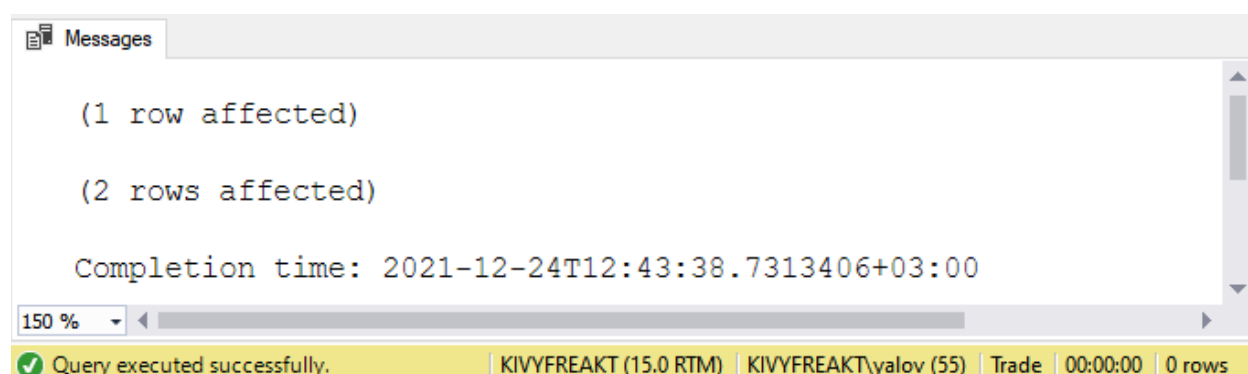


Рисунок 19

	department_number	shop_number	manager_name	state	note
1	1	278023	А. В. Игнатова	0	NULL
2	2	278023	К. П. Овчинников	0	Перестановка
3	3	278023	К. П. Овчинников	1	NULL
4	4	283623	А. И. Смирнов	1	NULL
5	5	285023	Д. П. Балашова	1	NULL
6	6	223423	Д. В. Крючкова	1	NULL


 Query executed successfully.
 | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 6 rows

Рисунок 20

Results				Messages	
	department_number	code	amount		
1	3	111112	1000		
2	3	111113	23423		
3	3	111114	1234234		
4	4	323322	7756		
5	4	323324	45		
6	6	323323	4643		


 Query executed successfully.
 | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (55) | Trade | 00:00:00 | 6 rows

Рисунок 21

Создание хранимой процедуры MoveFromBaseToDepartment. Эта процедура переводит товар из склада в отдел. При этом следует указать цену товара в отделе. При попытке поставить id, которое отсутствует в таблицах, произойдет исключение.

```
GO
CREATE PROCEDURE MoveFromBaseToDepartment
@old_code int,
@new_code int,
@base int,
@department int,
@price int
AS
BEGIN

IF (NOT EXISTS (SELECT code FROM products_b Where code = @old_code))
THROW 51000, 'Thes product dont exist in the database', 1;

INSERT INTO products_d VALUES (
    @new_code,
    (SELECT name FROM products_b WHERE code = @old_code),
    (SELECT sort FROM products_b WHERE code = @old_code),
    GETDATE(),
    @price,
    NULL);

DELETE FROM products_b WHERE code = @old_code;

INSERT INTO products_departments VALUES (
    @department,
    @new_code,
    (SELECT amount FROM products_bases WHERE code = @old_code));

DELETE FROM products_bases WHERE code = @old_code and base_number = @base;

END
```

Проверка:

```
EXEC MoveFromBaseToDepartment 222222, 222222, 183623, 5, 10000
```

