

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное
учреждение
высшего профессионального образования
Национальный исследовательский университет
"Высшая школа экономики"**

Московский институт электроники и математики

Департамент прикладной математики

Образовательная программа "Компьютерная безопасность"

Отчёт по практической работе по дисциплине "СУБД"

**Домашняя практическая работа № 2
"Администрирование баз данных"**

Вариант 26

**Выполнил:
Рымкулова Диана СКБ181**

**Проверил:
Профессор, Белов А.В.**

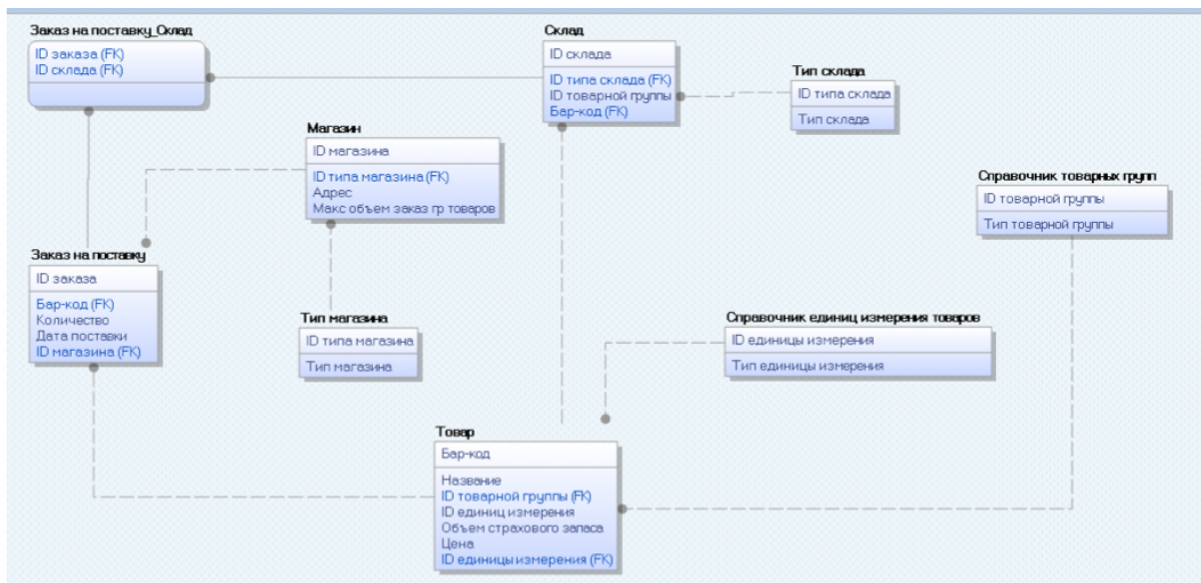
Москва-2022

Постановка задачи

Выбранная среда реализации работы

PostgreSQL

Создание БД



Код создания базы данных и базовых объектов размещен в [приложении 1](#).

Настройка ограничений целостности

При создании базы данных были использованы следующие операторы: PRIMARY KEY, UNIQUE, NOT NULL, ON DELETE RESTRICT, ON UPDATE CASCADE, CHECK. Ниже подробнее описано для чего применяются такие ограничения.

1. PRIMARY KEY - это параметр, который устанавливается для однозначной идентификации той или иной записи в таблице. Значения PRIMARY KEY должны быть всегда уникальны, а так же не содержать значений NULL.
2. UNIQUE - это ограничение предотвращает любую попытку ввода в это поле для одной из строк, значения, которое уже представлено в другой строке.
3. NOT NULL - это ограничение предотвращает появление пустых полей.

4. ON DELETE RESTRICT - это ограничение не позволит удалить записи из таблиц, на которые ссылаются записи из других таблиц.
5. ON UPDATE CASCADE - это ограничение позволяет осуществлять следующее действие: при изменении записи, на которую ссылаются другие записи, все связанные внешним ключом записи также изменятся.
6. CHECK - это ограничение, которое позволяет осуществлять проверку вводимых значений записей.

