

Управление схемами и таблицами

! В лабораторном практикуме будет использоваться база данных «Склад». Ниже представлено описание предметной области данной базы данных.

В системе управления складом основное внимание уделяется **эффективному** управлению запасами товарных единиц. Эта система необходима для обеспечения точности учета товаров, мониторинга их состояния и управления потоками материалов, что особенно важно для повышения эффективности бизнеса.

Описание системы

Основные цели системы:

- Учет запасов:** Поддержка актуальной информации о количестве и состоянии товаров на складе.
- Оптимизация процессов:** Упрощение логистических процессов, таких как приемка, хранение, перемещение и отгрузка товаров.
- Повышение точности:** Снижение ошибок при обработке заказов и инвентаризации.
- Анализ данных:** Предоставление отчетов и аналитики по запасам, продажам и поставкам для принятия более обоснованных управленческих решений.

Ключевые сущности и их атрибуты

Товар:

- ID:** Уникальный идентификатор товара (INTEGER, PRIMARY KEY, GENERATED BY DEFAULT AS IDENTITY);
- Название:** Полное название товара, которое поможет его идентифицировать (VARCHAR (50), NOT NULL, UNIQUE);
- Категория:** Классификация товара (например, электроника, одежда, продукты питания) (VARCHAR (50), NOT NULL);
- Единица измерения:** Указание, в каких единицах измеряется товар (например, шт., кг, литр) (VARCHAR (10), NOT NULL);
- Количество на складе:** Текущее количество товара, доступного для продажи или использования (INTEGER, NOT NULL, Проверка: (Количество_на_складе >= 0));
- Цена за единицу:** Стоимость одной единицы товара, которая может изменяться в зависимости от рыночной ситуации (DECIMAL (10, 2), NOT NULL, Проверка: (Цена_за_единицу >= 0));
- Описание:** Дополнительная информация о товаре, включая характеристики и особенности (VARCHAR (255)).

Поставщик:

- ID:** Уникальный идентификатор поставщика (INTEGER, PRIMARY KEY, GENERATED BY DEFAULT AS IDENTITY);
- Название:** Наименование компании-поставщика (VARCHAR (50), NOT NULL, UNIQUE);

- **Контактное лицо:** Имя и фамилия лица, ответственного за взаимодействие с поставщиком (VARCHAR (255), NOT NULL);
- **Телефон:** Контактный номер телефона для связи (VARCHAR (20), NOT NULL, UNIQUE);
- **Адрес:** Физический адрес поставщика, который может использоваться для доставки товаров (VARCHAR (255), NOT NULL);
- **Email:** Электронная почта для оперативного общения (VARCHAR (50) NOT NULL).

Заказ:

- **ID:** Уникальный идентификатор заказа (INTEGER, PRIMARY KEY, GENERATED BY DEFAULT AS IDENTITY);
- **Дата заказа:** Дата, когда был создан заказ (DATE, NOT NULL);
- **ID_поставщика:** Ссылка на идентификатор поставщика, у которого был сделан заказ (INTEGER, NOT NULL, Связь с Поставщик(id_Поставщика));
- **Статус:** Текущий статус заказа (VARCHAR (20), NOT NULL, Проверка:(Значения: 'Ожидается', 'Выполнен', 'Отменен'));
- **Общая сумма:** Общая стоимость всех товаров в заказе (DECIMAL (15, 2), NOT NULL, Проверка: (Общая_сумма >= 0));

Заказанный товар:

- **ID:** Уникальный идентификатор записи о заказанном товаре (INTEGER, PRIMARY KEY, GENERATED BY DEFAULT AS IDENTITY);
- **ID_заказа:** Ссылка на идентификатор заказа, к которому относится данный товар (INTEGER, NOT NULL, Связь с Заказ(id_Заказа));
- **ID_товара:** Ссылка на идентификатор товара (INTEGER, NOT NULL, Связь с Товар(id_Товара));
- **Количество:** Количество единиц товара, заказанных в рамках заказа (INTEGER, NOT NULL, Проверка:(Количество > 0)).

