

Documentation for project:
Trade Purchasing company
Github: NiceOneFox

Содержание

Введение	3
Анализ предметной области.....	4
Схема базы данных.....	9
Код создания таблиц	10
Запросы к базе данных.....	16
Представления:.....	16
Запросы на вывод:	16
Запросы на добавление:.....	17
Запросы на удаление:	18
Запросы на изменение:.....	18
Сложные запросы:.....	19
Хранимые процедуры	21
Клиентское приложение	25
Заключение. Вывод.....	30
Список используемой литературы.....	31

Введение

В рамках курсовой работы планируется изучения предметной области – торгово-закупочная фирма. На основе анализа предметной области необходимо разработать схему базу данных для хранения данных. Используя СУБД, реализовать базу данных и написать необходимые хранимые процедуры и представления для взаимодействия с клиентского приложения и базы данных. На последнем этапе необходимо многократно протестировать приложение.

Анализ предметной области

Пользователи:

Администратор: добавляет новых пользователей базы данных.

Агент-реализатор: организация занимающиеся сбытом товаров со склада.

Менеджер по продажам: занимается коммуникацией с агентами-реализаторами.

Менеджер склада: контролирует поставки и выгрузки товара со склада.

№ пп	Хранимые данные	Пользователи которым разрешён доступ	Ограничения по типу и значению
Товары			
1	Название товара	Менеджер склада, Менеджер	Varchar(80)
2	Код товара	Менеджер склада, Менеджер	Varchar(30)
3	Категория	Менеджер склада, Менеджер	Varchar(40)
4	описание товара	Менеджер склада, Менеджер	Varchar(255)
5	Отпускная цена	Менеджер склада, Менеджер	Money
6	Комиссия агенту	Менеджер склада, Менеджер	float
Товары на складе			
7	Количество на складе	Менеджер склада, Менеджер	int
8	Код товара	Менеджер склада, Менеджер	Varchar(30)
9	Id склада	Менеджер склада, Менеджер	int
10	id	Менеджер склада, Менеджер	int
Склад			
10	Id склада	Менеджер склада, Менеджер	Int
11	Адрес	Менеджер склада, Менеджер	Varchar(120)
12	телефон	Менеджер склада, Менеджер	Varchar(16)
Товары у агента			
13	Количество у агента	Менеджер	Int
14	Код агента	Менеджер	Varchar(30)
15	Код товара у агента	Менеджер	Varchar(30)
6	Id	Менеджер	Int
Агенты-реализаторы			
17	Код агента	Менеджер	Varchar(30)
18	ФИО агента	Менеджер	Varchar(80)
19	адрес	Менеджер	Varchar(100)
20	телефон	Менеджер	Varchar(16)
21	Id аутентификации	Администратор	int
Поступление товара на склад			
22	Дата поступления на склад	Менеджер склада	Datetime2
23	Код поступления	Менеджер склада	int
24	Код склада	Менеджер склада	int

Список поступивших товаров			
25	Id	Менеджер склада	Int
26	Код поступления	Менеджер склада	Int
27	Код товара	Менеджер склада	Varchar(30)
28	Количество	Менеджер склада	int
Сотрудники			
29	Id аутентификации	Администратор	int
30	ФИО сотрудника	Администратор	Varchar(80)
31	телефон	Администратор	Varchar(16)
32	Должность	Администратор	Varchar(50)
33	Адрес	Администратор	Varchar(120)
Заявка на поставку			
34	Код агента	Менеджер, Менеджер склада, Агент	Int
35	Статус заявки	Менеджер, Менеджер склада, Агент	int
36	Код списка товаров заявки	Менеджер, Менеджер склада, Агент	int
37	дата	Менеджер, Менеджер склада, Агент	Datetime2
Статус заявки			
38	Статус айди	Менеджер, Менеджер склада, Агент	int
39	статус	Менеджер, Менеджер склада, Агент	Varchar(120)
Список товаров заявки			
40	Код товара	Менеджер, Менеджер склада, Агент	Varchar(30)
41	Количество	Менеджер, Менеджер склада, Агент	int
42	Код списка товаров заявки	Менеджер, Менеджер склада, Агент	int
43	id	Менеджер, Менеджер склада, Агент	int
Аутентификация			
44	Id аутентификации	Администратор	Int
45	логин	Администратор	Varchar(30)
46	пароль	Администратор	Varchar(30)

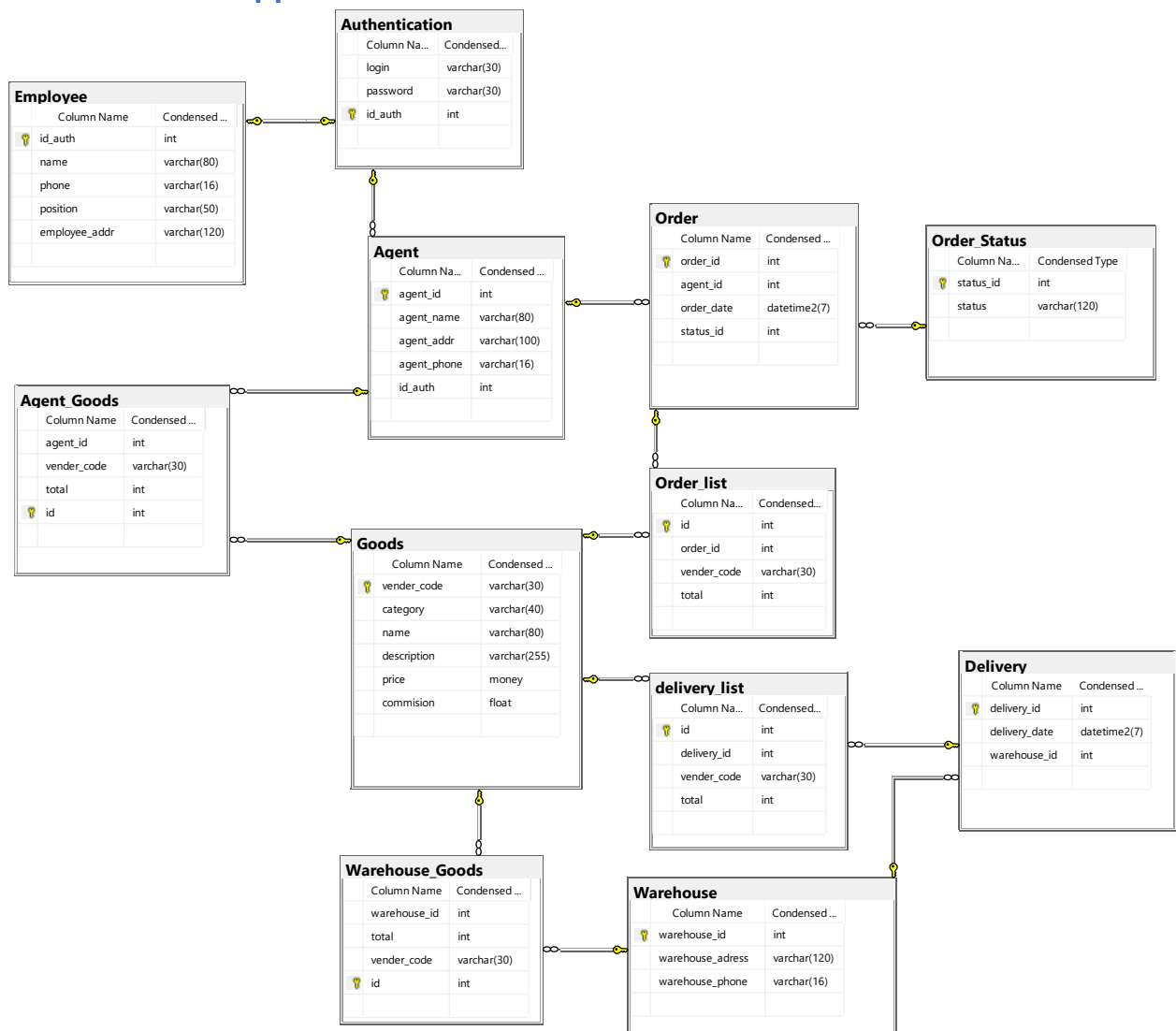
№ пп	Выполняемая функция	Входные данные	Выходные данные	Функции, которые должны быть реализованы в ИС
---------	------------------------	-------------------	--------------------	---