Импорт и экспорт данных

INSERT INTO "Справочник товарных групп" ("ID товарной группы", "Тип товарной группы")

VALUES

(1, 'Молочно-масляные'),
(2, 'Хлебобулочные изделия'),
(3, 'Фрукты и овощи'),
(4, 'Кондитерские'),
(5, 'Безалкогольные напитки'),
(6, 'Мясные и колбасные изделия'),
(7, 'Табачные'),
(8, 'Непродуктовые');

INSERT INTO "Справочник единиц измерения" ("ID единицы измерения", "Тип единицы измерения")

VALUES

(1, 'кг'),
(2, 'л'),
(3, 'паллет'),
(4, 'штука'),
(5, 'коробка');

INSERT INTO "Тип магазина" ("ID типа магазина", "Тип магазина")

VALUES

(1, 'Минимаркет'),
(2, 'Супермаркет'),
(3, 'Гипермаркет');

INSERT INTO "Тип склада" ("ID типа склада", "Тип склада")

VALUES

(1, 'Сырья, материалов, комплектующих'),

```

(2, 'Незавершенного производства/полуфабрикатов'),
(3, 'Готовой продукции'),
(4, 'Тары'),
(5, 'Остатков и отходов'),
(6, 'Инструментов');
INSERT INTO "Товар" ("Бар-код", "Название", "ID товарной группы", "ID
единицы измерения", "Объем страхового запаса", "Цена")
VALUES
(13457869, 'Творожок «Вкус»', 1, 1, 200, 57),
(34678654, 'Молоко «Простоквашино»', 1, 2, 400, 89),
(67548967, 'Гренки «Корочка»', 2, 4, 150, 43),
(56324356, 'Мука «Сдоба»', 2, 3, 350, 45),
(67345678, 'Помидоры абхазские', 3, 1, 120, 123),
(54324589, 'Яблоки «Голден»', 3, 1, 200, 75),
(45327863, 'Леденцы «Кактус»', 4, 5, 70, 220),
(67946324, 'Торт «Чародейка»', 4, 4, 50, 430),
(67453456, 'Напиток «Колокольчик»', 5, 4, 350, 65),
(34569078, 'Колбаса краковская', 6, 4, 120, 270),
(67508976, 'Коробка 35*45*45', 8, 4, 700, 55);

```

```

INSERT INTO "Склад" ("ID склада", "ID типа склада", "ID товарной
группы", "Бар-код")
VALUES

```

```

(1, 3, 1, 13457869),
(2, 5, 3, 67345678),
(3, 4, 8, 67508976),
(4, 3, 1, 34678654),
(5, 2, 2, 56324356),
(6, 3, 3, 54324589),
(7, 3, 2, 67548967),
(8, 3, 4, 67946324),
(9, 3, 4, 45327863);

```

```

INSERT INTO "Магазин" ("ID магазина", "ID типа магазина", "Название",
"Адрес", "Макс. объем заказа группы товаров")
VALUES

```

```

(1, 1, 'Подсолнух', 'Солнечная, 56', 400),
(2, 3, 'Ромашка', 'Бульвар Мира, 4к1', 500),