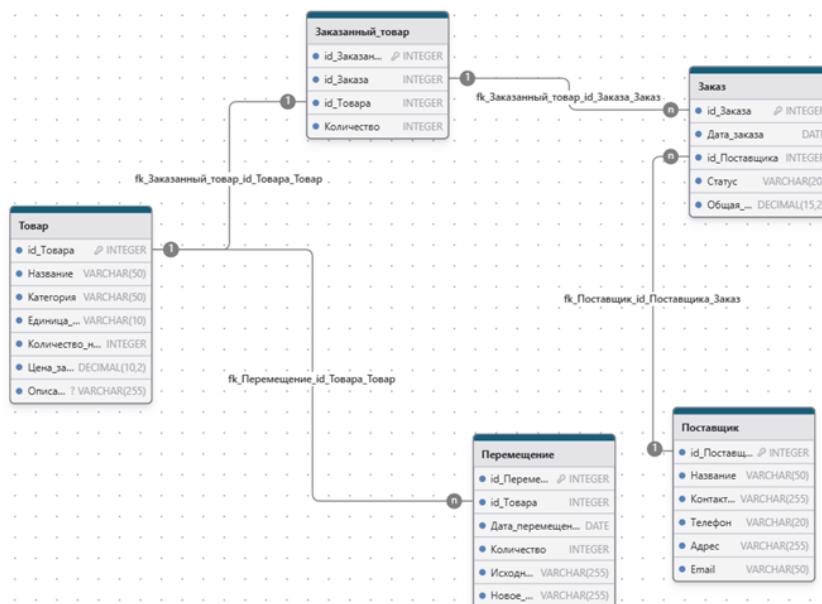
Перемещение:

- **ID:** Уникальный идентификатор записи о перемещении товара (INTEGER, PRIMARY KEY, GENERATED BY DEFAULT AS IDENTITY);
- **ID_товар:** Ссылка на идентификатор товара, который перемещается (INTEGER, NOT NULL, Связь с Товар(id_Товара));
- **Дата перемещения:** Дата, когда было осуществлено перемещение (DATE, NOT NULL);
- **Количество:** Количество единиц товара, которые были перемещены (INTEGER, NOT NULL, Проверка: (Количество > 0));
- **Исходное местоположение:** Место, откуда товар был перемещен (VARCHAR(255), NOT NULL);
- **Новое местоположение:** Место, куда товар был перемещен (VARCHAR(255) NOT NULL).

Поставщик ---> **Заказ** - Поставщик может генерировать множество заказов. Каждый заказ связан только с одним поставщиком.

Заказанный товар ---> **Товар** - Каждый заказанный товар соответствует одному конкретному товару, но один товар может быть частью множества различных заказов.

Товар ---> **Перемещение** - Товар может перемещаться между различными местоположениями на складе, создавая записи о перемещении. Одно перемещение может касаться только одного товара, но одно и то же местоположение может быть связано со множеством записей о перемещениях разных товаров.



Создание системы

Создадим базу данных «**Склад**», затем, нажав на нее правой кнопкой мыши выбрать «**Запросник**». И выполним запрос, показанный на рисунке 21. После выполнения данного запроса будут созданы 3 таблицы: «**Товар**», «**Поставщик**» и «**Заказ**».