Менеджер склада				
1	Добавление нового товара при поставке	Название товара, количество товара	Изменение БД	Увеличение количества товара на складе при новом поступлении
2	Уменьшение количества товара на складе при отгрузке со склада	Название товара	Изменение БД	Уменьшение количества товара при разгрузке со склада
3	Просмотр всех товаров из выбранной категории	Название категории	Список товаров выбранной категории	Вывод товаров, удовлетворяющих выбранной категории
4	Просмотр информации о поступлении товара на склад	№ пп	Информация о поступлении товара на склад	Возможность узнать информацию о поступлении товара
5	Редактирование заявки на поставку товара	Номер заявки, Список товаров	Изменение БД	Возможность изменить заявку в зависимости наличия товара на складе
6	Изменить статус заявки на поставку товара	Номер заявки, статус	Изменение БД	Возможность изменить текущий статус заявки
6.1	Просмотр всех товаров имеющихся на складе	Warehouse_id	Вывод информации, код товара и его количества	Возможность посмотреть товары имеющиеся на складе
6.2	Удаление товара из списка товаров поставки на склад	Delivery_id Vender_code	Изменение БД	Возможность удалить товаров из списка поступивших
Менеджер				
7	Просмотр количество товара у агента	Код агента, код товара	Количество товара у агента	Возможность узнать количество товара, имеющегося у агента
8	Просмотр количества выданного товара агенту	№ выдачи	Информация о выдаче товара	Вывод информации по номеру выдачи товара агенту

9	Просмотр информации об агенте-реализаторе	Код агента	Информация об агенте-реализаторе	Возможность узнать контактную информацию
10	Просмотр товара на складе	Наименование товара	Информация о товаре	Возможность просмотреть полную информацию о товаре
11	Просмотреть новый отчёт по продажам	Номер отчёта	Вывод информации отчёта. Изменение БД.	Возможность посмотреть статистику продаж товара и выручка
12	Создать новую заявку на поставку	Код агента, Список товаров, Дата, статус	Изменение БД	Возможность создать новую заявку на поставку товара агенту-реализатору
13	Просмотреть статус заявки на поставку	Номер заявки	Вывод информации о заявке	Возможность узнать информацию о заявке на поставку
13.1	Посчитать стоимость всех товаров в заявки на поставку	Order_id	Вывод стоимости всех товаров	Возможность узнать стоимость всех товаров
13.2	Удаление товара из списка заявки на поставку товаров	Order_list, Vender_code	Изменение БД	Возможность удалить товар из списка заявки на поставку
Агент-реализатор				
14	Просмотр заявок на поставку определенному агенту	Код агента	Информация о выдаче товаров агенту	Показ информации списка заявок их статуса
15	Поставка товара на склад агента-реализатора	Код агента, код товара	Изменение БД	Возможность изменить количество имеющегося товара у агента-реализатора
16	Создать новую заявку на поставку	Код агента, Список товаров, Дата, статус	Изменение БД	Возможность создать новую заявку на поставку товара агенту-реализатору

17	Продажа товара, имеющегося у агента-реализатора	Код агента, код товара	Изменение БД	Возможность изменить количество имеющегося товара у агента-реализатора на складе
18	Просмотреть статус заявки	Номер заявки	Информация о заявке на поставку	Возможность посмотреть статус заявки на поставку товара
Администратор				
19	Добавить нового агента-реализатора	ФИО агента, адрес, телефон	Изменение БД	Возможность добавить нового агента реализатора
20	Добавить нового Менеджера склада	ФИО, телефон, должность	Изменение БД	Возможность добавить нового менеджера склада
21	Добавить нового Менеджера	ФИО, телефон, должность	Изменение БД	Возможность добавить нового менеджера

Схема базы данных



Описание схемы:

Warehouse(склад) содержит адрес и контактный телефон

Warehouse_Goods таблица товаров, имеющих на складе.

Goods таблица товаров содержит информацию о товаре.

Delivery (поставка товаров на склад) содержит id списка товаров (goods_id) время доставки и айди склада.

Delivery_list (список товаров поступивших на склад) содержит артикул товара его и его количество и айди доставки для связи с Delivery.

Oder(заявка на поставку товара) содержит код агента, код списка товаров, дату и айди статус заказа.

Order_list содержит артикул товара, его количество и айди заказа для связи с таблицей Order.

Order_status содержит описание состояние заказа и столбец указывающий на завершённость заказа.

Agent(агент-реализатор) содержит контактную информацию об агенте и id для аутентификации в бд.

Agent_Goods(список товаров у агента). Содержит код агента, код товара и его количество.

Employee(сотрудник) содержит контактную информацию о сотруднике и его должность, а так же id для аутентификации в бд.

Authentication содержит логин и пароль для использования БД.