```

```

(3, 2, 'Василек', 'Проспект Мира, 67', 350),
(4, 2, 'Роза', 'Республиканская, 78', 400),
(5, 1, 'Лепесток', 'Ленина, 45к1', 450),
(6, 3, 'Гвоздика', 'Проспект Маршала Катукова, 56', 600),
(7, 2, 'Лютик', 'Пресненская набережная, 1', 350),
(8, 1, 'Лилия', 'Авиационная, 79', 550),
(9, 3, 'Тюльпан', 'Андреевка, 4к2', 400),
(10, 2, 'Фиалка', 'Астраханский проезд, 23', 200);
INSERT INTO "Заказ на поставку" ("ID заказа", "ID магазина", "Бар-код",
"Количество", "Дата поставки")
VALUES
(1, 5, 54324589, 1000, '2/3/2022'),
(2, 4, 67548967, 40, '4/3/2022'),
(3, 3, 13457869, 70, '3/3/2022'),
(4, 2, 67345678, 30, '5/3/2022'),
(5, 7, 67946324, 40, '7/3/2022'),
(6, 8, 34678654, 150, '10/3/2022'),
(7, 6, 67508976, 50, '11/3/2022'),
(8, 9, 56324356, 20, '6/3/2022'),
(9, 10, 45327863, 60, '5/3/2022'),
(10, 1, 54324589, 90, '12/3/2022');
INSERT INTO "Заказ на поставку_Склад" ("ID заказа", "ID склада")
VALUES
(1, 6),
(2, 7),
(3, 1),
(4, 2),
(5, 8),
(6, 4),
(7, 3),
(8, 5),
(9, 9),
(10, 6);

```

С помощью команды COPY:

```

CREATE TABLE "Справочник" ("ID товарной группы " INT NOT NULL,
"Тип товарной группы " VARCHAR (255) NOT NULL);

```

COPY "Справочник" FROM '/Users/dianarymkulova/Documents/group.csv'
DELIMITER ',' CSV HEADER;

С помощью Import/Export data:

Выбираем необходимый файл из которого загружаем записи

The screenshot shows the 'Import/Export data - table 'Справочник'' dialog box with the 'Options' tab selected. The 'Import/Export' section has 'Import' selected. Under 'File Info', 'Filename' is '/Users/dianarymkulova/Documents/group.csv', 'Format' is 'csv', and 'Encoding' is 'Select an item...'. Under 'Miscellaneous', 'OID' and 'Header' are toggled off, and 'Delimiter' is 'Select an item...'. A descriptive text for the delimiter is provided. At the bottom are buttons for 'Close', 'Reset', and 'OK'.

Указываем колонки из файла в верном порядке, после чего нажимаем ОК и получаем заполненную таблицу

The screenshot shows the 'Import/Export data - table 'Справочник'' dialog box with the 'Columns' tab selected. The 'Columns to export' section shows a list with 'ID товарной группы' and 'Тип товарной группы'. Below this is a text box for 'NULL Strings' and another for 'Not null columns', both with descriptive text. At the bottom are buttons for 'Close', 'Reset', and 'OK'.

Результатом такого импорта являются отдельные таблицы вида:

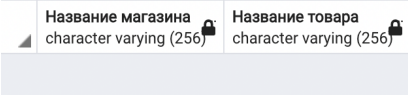
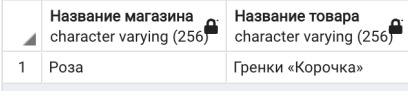
	ID заказа [PK] integer	ID магазина integer	Бар-код integer	Количество integer	Дата поставки date
1	2	4	67548967	40	2022-04-03
2	3	3	13457869	70	2022-03-03
3	4	2	67345678	30	2022-05-03
4	5	7	67946324	40	2022-07-03
5	6	8	34678654	150	2022-10-03
6	7	6	67508976	50	2022-11-03
7	8	9	56324356	20	2022-06-03
8	9	10	45327863	60	2022-05-03
9	10	1	54324589	90	2022-12-03
10	1	5	54324589	1000	2022-02-03