Results		Messages					
	code	name	sort	date	price	note	
1	111112	Молоко	Давлеканово	2021-12-24 11:56:16.057	65,34	NULL	
2	111113	Вода	Газированная	2021-12-24 11:56:16.057	45,11	NULL	
3	111114	Хлеб	Бородинский	2021-12-24 11:56:16.057	45,32	NULL	
4	222222	Стол	160 x 90	2021-12-24 13:15:29.280	10000	NULL	
5	323322	Матрас	мягкий)	2021-12-24 11:56:16.057	23999,99	NULL	
6	323323	Наушники	Проводные	2021-12-24 11:56:16.057	3400	NULL	
7	323324	Ковер	300x300	2021-12-24 11:56:16.057	39999,32	NULL	

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yakov (55) | Trade | 00:00:00 | 7 rows

Рисунок 22

Results		Messages			
	code	name	sort	date	note
1	111111	Сырок	Б.У.АПЕХИН	2021-12-24 11:48:20.950	NULL
2	111112	Молоко	Давлеканово	2021-12-24 11:48:20.950	NULL
3	111114	Хлеб	Бородинский	2021-12-24 11:48:20.950	NULL
4	222221	Кровать	160x200	2021-12-24 11:48:20.950	NULL
5	444441	Вода	Газированная	2021-12-24 11:48:20.950	NULL

Query executed successfully. | KIVYFREAKT (15.0 RTM) | KIVYFREAKT\yalov (55) | Trade | 00:00:00 | 5 rows

Рисунок 23

Results		Messages	
	base_number	code	amount
1	123423	444441	4643
2	178023	111114	23423
3	183623	222221	7756
4	189023	111111	1000
5	189023	111112	1234234

Query executed successfully. | KIVYFREAKT (15.0 RTM) | KIVYFREAKT\yalov (55) | Trade | 00:00:00 | 5 rows

Рисунок 24

Results		Messages	
	department_number	code	amount
1	3	111112	1000
2	3	111113	23423
3	3	111114	1234234
4	3	222222	45
5	4	323322	7756
6	4	323324	45
7	6	323323	4643

Query executed successfully. | KIVYFREAKT (15.0 RTM) | KIVYFREAKT\yalov (55) | Trade | 00:00:00 | 7 rows

Рисунок 25

UDF

Создание определяемой пользователем функции, которая будет возвращать информацию о списке всех магазинов:

```
CREATE FUNCTION ListOfShops ()  
RETURNS TABLE  
AS  
RETURN  
SELECT * FROM shops
```

Вызов определяемой пользователем функции:

```
SELECT * FROM ListOfShops();
```

Results

Messages

	shop_number	base_number	class	name	address	director_name	note
1	223423	178023	Двери	Запилил	пл. Будапештская, 99	Д. В. Крючкова	NULL
2	278023	189023	Продукты	Микси	спуск Домодедовская, 17	А. В. Игнатова	NULL
3	283623	189023	Продукты	Переулок	ул. Ладыгина, 31	А. И. Смирнов	NULL
4	285023	178023	Мебель	Елена	пер. Балканская, 49	Д. П. Балашова	NULL
5	289023	185023	Игрушки	Растишка	пл. Гоголя, 95	А. Г. Гуцин	В ремонте

Query executed successfully.

KIVYFREACT (15.0 RTM)

KIVYFREACT\yalov (55)

Trade

00:00:00

5 rows

Рисунок 26

Создание определяемой пользователем функции DateProductsBase, которая будет возвращать список всех товаров, которые поступили на базу в определенное время:

```
CREATE FUNCTION DateProductsBase (@time datetime)
RETURNS TABLE
AS
RETURN
SELECT * FROM products_b WHERE DAY(@time) = DAY(date) and MONTH(@time) = MONTH(date) and
YEAR(@time) = YEAR(date);
```

Вызов хранимой функции:

```
SELECT * FROM DateProductsBase (GETDATE())
```

Results

Messages

	code	name	sort	date	note
1	111111	Сырок	Б.У.АПЕХИН	2021-12-24 11:48:20.950	NULL
2	111112	Молоко	Давлеканово	2021-12-24 11:48:20.950	NULL
3	111114	Хлеб	Бородинский	2021-12-24 11:48:20.950	NULL
4	222221	Кровать	160x200	2021-12-24 11:48:20.950	NULL
5	444441	Вода	Газированная	2021-12-24 11:48:20.950	NULL

✓ Query executed successfully.