Таблица хранящая статусы заказов:

Status_id	status
1	‘Заказ создан’
2	‘Заказ обрабатывается’
3	‘Заказ отправлен’
4	‘Заказ доставлен’
5	‘Заказ отменён’
6	‘Заказ завершён’

Код создания таблиц

```
1. CREATE TABLE [dbo].[Agent](
    [agent_id] [int] IDENTITY(1,1) NOT NULL,
    [agent_name] [varchar](80) NOT NULL,
    [agent_addr] [varchar](100) NOT NULL,
    [agent_phone] [varchar](16) NULL,
    [id_auth] [int] NOT NULL,
    CONSTRAINT [PK_Agent] PRIMARY KEY CLUSTERED
(
    [agent_id] 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],
    CONSTRAINT [IX_Agent] UNIQUE NONCLUSTERED
(
    [id_auth] 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].[Agent] WITH CHECK ADD CONSTRAINT [FK_Agent_Authentication] FOREIGN
KEY([id_auth])
REFERENCES [dbo].[Authentication] ([id_auth])
GO

ALTER TABLE [dbo].[Agent] CHECK CONSTRAINT [FK_Agent_Authentication]
GO
```

2.

```
CREATE TABLE [dbo].[Agent_Goods](
    [agent_id] [int] NOT NULL,
    [vender_code] [varchar](30) NOT NULL,
    [total] [int] NOT NULL,
    [id] [int] IDENTITY(1,1) NOT NULL,
    CONSTRAINT [PK_Товары у агента] PRIMARY KEY CLUSTERED
(
    [id] 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].[Agent_Goods] WITH CHECK ADD CONSTRAINT [FK_Agent_Goods_Agent]
FOREIGN KEY([agent_id])
REFERENCES [dbo].[Agent] ([agent_id])
GO

ALTER TABLE [dbo].[Agent_Goods] CHECK CONSTRAINT [FK_Agent_Goods_Agent]
GO

ALTER TABLE [dbo].[Agent_Goods] WITH CHECK ADD CONSTRAINT [FK_Agent_Goods_Goods]
FOREIGN KEY([vender_code])
REFERENCES [dbo].[Goods] ([vender_code])
GO

ALTER TABLE [dbo].[Agent_Goods] CHECK CONSTRAINT [FK_Agent_Goods_Goods]
GO

ALTER TABLE [dbo].[Agent_Goods] WITH CHECK ADD CONSTRAINT [agent_goods_total] CHECK
(([total]>=(0)))
GO

ALTER TABLE [dbo].[Agent_Goods] CHECK CONSTRAINT [agent_goods_total]
GO
```

3. CREATE TABLE [dbo].[Authentication](

```
[login] [varchar](30) NOT NULL,
[password] [nchar](64) NOT NULL,
[id_auth] [int] IDENTITY(1,1) NOT NULL,
CONSTRAINT [PK_Authentication] PRIMARY KEY CLUSTERED
(
    [id_auth] 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],
CONSTRAINT [IX_Authentication] UNIQUE NONCLUSTERED
(
    [login] 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
```

4. CREATE TABLE [dbo].[Employee](

```
[id_auth] [int] NOT NULL,
[name] [varchar](80) NOT NULL,
```

```

        [phone] [varchar](16) NULL,
        [position] [varchar](50) NULL,
        [employee_addr] [varchar](120) NULL,
    CONSTRAINT [IX_Employee] UNIQUE NONCLUSTERED
    (
        [id_auth] 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].[Employee] WITH CHECK ADD CONSTRAINT [FK_Employee_Authentication]
FOREIGN KEY([id_auth])
REFERENCES [dbo].[Authentication] ([id_auth])
GO

```

```

ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK_Employee_Authentication]
GO

```

```

5. CREATE TABLE [dbo].[Goods](
    [vender_code] [varchar](30) NOT NULL,
    [category] [varchar](40) NULL,
    [name] [varchar](80) NOT NULL,
    [description] [varchar](255) NULL,
    [price] [money] NOT NULL,
    [commission] [float] NOT NULL,
    CONSTRAINT [PK_Товары] PRIMARY KEY CLUSTERED
    (
        [vender_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].[Goods] WITH CHECK ADD CONSTRAINT [Goods_commission] CHECK
(((commission)>=(0)))
GO

```

```

ALTER TABLE [dbo].[Goods] CHECK CONSTRAINT [Goods_commission]
GO

```

```

ALTER TABLE [dbo].[Goods] WITH CHECK ADD CONSTRAINT [Goods_price] CHECK
(((price)>=(0)))
GO