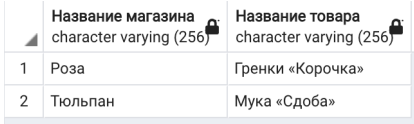
Формирование запросов к БД на языке SQL

1. Сформировать список заказов по магазинам

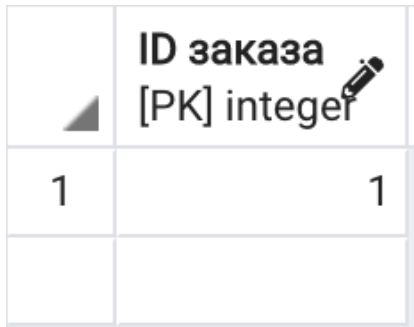
Запрос	Демонстрация результата	Комментарий																																	
SELECT "Название", "ID заказа" FROM "Магазин" INNER JOIN "Заказ на поставку" ON "Заказ на поставку"."ID магазина" = "Магазин"."ID магазина»	<table> <tr> <th></th><th>Название character varying (256)</th><th>ID заказа integer</th></tr> <tr><td>1</td><td>Лепесток</td><td>1</td></tr> <tr><td>2</td><td>Роза</td><td>2</td></tr> <tr><td>3</td><td>Василек</td><td>3</td></tr> <tr><td>4</td><td>Ромашка</td><td>4</td></tr> <tr><td>5</td><td>Лютик</td><td>5</td></tr> <tr><td>6</td><td>Лилия</td><td>6</td></tr> <tr><td>7</td><td>Гвоздика</td><td>7</td></tr> <tr><td>8</td><td>Тюльпан</td><td>8</td></tr> <tr><td>9</td><td>Фиалка</td><td>9</td></tr> <tr><td>10</td><td>Подсолнух</td><td>10</td></tr> </table>		Название character varying (256)	ID заказа integer	1	Лепесток	1	2	Роза	2	3	Василек	3	4	Ромашка	4	5	Лютик	5	6	Лилия	6	7	Гвоздика	7	8	Тюльпан	8	9	Фиалка	9	10	Подсолнух	10	Результат содержит несколько записей
	Название character varying (256)	ID заказа integer																																	
1	Лепесток	1																																	
2	Роза	2																																	
3	Василек	3																																	
4	Ромашка	4																																	
5	Лютик	5																																	
6	Лилия	6																																	
7	Гвоздика	7																																	
8	Тюльпан	8																																	
9	Фиалка	9																																	
10	Подсолнух	10																																	

2. Получить список товаров выбранной товарной группы,
поставленных в конкретный магазин за период

Запрос	Демонстрация результата	Комментарий
<p>SELECT "Магазин"."Название", "Товар"."Название" FROM "Заказ на поставку"</p> <p>INNER JOIN "Товар" ON "Товар"."Бар-код" = "Заказ на поставку"."Бар-код"</p> <p>INNER JOIN "Магазин" ON "Магазин"."ID магазина" = "Заказ на поставку"."ID магазина"</p> <p>WHERE ("Дата поставки" BETWEEN '2021-02-03' AND '2021-04-03') AND "ID товарной группы" = '2'</p>		<p>Пустой результат (нет поставок такого товара за период)</p>
<p>SELECT "Магазин"."Название", "Товар"."Название" FROM "Заказ на поставку"</p> <p>INNER JOIN "Товар" ON "Товар"."Бар-код" = "Заказ на поставку"."Бар-код"</p> <p>INNER JOIN "Магазин" ON "Магазин"."ID магазина" = "Заказ на поставку"."ID магазина"</p> <p>WHERE ("Дата поставки" BETWEEN '2022-02-03' AND '2022-04-03') AND "ID товарной группы" = '2'</p>		<p>Результат состоит из одной записи</p>

<pre>SELECT "Магазин"."Название", "Товар"."Название" FROM "Заказ на поставку" INNER JOIN "Товар" ON "Товар"."Бар-код" = "Заказ на поставку"."Бар-код" INNER JOIN "Магазин" ON "Магазин"."ID магазина" = "Заказ на поставку"."ID магазина" WHERE ("Дата поставки" BETWEEN '2021-02-03' AND '2023-04-03') AND "ID товарной группы" = '2'</pre>		<p>Результат содержит несколько записей</p>
---	--	---

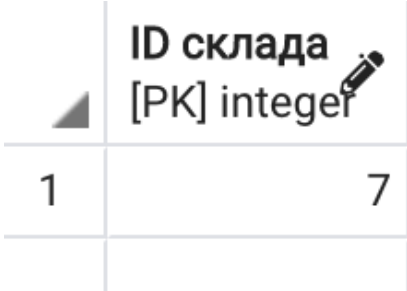
3. Получить список товаров объем заказов, на которые превышают страховой запас.