! Обратите внимание на ограничения!

```

CREATE TABLE Товар (
    id_Товара INTEGER PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY,
    Название VARCHAR(50) NOT NULL UNIQUE,
    Категория VARCHAR(50) NOT NULL,
    Единица_измерения VARCHAR(10) NOT NULL,
    Количество_на_складе INTEGER NOT NULL CHECK (Количество_на_складе >= 0),
    Цена_за_единицу DECIMAL(10, 2) NOT NULL CHECK (Цена_за_единицу >= 0),
    Описание VARCHAR(255)
);

CREATE TABLE Поставщик (
    id_Поставщика INTEGER PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY,
    Название VARCHAR(50) NOT NULL UNIQUE,
    Контактное_лицо VARCHAR(255) NOT NULL,
    Телефон VARCHAR(20) NOT NULL UNIQUE,
    Адрес VARCHAR(255) NOT NULL,
    Email VARCHAR(50) NOT NULL
);

CREATE TABLE Заказ (
    id_Заказа INTEGER PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY,
    Дата_заказа DATE NOT NULL,
    id_Поставщика INTEGER NOT NULL,
    Статус VARCHAR(20) NOT NULL CHECK(Статус IN ('Ожидается', 'Выполнен', 'Отменен')),
    Общая_сумма DECIMAL(15, 2) NOT NULL CHECK (Общая_сумма >= 0),
    FOREIGN KEY (id_Поставщика) REFERENCES Поставщик(id_Поставщика)
);

```

В каждой таблице имеются идентификаторы (например, в таблице «Товар» — это «`id_Товара`»). Данные поля предназначены для уникальной идентификации записей в таблицах.

У этих полей имеются следующие ограничения:

- PRIMARY KEY** – данное поле будет ключевым (этот параметр делает каждое значение этого поля уникальным, и именно благодаря ему будет происходить уникальная идентификация записей);
- GENERATED BY DEFAULT AS IDENTITY** – автоматическая нумерация строк (столбец с автоинкрементными значениями).

Также в таблицах присутствуют другие ограничения:

- **CHECK** – проверка соответствия заданному условию
 - например, в таблице «Товар» столбец «Количество_на_складе» должен иметь значение большему, либо равному 0;
 - или в таблице «Заказ» столбец «Статус» может быть занесено только одно из трех значений: «Ожидается», «Выполнен» или «Отменен»).
- **UNIQUE** – проверка уникальности значения в столбце, то есть значения в данном столбце не могут повторяться.
- **FOREIGN KEY** – это столбец или столбцы, которые ссылаются на первичный ключ другой таблицы. Он нужен для того, чтобы связать две разные таблицы между собой.
 - В таблице «Заказ» столбец «`id_Поставщика`» ссылается на ключевое поле в таблице «Поставщик». Значения в данном поле не должны противоречить значениям одноименного поля из таблицы «Поставщик».

! Все имеющиеся типы данных изучите на официальном сайте PostgreSQL
<https://www.postgresql.org/docs/current/datatype.html>.

В нашем случае изучите следующие типы данных: **INTEGER, VARCHAR, DECIMAL, DATE**.

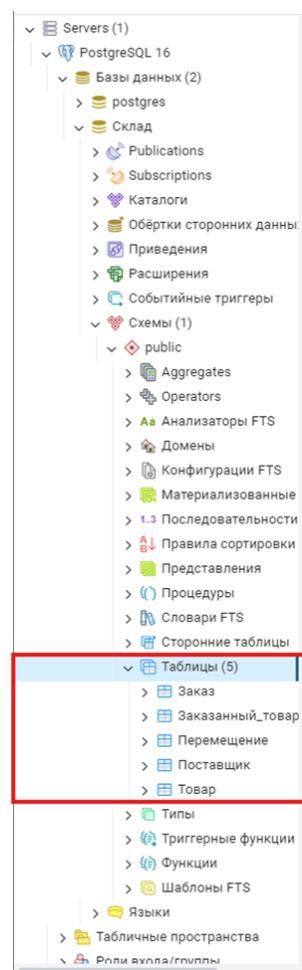
После завершения запроса, должно отобразиться сообщение. Данное сообщение информирует о том, что запрос выполнен успешно.

```
Data Output Сообщения Notifications
CREATE TABLE

Запрос завершён успешно, время выполнения: 53 msec.
```

Исходя из описания предметной области, создайте оставшиеся таблицы.

После создания таблиц раскройте вкладку «**Схемы**», «**Таблицы**» и убедитесь, что все таблицы созданы.