```

```

ALTER TABLE [dbo].[Goods] CHECK CONSTRAINT [Goods_price]
GO

```

```

6. CREATE TABLE [dbo].[Goods_List](
    [goods_id] [int] NOT NULL,
    [vender_code] [varchar](30) NOT NULL,
    [total] [int] NOT NULL,
    [id] [int] NOT NULL,
    CONSTRAINT [PK_Список товаров] PRIMARY KEY CLUSTERED
    (
        [id] 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
```

```
7. CREATE TABLE [dbo].[Order](
    [order_id] [int] IDENTITY(1,1) NOT NULL,
    [agent_id] [int] NOT NULL,
    [order_date] [datetime2](7) NOT NULL,
    [status_id] [int] NOT NULL,
    CONSTRAINT [PK_Заявка на поставку товара] PRIMARY KEY CLUSTERED
(
    [order_id] 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].[Order] ADD CONSTRAINT [DF_Order_order_date] DEFAULT (getdate()) FOR
[order_date]
GO
```

```
ALTER TABLE [dbo].[Order] ADD CONSTRAINT [DF_Order_status_id] DEFAULT ((0)) FOR
[status_id]
GO
```

```
ALTER TABLE [dbo].[Order] WITH CHECK ADD CONSTRAINT [FK_Order_Agent1] FOREIGN
KEY([agent_id])
REFERENCES [dbo].[Agent] ([agent_id])
GO
```

```
ALTER TABLE [dbo].[Order] CHECK CONSTRAINT [FK_Order_Agent1]
GO
```

```
ALTER TABLE [dbo].[Order] WITH CHECK ADD CONSTRAINT [FK_Order_Order_Status] FOREIGN
KEY([status_id])
REFERENCES [dbo].[Order_Status] ([status_id])
GO
```

```
ALTER TABLE [dbo].[Order] CHECK CONSTRAINT [FK_Order_Order_Status]
GO
```

```
8. CREATE TABLE [dbo].[Order_list](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [order_id] [int] NOT NULL,
    [vender_code] [varchar](30) NOT NULL,
    [total] [int] NOT NULL,
    CONSTRAINT [PK_Order_list] PRIMARY KEY CLUSTERED
(
    [id] 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].[Order_list] WITH CHECK ADD CONSTRAINT [FK_Order_list_Goods] FOREIGN
KEY([vender_code])
REFERENCES [dbo].[Goods] ([vender_code])
GO
```

```
ALTER TABLE [dbo].[Order_list] CHECK CONSTRAINT [FK_Order_list_Goods]
GO
```

```
ALTER TABLE [dbo].[Order_list] WITH CHECK ADD CONSTRAINT [FK_Order_list_Order] FOREIGN
KEY([order_id])
REFERENCES [dbo].[Order] ([order_id])
GO
```

```
ALTER TABLE [dbo].[Order_list] CHECK CONSTRAINT [FK_Order_list_Order]
GO
```

```
ALTER TABLE [dbo].[Order_list] WITH CHECK ADD CONSTRAINT [Order_list_total] CHECK
(((total]>=(0)))
GO
```

```
ALTER TABLE [dbo].[Order_list] CHECK CONSTRAINT [Order_list_total]
GO
```

```
9. CREATE TABLE [dbo].[Order_Status](
    [status_id] [int] IDENTITY(1,1) NOT NULL,
    [status] [varchar](120) NULL,
    CONSTRAINT [PK_Order_Status] PRIMARY KEY CLUSTERED
(
    [status_id] 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
```

```
10. CREATE TABLE [dbo].[Warehouse](
    [warehouse_id] [int] NOT NULL,
    [warehouse_addr] [varchar](120) NULL,
    [warehouse_phone] [varchar](16) NOT NULL,
    CONSTRAINT [PK_Склад] PRIMARY KEY CLUSTERED
(
    [warehouse_id] 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
```

```
11. CREATE TABLE [dbo].[Warehouse_Goods](
    [warehouse_id] [int] NOT NULL,
    [total] [int] NOT NULL,
    [vender_code] [varchar](30) NOT NULL,
    [id] [int] IDENTITY(1,1) NOT NULL,
    CONSTRAINT [PK_Warehouse_Goods_1] PRIMARY KEY CLUSTERED
(
    [id] 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].[Warehouse_Goods] WITH CHECK ADD CONSTRAINT  
[FK_Warehouse_Goods_Goods] FOREIGN KEY([vender_code])  
REFERENCES [dbo].[Goods] ([vender_code])  
GO
```

```
ALTER TABLE [dbo].[Warehouse_Goods] CHECK CONSTRAINT [FK_Warehouse_Goods_Goods]  
GO
```

```
ALTER TABLE [dbo].[Warehouse_Goods] WITH CHECK ADD CONSTRAINT  
[FK_Warehouse_Goods_Warehouse] FOREIGN KEY([warehouse_id])  
REFERENCES [dbo].[Warehouse] ([warehouse_id])  
GO
```

```
ALTER TABLE [dbo].[Warehouse_Goods] CHECK CONSTRAINT [FK_Warehouse_Goods_Warehouse]  
GO
```

```
ALTER TABLE [dbo].[Warehouse_Goods] WITH CHECK ADD CONSTRAINT [Warehouse_goods_total]  
CHECK ((([total]>=(0))))  
GO
```

```
ALTER TABLE [dbo].[Warehouse_Goods] CHECK CONSTRAINT [Warehouse_goods_total]  
GO
```

Запросы к базе данных

Представления:

1. Создаёт представление из таблицы warehouse_goods, которое выводит полную информацию о товаре по vender_code

```
CREATE VIEW warehouse_goods_info_view3 AS
SELECT w.vender_code, total, g.name, g.category, g.description, g.price, g.commission
FROM Warehouse_Goods AS w
LEFT JOIN Goods AS g ON w.vender_code = g.vender_code
```

2. Создаёт представление из таблицы Order, которое выводит информацию о статусе заказа по номеру заявки

```
CREATE VIEW Order_status_view AS
SELECT d.agent_id, d.order_date, s.status
FROM [Order] d
LEFT JOIN Order_Status s ON d.status_id = s.status_id
```

3. Создать представление, где видно стоимость товара и его количество в списке поступивших товаров на склад.

```
CREATE VIEW delivery_goods_view AS
SELECT t.vender_code, g.name, g.price, t.total, d.delivery_id
FROM Delivery d
LEFT JOIN delivery_list t ON d.delivery_id = t.delivery_id
INNER JOIN Goods g ON t.vender_code = g.vender_code;
```

4. Вывод из таблицы(goods) все товары из выбранной категории.

```
CREATE VIEW goods_category_view AS
SELECT *
FROM Goods
WHERE category = 'furniture';
```

Запросы на вывод:

5. Вывод из таблицы Delivery которое выводит всю информацию о доставке по определённому id

```
SELECT *
FROM delivery
WHERE delivery_id == 2;
```

6. Вывод из таблицы Agent, которое выводит контактную информацию об агенте

```
SELECT agent_name, agent_addr, agent_phone
FROM Agent
WHERE agent_name = 'Ivanov Ivan Ivanovich';
```

7. Вывод всех товаров чья стоимость больше N.

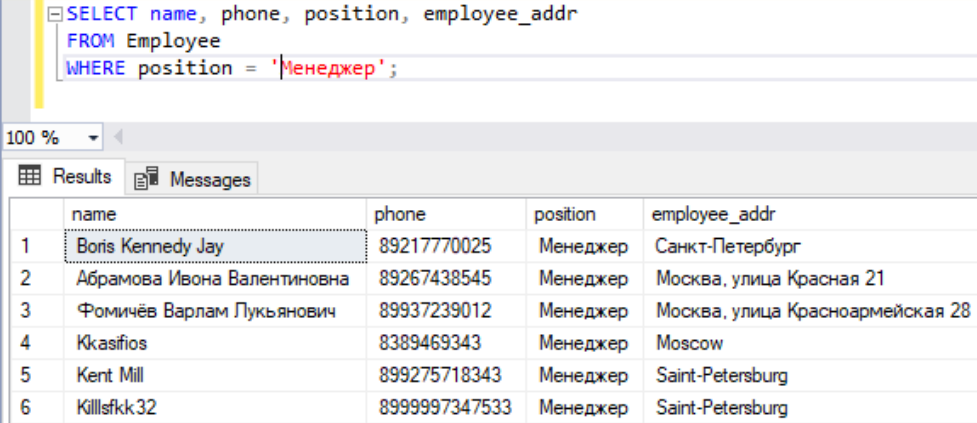
```
SELECT *
FROM Goods
WHERE price >= 500.50;
```


8. Вывод всех сотрудников чья должность “Менеджер”

```
SELECT name, phone, position, employee_addr
```

```
FROM Employee
```

```
WHERE position = 'Менеджер';
```



The screenshot shows a database query window with the following SQL query:

```
SELECT name, phone, position, employee_addr
FROM Employee
WHERE position = 'Менеджер';
```

Below the query window, the results are displayed in a table with 5 columns: name, phone, position, and employee_addr. The table contains 6 rows of data.