KIVYFREACT (15.0 RTM)

KIVYFREACT\yalov (55)

Trade

00:00:00

5 rows

Рисунок 27

Резервное копирование

Для обеспечения целостности и сохранности базы данных на случай возникновения сбоев в системе, хакерских атак, физического повреждения носителя базы данных и случайного изменения (удаления) данных необходимо

разработать стратегию резервного копирования базы данных. В работе были сделаны: полная резервная копия самой базы данных, дифференциальная копия базы данных, которая учитывает изменения с последнего копирования, и журнал транзакций базы данных.

Создание резервной копии базы данных, и результат его выполнения:

```
USE [master];  
  
ALTER DATABASE [Trade] SET RECOVERY FULL;  
BACKUP DATABASE Trade  
TO DISK = 'D:\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup\Trade_full.bak';  
BACKUP DATABASE Trade  
TO DISK = 'D:\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup\Trade_diff.bak'  
WITH DIFFERENTIAL;  
BACKUP LOG Trade  
TO DISK = 'D:\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup\Trade_logs.bak'  
WITH INIT;
```

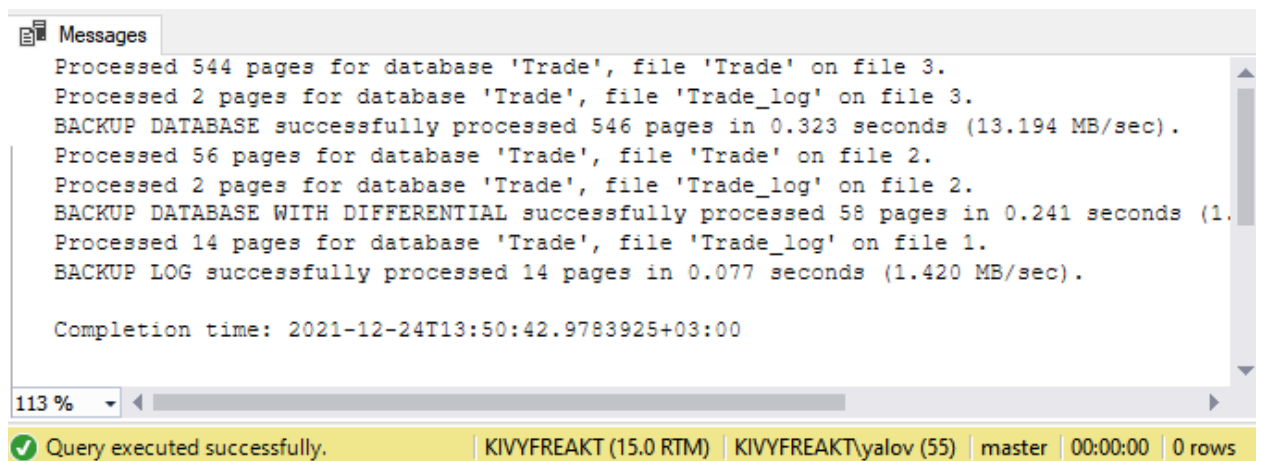


Рисунок 28

Вывод

В результате проделанного индивидуального домашнего задания были приобретены практические навыки проектирования базы данных. Были установлены связи между таблицами, созданы объекты промежуточного слоя, а также установлена стратегия резервного копирования.

Список используемых 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. Базы данных: основы, проектирование, использование: учеб. пособие для вузов по направлению подгот. "Информатика и вычисл. техника" / М.П. Малыхина. - 2-е изд. - СПб. : БХВ-Петербург, 2007. - X, 517 с.
6. Базы данных. Теория и практика: учеб. для вузов по направлениям "Информатика и вычисл. техника" и "Информац. системы" / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. - М. : Высш. шк., 2005. - 463 с.