Заполнение системы

Добавим данные в таблицу «**Товар**». Для этого создадим запрос.

```
1 ✓ INSERT INTO Товар (Название, Категория, Единица_измерения, Количество_на_складе, Цена_за_единицу, Описание)
2 VALUES
3 ('Смартфон Samsung Galaxy S23 Ultra 12/256GB Black', 'Электроника', 'шт.', 50, 119990, '6.8" Dynamic AMOLED 2X, 200MP камера, Snapdragon 8 Gen 2, 5000mAh');
```

Data Output Сообщения Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 57 msec.

Памятка

```
INSERT INTO Таблица (Столбец)
VALUES ('Значение1'), ('Значение2');
```

Конструкция **INSERT INTO** отвечает за добавление данных в таблицу:

сначала необходимо написать ту таблицу, в которую добавляются данные, затем в скобках перечислить столбцы, в которые нужно добавить данные. После этого написать ключевое слово **VALUES** и в скобках перечислить те значения, которые требуется добавить (обратите внимание, что текстовые значения пишутся в одинарных кавычках). Если требуется добавить несколько строк, то после закрытия скобки, где перечисляются данные, необходимо поставить запятую , и в скобках перечислить значения второй строки и т.д. Когда требуется завершить процесс добавления, в конце нужно поставить ;

В нашем примере добавлена одна строка. Также обратите внимание на то, что первое поле **id_Товара** не заполняется.

? Объясните почему его не нужно вписывать в конструкцию **INSERT INTO**?

Убедимся, что данные добавлены. Для этого выполним запрос.

Запрос История запросов							Scratch Pad >
1 SELECT * FROM Товар							
Data Output Сообщения Notifications							
	id_Товара	Название	Категория	Единица_измерения	Количество_на_складе	Цена_за_единицу	Описание
1	1	Смартфон Samsung Galaxy S23 Ultra 12/256GB Black	Электроника	шт.	50	119990.00	6.8" Dynamic AMOLED 2X, 200MP камера, Snapdragon 8 Gen 2, 5000mAh

Памятка

```
SELECT Столбец FROM Таблица
```

Конструкция **SELECT** отвечает за выборку данных.

В нашем случае мы выводим все данные, поэтому после ключевого слова **SELECT** пишется * (это значит, что выводятся данные из всех столбцов). После **FROM** пишется название таблицы.

Изменим размер поля «Название» в таблице «Товар» с 50 символов на 100. Для этого в списке с таблицами нажмем на таблицу «Товар» правой кнопкой мыши и выберем «Properties...». В разделе «Столбцы» изменим значение «Length/Precision» у столбца «название» с 50 на 100.

Товар

General Столбцы Дополнительно Ограничения Параметры Безопасность SQL

Наследуется из таблицы(ы) Выберите источник наследования...

Столбцы

	Имя	Тип данных	Length/Precision	Масш...	Не NULL?	Первичный ...	По умолчан...
	id_Товара	integer	100		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	Название	character varying	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Категория	character varying	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Единица_измерени	character varying	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Количество_на_ск.	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Цена_за_единицу	numeric	10	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Описание	character varying	255		<input type="checkbox"/>	<input type="checkbox"/>	

Закрыть Сбросить Сохранить

Заполните самостоятельно

Заполните все имеющиеся таблицы. Все данные представлены ниже.