	name	phone	position	employee_addr
1	Boris Kennedy Jay	89217770025	Менеджер	Санкт-Петербург
2	Абрамова Илона Валентиновна	89267438545	Менеджер	Москва, улица Красная 21
3	Фомичёв Варлам Лукьянович	89937239012	Менеджер	Москва, улица Красноармейская 28
4	Kkasifios	8389469343	Менеджер	Moscow
5	Kent Mill	899275718343	Менеджер	Saint-Petersburg
6	Killsfkk32	8999997347533	Менеджер	Saint-Petersburg

9. Вывод адресов всех складов

```
SELECT warehouse_id, warehouse_addr
```

```
FROM Warehouse
```

Запросы на добавление:

10. Создать новый Order_list введя список товаров

```
INSERT INTO Order_list (order_id, vender_code, total)
```

```
VALUES (1, 'AK69520', 10);
```

11. Создать новый Order введя полную информацию о заявке

```
INSERT INTO Order(agent_id, status_id)
```

```
VALUES (8, DEFAULT);
```

```
// default value equals 0
```

12. Создать новый Delivery_list введя список товаров

```
INSERT INTO Delivery_list (delivery_id, vender_code, total)
```

```
VALUES (12, 'LD56919', 10);
```

13. Создать новый Delivery введя полную информацию о заявке

```
INSERT INTO Delivery (delivery_date, warehouse_id)
```

```
VALUES (DEFAULT, 4);
```

```
//default date is current datetime
```

14. Добавить нового агента-реализатора введя полную информацию о нём

```
INSERT INTO Agent(agent_name, agent_phone, agent_addr)
```

VALUES ('Josh Petrov Ivanovich', '8999002367', 'ulitsa Lenina 25, Saint-Petersburg');

15. Добавить нового менеджера склада в таблицу employee введя полную информацию

INSERT INTO Employee(name, phone, position, employee_addr)

VALUES ('Ivan Sidorov Boris', '+7921778901', 'Менеджер', 'Prospekt nevsky 45, Saint-Petersburg');

16. Добавить нового менеджера в таблицу employee введя полную информацию

INSERT INTO Employee(name, phone, position, employee_addr)

VALUES ('Boris Sidorov Ivanovich', '+7921778901', 'Менеджер склада', 'Prospekt nevsky 45, Saint-Petersburg');

16.1 Добавить нового пользователя БД

INSERT INTO Authentication(login, password)

VALUES('Admin', '12345');

Запросы на удаление:

17. Удаление товара из Delivery_list

DELETE FROM Delivery_list

WHERE delivery_id = 3 AND vender_code = 'AK69520';

18. Удаление товара из Order_list

DELETE FROM Order_list

WHERE Order_id = 5 AND vender_code = 'LV32419';

19. Удалить сотрудника из Employee

DELETE FROM Employee

WHERE Employee.name = 'Boris Petrov Petrovich';

Запросы на изменение:

20. Уменьшить кол-во товара на складе у агента-реализатора

UPDATE Agent_goods

SET total = total - 25;

WHERE agent_id = 2 AND vender_code = 'LA77220';

21. Увеличить кол-во товара на складе у агента-реализатора

UPDATE Agent_goods

SET total = total + 10;

WHERE agent_id = 2 AND vender_code = 'LA77220';

22. Уменьшить кол-во товара в Warehouse_goods

```
UPDATE Warehouse_Goods
```

```
SET total = total - 30
```

```
WHERE warehouse_id = 3 AND vender_code 'DK44319'
```

23. Увеличить кол-во товара в Warehouse_goods

```
UPDATE Warehouse_Goods
```

```
SET total = total + 15
```

```
WHERE warehouse_id = 3 AND vender_code 'DK44319'
```

24. Изменить статус заявки на поставку товара

```
UPDATE Order
```

```
Set status_id = 3;
```

```
WHERE order_id = 7;
```

Сложные запросы:

25. Просмотр название, код товара и его количество по таблице Order определённому агенту

```
SELECT g.name, g.vender_code, t.total, d.agent_id, d.order_id
FROM [Order] d
LEFT JOIN Order_list t ON t.order_id = d.order_id
INNER JOIN Goods g ON g.vender_code = t.vender_code
WHERE d.agent_id = 7
```

26. Просмотр код товара и его количество по таблице Order чьи заявки являются выполненными за временной период.

```
SELECT t.vender_code, t.total, d.order_id, d.order_date, d.agent_id
FROM [Order] d
INNER JOIN Order_list t ON t.order_id = d.order_id
WHERE d.status_id = 5
AND d.order_date BETWEEN '2020/08/25' and '2020/10/11';
```

27. Просмотр товара с наибольшим количеством продаж по таблице Order со статусом выполненные

```
SELECT g.name, g.price, t.total
FROM [Order] d, Goods g, Order_list t
WHERE d.status_id = 5
ORDER BY t.total;
```

28. Посмотреть все заявки на поставку которые находятся в обработке

```
SELECT g.name, g.category, d.order_id, d.order_date, d.agent_id
FROM [Order] d
LEFT JOIN Order_list t ON t.order_id = d.order_id
INNER JOIN Goods g ON g.vender_code = t.vender_code
WHERE d.status_id = 1 OR d.status_id = 2;
```

29. Вывести список товаров, имеющихся на всех складах и номер склада

```
SELECT g.name, wg.vender_code, wg.total, w.warehouse_id, w.warehouse_addr
FROM Warehouse_Goods wg
INNER JOIN Goods g ON g.vender_code = wg.vender_code
LEFT JOIN Warehouse w ON w.warehouse_id = wg.warehouse_id
```

```
WHERE wg.total > 0
ORDER BY w.warehouse_id
```

30. Вывести все товары определённого агента из определённой категории

```
SELECT g.name, g.category, g.price, ag.total, g.commission, g.description
FROM Agent_Goods ag
INNER JOIN Goods g ON g.vender_code = ag.vender_code
LEFT JOIN Agent a ON a.agent_id = ag.agent_id
WHERE ag.agent_id = 7
AND g.category = 'Мебель'
```

31. Вывести полную информацию о товаре в порядке возрастания кол-ва на определённом складе (узнать какой товар закончился либо заканчивается)

```
SELECT g.name, wg.vender_code, wg.total, w.warehouse_addr, w.warehouse_phone
FROM Warehouse w
LEFT JOIN Warehouse_Goods wg ON wg.warehouse_id = w.warehouse_id
INNER JOIN Goods g ON g.vender_code = wg.vender_code
ORDER BY wg.total;
```

32. Вывести все склады, где имеется в наличие определённый товар

```
SELECT g.name, g.price, g.commission, wg.total, w.warehouse_addr, w.warehouse_phone
FROM Warehouse w
LEFT JOIN Warehouse_Goods wg ON wg.warehouse_id = w.warehouse_id
INNER JOIN Goods g ON g.vender_code = wg.vender_code
WHERE wg.vender_code = 'SD12319' AND wg.total > 0;
```