Запрос	Демонстрация результата	Комментарий
<pre>SELECT "ID заказа" FROM "Заказ на поставку" INNER JOIN "Товар" ON "Товар"."Бар-код" = "Заказ на поставку"."Бар-код" WHERE "Заказ на поставку"."Количество" > "Товар"."Объем страхового запаса"</pre>		<p>Результат содержит одну запись</p>

4. Получить список складов, в которых хранятся товары в паллетах.

Запрос	Демонстрация результата	Комментарий						
<pre>SELECT "ID склада" FROM "Склад" INNER JOIN "Товар" ON "Товар"."ID товарной группы" = "Склад"."ID товарной группы" INNER JOIN "Справочник единиц измерения" ON "Справочник единиц измерения"."ID единицы измерения" = "Товар"."ID единицы измерения"</pre>	<table><tr><td></td><td>ID склада [PK] integer</td></tr><tr><td>1</td><td>5</td></tr><tr><td>2</td><td>7</td></tr></table>		ID склада [PK] integer	1	5	2	7	Результат содержит одну запись
	ID склада [PK] integer							
1	5							
2	7							

5. Получить номер склада, на котором хранятся те же товарные группы, какие заказываются выбранным магазином

Запрос	Демонстрация результата	Комментарий				
<pre>SELECT "ID склада" FROM "Склад" INNER JOIN "Заказ на поставку" ON "Заказ на поставку"."ID заказа" = "Склад"."ID склада" WHERE "ID магазина" = '6' AND "ID товарной группы" = '2'</pre>	 <table><thead><tr><th></th><th>ID склада [PK] integer</th></tr></thead><tbody><tr><td>1</td><td>7</td></tr></tbody></table>		ID склада [PK] integer	1	7	Результат содержит одну запись
	ID склада [PK] integer					
1	7					

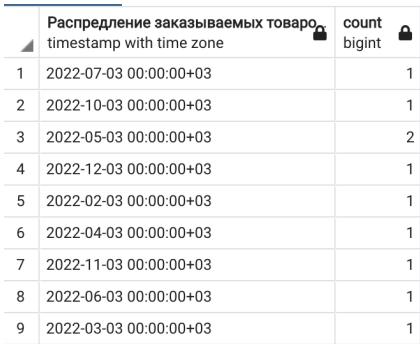
<pre>SELECT "ID склада" FROM "Склад" INNER JOIN "Заказ на поставку" ON "Заказ на поставку"."ID заказа" = "Склад"."ID склада" WHERE "ID магазина" = '6' AND "ID товарной группы" = '2' OR "ID товарной группы" = '3'</pre>	<table><tr><th></th><th>ID склада [PK] integer</th></tr><tr><td>1</td><td>2</td></tr><tr><td>2</td><td>6</td></tr><tr><td>3</td><td>7</td></tr><tr><td></td><td></td></tr></table>		ID склада [PK] integer	1	2	2	6	3	7			Результат содержит несколько записей
	ID склада [PK] integer											
1	2											
2	6											
3	7											

6. Найти десять самых популярных товаров и десять самых не раскупаемых товаров указанного магазина за период