ID	Название	Категория	Единица измерения	Количество на складе	Цена за единицу	Описание
1	Смартфон Samsung Galaxy S23 Ultra 12/256GB Black	Электроника	шт.	50	119990.00	6.8" Dynamic AMOLED 2X, 200MP камера, Snapdragon 8 Gen 2, 5000mAh
2	Футболка мужская хлопковая однотонная, размер L	Одежда	шт.	200	999.00	100% хлопок, классический крой, разные цвета в ассортименте
3	Молоко "Простоквашино" 3.2% 1 литр	Продукты питания	литр	150	85.50	Пастеризованное, срок годности 7 дней
4	Жесткий диск Seagate Barracuda 2TB 7200RPM	Электроника	шт.	25	5490.00	3.5", SATA III, 7200RPM, 256MB cache, для настольных компьютеров
5	Сахар-песок развесной	Продукты питания	кг	500	65.00	Белый кристаллический сахар, ГОСТ 33222-2015
6	Джинсы мужские классические, размер 32/34	Одежда	шт.	100	2999.00	Прямой крой, деним средней плотности, темно-синий цвет
7	Наушники беспроводные Sony WH-1000XM5 Black	Электроника	шт.	30	34990.00	Шумоподавление, Bluetooth 5.2, до 30 часов работы, Hi-Res Audio
8	Яблоки Голден	Продукты питания	кг	300	99.00	Сладкие и сочные, сорт Голден Делишес
9	Рубашка женская офисная белая, размер 44	Одежда	шт.	75	1799.00	Приталенный силуэт, длинный рукав, 100% хлопок
10	Мышь компьютерная беспроводная Logitech MX Master 3S	Электроника	шт.	20	9990.00	Высокоточный сенсор, MagSpeed колесо прокрутки, эргономичный дизайн, Bluetooth/Logi Bolt USB Receiver

Данные для таблицы Товар

ID	Название	Контактное лицо	Телефон	Адрес	Email
1	ООО "ТехноИмпорт"	Иванов Иван Иванович	+7 (495) 123-45-67	Москва, ул. Ленина, д. 10, офис 201	info@tehnoimport.ru
2	ИП Сидоров А.П.	Сидоров Алексей Петрович	+7 (916) 987-65-43	Московская обл., г. Химки, Юбилейный проспект, д. 15, кв. 32	sidorov.ap@mail.ru
3	ООО "Продуктовый Рай"	Петрова Елена Сергеевна	+7 (812) 555-33-22	Санкт-Петербург, Невский проспект, д. 5, офис 12	sales@produktrai.spb.ru
4	ООО "ТекстильОпт"	Кузнецова Ольга Ивановна	+7 (4932) 77-88-99	Иваново, ул. Калинина, д. 25, склад 7	opt.textile@mail.com
5	ООО "Электроника Плюс"	Смирнов Дмитрий Андреевич	+7 (383) 222-11-00	Новосибирск, Красный проспект, д. 82, офис 505	electronics.plus@ngs.ru
6	ООО "Фруктовый сад"	Васильев Игорь Николаевич	+7 (861) 333-22-11	Краснодар, ул. Северная, д. 400, склад 10	fruitsad.krd@yandex.ru
7	ИП Морозова Е.В.	Морозова Елена Васильевна	+7 (926) 111-22-33	Москва, ул. Тверская, д. 16/2, оф. 4	morozova.ev@gmail.com
8	ООО "Одежда для всех"	Зайцева Анна Петровна	+7 (473) 444-33-22	Воронеж, ул. Плехановская, д. 12, офис 3	odejda.vrn@mail.ru
9	ООО "Компьютерные системы"	Лебедев Сергей Иванович	+7 (843) 555-44-33	Казань, ул. Баумана, д. 70, офис 20	compsys.kzn@inbox.ru
10	ООО "Бакалея от Иванича"	Иванов Иван Петрович	+7 (903) 666-55-44	Москва, Варшавское шоссе, д. 141, стр. 8, офис 101	bakaleya@ivanich.ru

Данные для таблицы Поставщик

ID	Дата заказа	ID_поставщика	Статус	Общая сумма
1	2024-01-15	1	Выполнен	120000.00
2	2024-01-20	3	Ожидается	50000.00
3	2024-01-25	2	Выполнен	25000.00
4	2024-02-01	4	Ожидается	75000.00
5	2024-02-05	1	Отменен	10000.00
6	2024-02-10	5	Выполнен	150000.00
7	2024-02-15	3	Ожидается	30000.00
8	2024-02-20	6	Выполнен	60000.00
9	2024-02-25	2	Ожидается	18000.00
10	2024-03-01	1	Выполнен	200000.00