33. Вывести полную информацию о товарах, имеющихся у определённого агента + кол-во

```
SELECT a.agent_name, g.name AS good_name, g.category, g.description, g.price,
g.commission, ag.total, ag.agent_id
FROM Goods g
INNER JOIN Agent_Goods ag ON g.vender_code = ag.vender_code
INNER JOIN Agent a ON a.agent_id = ag.agent_id
WHERE a.agent_name = 'Мясников Виссарион Филатович';
```

34. Удалить все order заказы которых были отменены

```
DELETE t
FROM [Order_list] t
INNER JOIN [Order] d ON d.order_id = t.order_id
WHERE d.order_id = 6;
```

35. Вывести полную информацию о товаре и его количество по delivery id

```
SELECT g.name, g.description, g.price, g.commission, d.delivery_id, d.delivery_date
FROM Delivery d
LEFT JOIN delivery_list t ON t.delivery_id = d.delivery_id
INNER JOIN Goods g ON g.vender_code = t.vender_code
WHERE d.delivery_id = 2;
```

Хранимые процедуры

```
ALTER PROCEDURE [dbo].[Add_Agent]
@agent_name AS varchar(80),
@agent_phone AS varchar(16),
@agent_addr AS varchar(100),
@id_auth AS int AS
INSERT INTO Agent(agent_name, agent_phone, agent_addr, id_auth)
VALUES (@agent_name, @agent_phone, @agent_addr, @id_auth);
```

```
ALTER PROCEDURE [dbo].[Add_auth]
@login varchar(30),
@password nchar(64) AS
INSERT INTO Authentication(login, password)
VALUES (@login, @password);
```

```
ALTER PROCEDURE [dbo].[Add_Delivery]
@warehouse_id AS INT AS
INSERT INTO Delivery (delivery_date, warehouse_id)
VALUES (DEFAULT, @warehouse_id);
```

```
ALTER PROCEDURE [dbo].[Add_Delivery_list]
@delivery_id AS INT,
@vender_code AS varchar(30),
@total AS INT AS
INSERT INTO Delivery_list(delivery_id, vender_code, total)
VALUES (@delivery_id, @vender_code, @total);
```

```
ALTER PROCEDURE [dbo].[Add_Employee]
@name AS varchar(80),
@phone AS varchar(16),
@position AS varchar(100),
@addr AS varchar(150),
@id_auth AS int AS
INSERT INTO Employee([name], phone, position, employee_addr, id_auth)
VALUES (@name, @phone, @position, @addr, @id_auth);
```

```
ALTER PROCEDURE [dbo].[Add_Order]
@agent_id AS INT,
@status_id AS INT AS
INSERT INTO [Order](agent_id, status_id)
VALUES (@agent_id, @status_id);
```

```
ALTER PROCEDURE [dbo].[Add_Order_List]
@order_id AS INT,
@vender_code AS varchar(30),
@total AS INT AS
INSERT INTO Order_list (order_id, vender_code, total)
VALUES (@order_id, @vender_code, @total);
```

```
ALTER PROCEDURE [dbo].[Agent_goods_in_stock]
@agent_name AS varchar(80) AS
SELECT a.agent_name, g.name AS good_name, g.category, g.description, g.price,
g.commission, ag.total, ag.agent_id
```

```
FROM Goods g
INNER JOIN Agent_Goods ag ON g.vender_code = ag.vender_code
INNER JOIN Agent a ON a.agent_id = ag.agent_id
WHERE a.agent_name = @agent_name;
```

```
ALTER PROCEDURE [dbo].[Agent_info] @agent_name AS varchar(80) AS
SELECT agent_name, agent_addr, agent_phone
FROM Agent
WHERE agent_name = @agent_name;
```

```
ALTER PROCEDURE [dbo].[Change_order_status]
    @order_id int,
    @status_id int
AS
    UPDATE [Order]
    SET status_id = @status_id
    WHERE order_id = @order_id
```

```
ALTER PROCEDURE [dbo].[Delete_good_from_delivery]
    @delivery_id AS INT,
    @vender_code AS varchar(30) AS
DELETE FROM delivery_list
WHERE delivery_id = @delivery_id AND vender_code = @vender_code
```

```
ALTER PROCEDURE [dbo].[Delivery_Goods_info]
    @delivery_id as INT AS
SELECT g.name, g.description, g.price, g.commission, d.delivery_id, d.delivery_date
FROM Delivery d
LEFT JOIN delivery_list t ON t.delivery_id = d.delivery_id
INNER JOIN Goods g On g.vender_code = t.vender_code
WHERE d.delivery_id = @delivery_id;
```

```
ALTER PROCEDURE [dbo].[Done_orders]
    @left as datetime,
    @right as datetime AS
SELECT t.vender_code, t.total, d.order_id, d.order_date, d.agent_id
FROM [Order] d
INNER JOIN Order_list t ON t.order_id = d.order_id
WHERE d.status_id = 5
AND d.order_date BETWEEN @left and @right;
```

```
ALTER PROCEDURE [dbo].[Employee_position] @position AS varchar(50) AS
SELECT name, phone, position, employee_addr
FROM Employee
WHERE position = @position;
```

```
ALTER PROCEDURE [dbo].[Get_last_id_auth]
    @id as INT OUT AS
BEGIN
SET @id = (SELECT TOP 1 id_auth FROM Authentication
ORDER BY id_auth DESC)
```

```
RETURN @id
END
```

```
ALTER PROCEDURE [dbo].[Get_last_id_delivery]
@id as INT OUT AS
BEGIN
SET @id = (SELECT TOP 1 delivery_id FROM Delivery
ORDER BY delivery_id DESC)
RETURN @id
END
```

```
ALTER PROCEDURE [dbo].[Get_last_id_order]
@idReturn as INT OUT AS
BEGIN
SET @idReturn = (SELECT TOP 1 order_id FROM [Order]
ORDER BY order_id DESC)
RETURN @idReturn
END
```

```
ALTER PROCEDURE [dbo].[Get_user_password]
@login varchar(30) AS
SELECT password
FROM [Authentication]
WHERE login = @login
```

```
ALTER PROCEDURE [dbo].[Good_price_more] @price_edge AS FLOAT AS
SELECT *
FROM Goods
WHERE price >= @price_edge;
```

```
ALTER PROCEDURE [dbo].[Goods_Agent]
@agent_id as INT AS
SELECT g.name, g.vender_code, t.total, d.agent_id, d.order_id
FROM [Order] d
LEFT JOIN Order_list t ON t.order_id = d.order_id
INNER JOIN Goods g ON g.vender_code = t.vender_code
WHERE d.agent_id = @agent_id
```

```
ALTER PROCEDURE [dbo].[Goods_agent_category]
@agent_id as INT,
@category as varchar(40) AS
SELECT g.name, g.category, g.price, ag.total, g.commission, g.description
FROM Agent_Goods ag
INNER JOIN Goods g ON g.vender_code = ag.vender_code
LEFT JOIN Agent a ON a.agent_id = ag.agent_id
WHERE ag.agent_id = @agent_id
AND g.category = @category
```

```
ALTER PROCEDURE [dbo].[Warehouse_Goods_in_stock]
@vender_code as varchar(30) AS
SELECT g.name, g.price, g.commission, wg.total, w.warehouse_addr, w.warehouse_phone
FROM Warehouse w
LEFT JOIN Warehouse_Goods wg ON wg.warehouse_id = w.warehouse_id
INNER JOIN Goods g ON g.vender_code = wg.vender_code
WHERE wg.vender_code = @vender_code AND wg.total > 0;
```