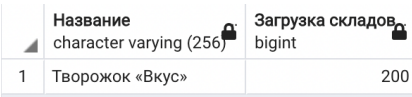
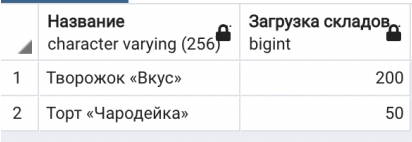
Запрос	Демонстрация результата	Комментарий									
<pre>INNER JOIN "Товар" ON "Товар"."Бар-код" = "Заказ на поставку"."Бар-код" GROUP BY "Название", "ID магазина", "Дата поставки" HAVING "ID магазина" = '3' AND ("Дата поставки" BETWEEN '2022-02-03' AND '2022-04-03') ORDER BY "Общее количество" DESC LIMIT 10) UNION (SELECT "Название", SUM("Количество") AS "Общее</pre>	<table border="1"> <thead> <tr> <th></th><th>Название character varying (256)</th><th>Общее количество bigint</th></tr> </thead> <tbody> <tr> <td>1</td><td>Гренки «Корочка»</td><td>40</td></tr> <tr> <td>2</td><td>Яблоки «Голден»</td><td>90</td></tr> </tbody> </table>		Название character varying (256)	Общее количество bigint	1	Гренки «Корочка»	40	2	Яблоки «Голден»	90	<p>Результат содержит несколько записей.</p> <p>Примечание: так, как база данных небольшая выборка получилась маленькая, но исходя из списка: первые 10 элементов списка - самые непопулярные, вторые 10 - самые популярные</p>
	Название character varying (256)	Общее количество bigint									
1	Гренки «Корочка»	40									
2	Яблоки «Голден»	90									

<p>количество" FROM "Заказ на поставку"</p> <p>INNER JOIN "Товар" ON "Товар"."Бар-код" = "Заказ на поставку"."Бар-код"</p> <p>GROUP BY</p> <p>"Название", "ID магазина", "Дата поставки"</p> <p>HAVING "ID магазина" = '3'</p> <p>AND ("Дата поставки" BETWEEN '2022-02-03' AND '2022-04-03')</p> <p>ORDER BY "Общее количество" ASC LIMIT 10)</p>		
--	--	--

7. Сформировать диаграмму распределения заказываемых товарных позиций по времени

Запрос	Демонстрация результата	Комментарий																														
<p>SELECT</p> <p>DATE_TRUNC('day', "Дата поставки")</p> <p>AS "Распределение заказываемых товаров",</p> <p>COUNT("ID заказа") AS count</p> <p>FROM "Заказ на поставку"</p> <p>GROUP BY DATE_TRUNC('day', "Дата поставки»);</p>	 <table border="1"> <thead> <tr> <th></th> <th>Распределение заказываемых товаров timestamp with time zone</th> <th>count bigint</th> </tr> </thead> <tbody> <tr><td>1</td><td>2022-07-03 00:00:00+03</td><td>1</td></tr> <tr><td>2</td><td>2022-10-03 00:00:00+03</td><td>1</td></tr> <tr><td>3</td><td>2022-05-03 00:00:00+03</td><td>2</td></tr> <tr><td>4</td><td>2022-12-03 00:00:00+03</td><td>1</td></tr> <tr><td>5</td><td>2022-02-03 00:00:00+03</td><td>1</td></tr> <tr><td>6</td><td>2022-04-03 00:00:00+03</td><td>1</td></tr> <tr><td>7</td><td>2022-11-03 00:00:00+03</td><td>1</td></tr> <tr><td>8</td><td>2022-06-03 00:00:00+03</td><td>1</td></tr> <tr><td>9</td><td>2022-03-03 00:00:00+03</td><td>1</td></tr> </tbody> </table>		Распределение заказываемых товаров timestamp with time zone	count bigint	1	2022-07-03 00:00:00+03	1	2	2022-10-03 00:00:00+03	1	3	2022-05-03 00:00:00+03	2	4	2022-12-03 00:00:00+03	1	5	2022-02-03 00:00:00+03	1	6	2022-04-03 00:00:00+03	1	7	2022-11-03 00:00:00+03	1	8	2022-06-03 00:00:00+03	1	9	2022-03-03 00:00:00+03	1	<p>Результат содержит диаграмму, в которой отображено какое количество товарных позиций было заказано в определенную дату</p>
	Распределение заказываемых товаров timestamp with time zone	count bigint																														
1	2022-07-03 00:00:00+03	1																														
2	2022-10-03 00:00:00+03	1																														
3	2022-05-03 00:00:00+03	2																														
4	2022-12-03 00:00:00+03	1																														
5	2022-02-03 00:00:00+03	1																														
6	2022-04-03 00:00:00+03	1																														
7	2022-11-03 00:00:00+03	1																														
8	2022-06-03 00:00:00+03	1																														
9	2022-03-03 00:00:00+03	1																														