Данные для таблицы Заказ

ID	ID_заказа	ID_товара	Количество
1	1	1	2
2	1	4	1
3	2	3	100
4	2	8	50
5	3	2	50
6	3	6	20
7	4	2	100
8	4	9	25
9	5	1	1
10	6	1	3
11	6	7	2
12	7	3	200
13	8	6	30
14	8	10	5
15	9	2	30
16	10	1	5
17	10	4	3
18	10	7	1

Данные для таблицы Заказанный товар

ID	ID_товар	Дата перемещения	Количество	Исходное местоположение	Новое местоположение
1	1	2024-03-05	10	Склад №1	Торговый зал
2	3	2024-03-05	50	Холодильная камера №2	Торговый зал
3	2	2024-03-06	20	Склад №2	Зона распродаж
4	4	2024-03-06	5	Склад №1	Сервисный центр
5	8	2024-03-07	100	Склад №3	Торговый зал
6	6	2024-03-07	30	Склад №2	Зона уцененных товаров
7	1	2024-03-08	5	Торговый зал	Резервный склад
8	7	2024-03-08	2	Склад №1	Торговый зал
9	10	2024-03-09	3	Склад №1	Сервисный центр
10	3	2024-03-09	25	Торговый зал	Холодильная камера №1

Данные для таблицы Перемещение

Экспорт данных

Следующим этапом выполним экспорт данных. Экспортируем таблицу «Заказы» в свою папку. Данные будут скопированы в файл **Заказы.csv**.

CSV — текстовый формат для представления табличных данных. Стока таблицы соответствует строке текста, которая содержит одно или несколько полей, разделённых запятыми.

Запрос История запросов

```
1 ✓ copy "Заказ" ("id_Заказа", "Дата_заказа", "id_Поставщика", "Статус", "Общая_сумма")
2 TO 'D:/Zakaz.csv'
3 DELIMITER ','
4 CSV HEADER
5 ENCODING 'WIN1251'
6 QUOTE """
7 ESCAPE """;
```

Data Output Сообщения Notifications

COPY 10

Запрос завершён успешно, время выполнения: 28 msec.

Объяснение запроса:

- **copy** - это команда PostgreSQL, используемая для копирования данных в файлы и из базы данных.
- **"Заказ"** - ссылается на таблицу **Заказ** в схеме public.
- **("id_Заказа", "Дата_заказа", "id_Поставщика", "Статус", "Общая_сумма")** - указывает столбцы, которые будут экспортированы.
- **TO 'D:/Zakaz.csv'** - Путь к выходному файлу CSV.
- **DELIMITER ','** - устанавливает запятую как разделитель.
- **CSV HEADER** - включает названия столбцов в первую строку CSV.
- **ENCODING 'WIN1251'** - указывает кодировку WIN1251 для выходного файла.
- **QUOTE ""** - определяет символ кавычек для строковых полей.
- **ESCAPE """** - устанавливает управляемый символ для одинарных кавычек.

После выполнения запроса в папке, указанной выше, появится файл с названием **Zakaz**. Запустите файл. Убедитесь, что данные экспорттировались.

	id_Заказа	Дата_заказа	id_Поставщика	Статус	Общая_сумма
1	2024-01-15	1	Выполнена	120000.00	
2	2024-01-20	3	Ожидается	50000.00	
3	2024-01-25	2	Выполнена	25000.00	
4	2024-02-01	4	Ожидается	75000.00	
5	2024-02-05	1	Отменен	10000.00	
6	2024-02-10	5	Выполнена	150000.00	
7	2024-02-15	3	Ожидается	30000.00	
8	2024-02-20	6	Выполнена	60000.00	
9	2024-02-25	2	Ожидается	18000.00	
10	2024-03-01	1	Выполнена	200000.00	

Самостоятельно экспортируйте 2 любые таблицы.