Триггеры

```
CREATE TRIGGER After_update_Goods_price ON Goods
FOR UPDATE AS
IF UPDATE (price)
BEGIN

    IF EXISTS(
        SELECT l.vender_code
        FROM Order_list l
        WHERE l.vender_code = ( SELECT del.vender_code FROM deleted del))
    BEGIN
        PRINT 'Нельзя изменить цену товара, если он находится в заказе'
        ROLLBACK TRANSACTION
    END
    ELSE
    BEGIN
        PRINT 'Цена изменена'
    END
END

DROP TRIGGER After_update_Goods_price
```

```
UPDATE Goods
SET price = 13000.5
WHERE name = 'Стул'
```

```
CREATE TRIGGER After_update_Goods On Goods
AFTER INSERT, UPDATE
AS
BEGIN
UPDATE Goods
SET category = 'Клавиатура компьютерная'
WHERE category = 'Клавиатура'
END;

INSERT Goods(vender_code, category, name, description, price, commission)
VALUES('BK18319', 'Клавиатура', 'Logitech K120', 'материал: пластик', 1200, 3)
```

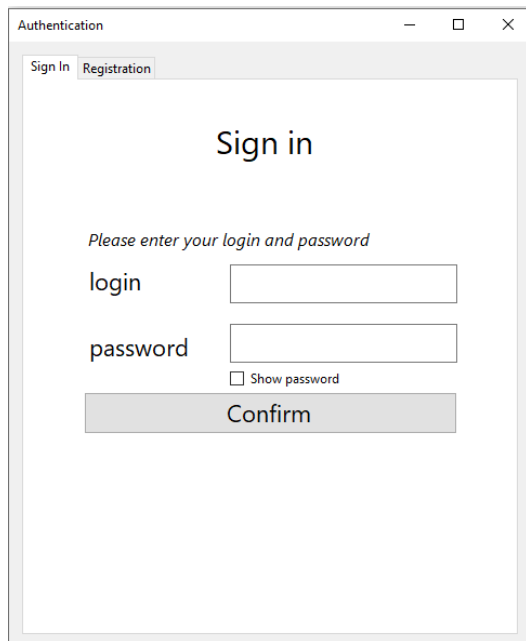
```
CREATE TRIGGER After_delete_order_list ON Order_list
AFTER DELETE
AS
BEGIN
INSERT INTO History_order_list
SELECT del.*
FROM deleted del
END;

CREATE TRIGGER After_delete_history_order_list On History_order_list
After DELETE
AS
BEGIN
INSERT INTO Order_list
SELECT del.*
FROM deleted del
END;
```


Клиентское приложение

Начальная форма приложения для аутентификации пользователей.

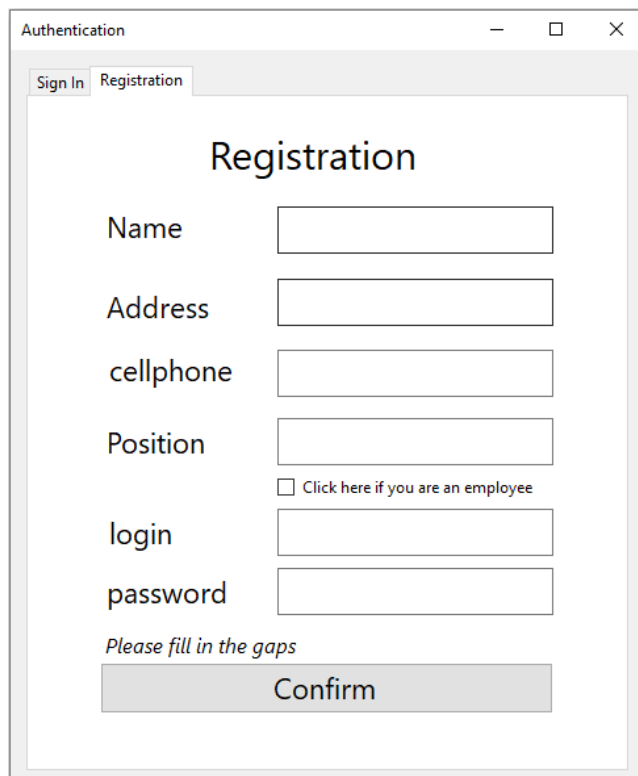
Первая вкладка отвечает за вход пользователей в систему



The screenshot shows a window titled "Authentication" with two tabs: "Sign In" (active) and "Registration". The "Sign in" tab contains the following elements:

- Header: "Sign in"
- Instruction: "Please enter your login and password"
- Form fields: "login" and "password" (both empty text boxes)
- Checkbox: "Show password" (unchecked)
- Button: "Confirm" (grey)

Вторая вкладка отвечает за регистрацию новых пользователей



The screenshot shows the same "Authentication" window, but with the "Registration" tab selected. The "Sign In" tab is now disabled. The "Registration" tab contains the following elements:

- Header: "Registration"
- Form fields: "Name", "Address", "cellphone", "Position", "login", and "password" (all empty text boxes)
- Checkbox: "Click here if you are an employee" (unchecked)
- Instruction: "Please fill in the gaps"
- Button: "Confirm" (grey)

После ввода данных и нажатия на кнопку аутентификации в базе

Пользователю предоставляется доступ к приложению и работе с данными БД

Authentication

Sign In Registration

Sign in

Please enter your login and password

login

password

☐ Show password

Confirm

Program

Update

	id	order_id	vender_code	total	Delete
▶	2	5	LB87218	2	Delete
	3	6	SD33117	5	Delete
	4	5	SD12319	4	Delete
	5	6	AL23519	8	Delete
	6	7	SD33117	6	Delete
	7	7	AL12420	10	Delete
	8	7	AL23519	4	Delete
	9	8	ME82319	5	Delete
	12	8	LB12415	12	Delete
	13	9	ME89120	12	Delete
	14	9	LB87218	12	Delete
	17	10	ME82319	14	Delete
	22	6	SD33117	5	Delete
	23	7	LB87218	3	Delete
	24	8	SD33117	2	Delete
	27	21	ME82319	3	Delete
	28	21	AL23519	6	Delete
	29	21	BK18319	6	Delete
	31	22	ME89120	5	Delete
	32	23	AL23519	5	Delete

Create new order Create new delivery

Order goods list

Warehouse goods in stock

Employee position

Agent info

Delivery list of goods

Agent goods

Change order status

Order_list Delivery list Agent goods Warehouse Warehouse goods Warehouse goods info Orders info

Order Delivery Agents Employee Goods Delivery goods info

Пользователь имеет возможность в приложение удалять, добавлять и обновлять данные таблиц, размещённых в нижней части (кроме таблиц Agents и Employee)

Update				
	warehouse_id	warehouse_addr	warehouse_phone	Delete
	1	Санкт-Петербург, Улиц...	89917345010	Delete
	2	Санкт-Петербург	87812358102	Delete
	3	Москва, Улица Красноа...	81635894131	Delete
	4	Москва, Улица Победы ...	80091923512	Delete
	5	Рязань, Пятая Линия, д 48	89723593054	Insert

После нажатия кнопки insert, данные добавляются в базу.

Демонстрация удаления строк таблицы

5	Рязань, Пятая Линия, д 48	89723593054	Delete

Удаление

Удалить эту строку?

[Да](#) [Нет](#)

Update				
	warehouse_id	warehouse_addr	warehouse_phone	Delete
	1	Санкт-Петербург, Улиц...	89917345010	Delete
	2	Санкт-Петербург	87812358102	Delete
	3	Москва, Улица Красноа...	81635894131	Delete
	4	Москва, Улица Победы ...	80091923512	Delete

В нижнем правом углу имеются представления для просмотра данных.

Program

Update

	vender_code	total	name	category	description	price	commission
▶	AL23519	10	Стол Шведс...	Мебель	материал: С...	12000.0000	3
	AL12420	25	Диван Шведс...	Мебель	материал: Х...	14000.5000	4
	SD12319	5	Ложки Стол...	Столовые п...	материал: С...	500.3900	2
	SD33117	12	Ножи	Столовые п...	материал: С...	1200.0000	2
	AL23519	19	Стол Шведс...	Мебель	материал: С...	12000.0000	3
	LB12415	45	Рюкзак шко...	Рюкзаки	материал: С...	1599.9900	6
	LB87218	20	Сумка Летняя	Рюкзаки	материал: к...	2800.0000	6
	ME89120	15	LG 7DA2	Монитор	материал: П...	9781.3000	8
	SD12319	10	Ложки Стол...	Столовые п...	материал: С...	500.3900	2
	SD33117	6	Ножи	Столовые п...	материал: С...	1200.0000	2
	SD33117	4	Ножи	Столовые п...	материал: С...	1200.0000	2
	LB87218	10	Сумка Летняя	Рюкзаки	материал: к...	2800.0000	6
	AL23519	2	Стол Шведс...	Мебель	материал: С...	12000.0000	3
	ME89120	0	LG 7DA2	Монитор	материал: П...	9781.3000	8
	LB12415	1	Рюкзак шко...	Рюкзаки	материал: С...	1599.9900	6
	ME89120	18	LG 7DA2	Монитор	материал: П...	9781.3000	8
	AL23519	4	Стол Шведс...	Мебель	материал: С...	12000.0000	3
	SD12319	23	Ложки Стол...	Столовые п...	материал: С...	500.3900	2
*							

Create new order

Create new delivery

Order goods list

Warehouse goods in stock

Employee position

Agent info

Delivery list of goods

Agent goods

Change order status

5

Заказ доставлен

Order_list

Delivery list

Agent goods

Warehouse

Warehouse goods

Warehouse goods info

Справа имеется ряд функциональных кнопок, взаимодействующих с выпадающими списками справа от них. Например, при выборе в списке позиции администратор и нажатия на клавишу

Employee position

Администратор

Администратор

Менеджер

Менеджер склада

Agent info

Update

	name	phone	position	employee_addr
	Зайцева Биргит Ивановна	89273834567	Администратор	Санкт-Петербург
	Main E	89237685355	Администратор	SPB
▶*				

Получаем таблицу всех администраторов БД

Create new order

Create new delivery

В правом верхнем углу имеет две кнопки для создания заказов и поставок на склад, работающих схожим образом.

Create order

Agent name: Alan Paul Net

Good name: Ложки Столовые

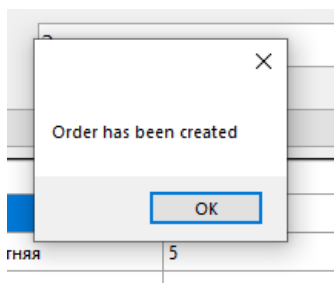
Amount: 3

Add new

Good	total
LG 7DA2	2
Сумка Летняя	5
Ложки Столовые	3

Done

Необходимо выбрать имя агента. Последовательно нажимая имя товара, выбирая его количество и нажимая на кнопку “Add new” создаём список товаров заказа. После выбора всех товаров нажимаем кнопку Done для создания заказа.



24	3	12/10/2020 8:18 PM	1	Delete
▶*				

Форма создания поставки на склад работает аналогичным образом за исключением того, что вместо выбора имени агента, выбирается адрес склада.

Заключение. Вывод

Для выполнения данной курсовой работы мною была выбрана СУБД Microsoft SQL Server из-за таких преимуществ, как хорошая документация, множество примеров в интернете, а также высокая производительность и простота использования. Среда разработки была использована самая популярная на рынке - Microsoft SQL Management Studio.

Наибольшую сложность с работой над курсовой вызвало правильное проектирование таблиц базы данных и отношений между таблицами. С остальными этапами проблем не возникло из-за удобного и понятной семантики языка T-SQL.

Приложение получилось весьма многофункциональное и понятное для пользователя, имеющие множество ограничений, не дающие пользователю вывести из строя программу. Из минусов можно отметить отсутствие ролей в программе, которые не были реализованы по причине нехватки времени и недостатка информационных ресурсов.

Список используемой литературы

1. Kalen, Delaney Inside Microsoft® SQL Server(TM) 2005: Query Tuning and Optimization / Kalen Delaney и др. - М.: Microsoft Press, 2014. - 448 с.
2. Joseph, J. Bambara SQL Server® Developer's Guide / Joseph J. Bambara, Paul R. Allen. - Москва: Мир, 2016. - 235 с.
3. Бьюли, А. Изучаем SQL / А. Бьюли. - М.: Символ-плюс, 2014. - 108 с.