Импорт данных

Выполним импорт данных в таблицу «Заказанный_товар». Для этого создадим файл Excel и сохраним его в формате CVS (разделитель – запятая). Название файла «Zakaz_tovar». В нем будут храниться данные для импорта в базу данных.

Имя файла:	Zakaz_tovar
Тип файла:	CSV (разделитель — запятая)

1. Откройте файл и заполните его содержимым. **Обратите внимание**, что все записано в ОДНОМ столбце. В первой строке написаны названия столбцов таблицы «Заказанный_товар» за исключением ключевого поля, так как он заполняется автоматически.

A
1 "id_Заказа","id_Товара","Количество"
2 1,1,2
3

2. Закройте файл.

Напишем запрос

```
Запрос История запросов
1 copy "Заказанный_товар" ("id_Заказа", "id_Товара", "Количество")
2 FROM 'D:/Zakaz_tovar.csv'
3 WITH (FORMAT CSV, DELIMITER ',', HEADER, ENCODING 'WIN1251', QUOTE '\"', ESCAPE '\"');
```

Объяснение запроса:

- **copy** - это мета-команда, специфичная для утилиты psql (командной строки PostgreSQL). Она используется для быстрого импорта данных из файла в таблицу (или экспорта данных из таблицы в файл). Важно помнить, что это не стандартный SQL запрос.
- **“Заказанный_товар”** - это имя таблицы, в которую будут импортированы данные.
- **“id_Заказанного_товара”, “id_Заказа”, “id_Товара”, “Количество”** - это имена столбцов. Опять же, двойные кавычки используются для обработки имен столбцов, содержащих специальные символы или требующих сохранения регистра. Порядок следования колонок должен совпадать с порядком столбцов в CSV файле.
- **FROM 'D:/Zakaz_tovar.csv'** - это указывает на путь к файлу CSV, из которого будут считаны данные.
- **WITH (FORMAT CSV, DELIMITER ',', HEADER, ENCODING 'WIN1251', QUOTE '\"', ESCAPE '\"')** - это список опций, которые определяют формат данных в файле CSV.
 - **FORMAT CSV** - указывает, что файл имеет формат CSV (Comma Separated Values - значения, разделенные запятыми).
 - **DELIMITER ','** - указывает, что запятая , используется в качестве разделителя между полями в файле CSV.
 - **HEADER** - указывает, что первая строка файла содержит заголовки столбцов. Эти заголовки будут использованы для сопоставления данных со столбцами в таблице.

- **ENCODING 'WIN1251'** - указывает, что файл закодирован в кодировке WIN1251. Это важно указать правильно, чтобы символы в файле (особенно русские буквы) были интерпретированы корректно.
- **QUOTE ""** - указывает, что двойные кавычки " используются для заключения полей, содержащих запятые или другие специальные символы.
- **ESCAPE '""'** - указывает, что одинарная кавычка ' используется в качестве escape-символа. Escape-символ используется для экранирования специальных символов внутри полей, заключенных в кавычки. Например, если поле содержит двойную кавычку, ее можно экранировать с помощью escape-символа. Так как сам escape-символ должен быть внутри строки, то его нужно экранировать.

После выполнения запроса для импорта, убедимся, что данные добавились в таблицу «Заказанный_товар». Для этого выполним запрос на выборку данных.

1 SELECT * FROM Заказанный_товар					
Data Output Сообщения Notifications					
	id_Заказанного_товара [PK] integer	id_Заказа integer	id_Товара integer	Количество integer	
1		1	1	1	2
2		2	1	4	1
3		3	2	3	100
4		4	2	8	50
5		5	3	2	50
6		6	3	6	20
7		7	4	2	100
8		8	4	9	25
9		9	5	1	1
10		10	6	1	3
11		11	6	7	2
12		12	7	3	200
13		13	8	6	30
14		14	8	10	5
15		15	9	2	30
16		16	10	1	5
17		17	10	4	3
18		18	10	7	1
19		19	1	1	2

Самостоятельно Импортируйте данные в 2 любые таблицы (2-3 строки).