8. Показать Загрузку складских площадей заказываемыми товарами на период.

Запрос	Демонстрация результата	Комментарий									
<pre>SELECT "Название", SUM("Объем страхового запаса") AS "Загрузка складов" FROM "Товар" INNER JOIN "Склад" ON "Склад"."Бар-код" = "Товар"."Бар-код" INNER JOIN "Заказ на поставку" ON "Заказ на поставку"."Бар-код" = "Товар"."Бар-код" GROUP BY "Название", "ID заказа", "Дата поставки", "Товар"."Бар-код", "Склад"."ID склада" HAVING "Товар"."Бар-код" = '13457869' AND ("Дата поставки" BETWEEN '2022-02-02' AND '2022-04-04')</pre>	 <table border="1"> <thead> <tr> <th></th><th>Название character varying (256)</th><th>Загрузка складов bigint</th></tr> </thead> <tbody> <tr> <td>1</td><td>Творожок «Вкус»</td><td>200</td></tr> </tbody> </table>		Название character varying (256)	Загрузка складов bigint	1	Творожок «Вкус»	200	Результат содержит одну запись			
	Название character varying (256)	Загрузка складов bigint									
1	Творожок «Вкус»	200									
<pre>SELECT "Название", SUM("Объем страхового запаса") AS "Загрузка складов" FROM "Товар"</pre>	 <table border="1"> <thead> <tr> <th></th><th>Название character varying (256)</th><th>Загрузка складов bigint</th></tr> </thead> <tbody> <tr> <td>1</td><td>Творожок «Вкус»</td><td>200</td></tr> <tr> <td>2</td><td>Торт «Чародейка»</td><td>50</td></tr> </tbody> </table>		Название character varying (256)	Загрузка складов bigint	1	Творожок «Вкус»	200	2	Торт «Чародейка»	50	Результат содержит несколько записей
	Название character varying (256)	Загрузка складов bigint									
1	Творожок «Вкус»	200									
2	Торт «Чародейка»	50									

<pre> INNER JOIN "Склад" ON "Склад"."Бар-код" = "Товар"."Бар-код" INNER JOIN "Заказ на поставку" ON "Заказ на поставку"."Бар-код" = "Товар"."Бар-код" GROUP BY "Название", "ID заказа", "Дата поставки", "Товар"."Бар-код", "Склад"."ID склада" HAVING ("Товар"."Бар-код" = '13457869' OR "Товар"."Бар-код" = '67946324') AND ("Дата поставки" BETWEEN '2021-02-02' AND '2023-04-04') </pre>		
--	--	--

Создание хранимой процедуры

CREATE OR REPLACE FUNCTION "добавить магазин" (shop_name text,
address text, volume int, shop_type text)

RETURNS TABLE("ID магазина" int, "ID Тип магазина" int)

language plpgsql AS \$\$

DECLARE shop_type_code integer;

DECLARE shop_code integer;

BEGIN

IF NOT EXISTS (SELECT 1 FROM "Тип магазина" WHERE "Тип
магазина" = shop_type) THEN

-- Если такой тип магазина не существует

shop_type_code := (SELECT COUNT("ID типа магазина") from "Тип
магазина");

INSERT INTO "Тип магазина"("ID типа магазина", "Тип магазина")
VALUES (shop_type_code, shop_type);

```

ELSE
-- Если такой тип магазина существует
    shop_type_code := (SELECT "ID типа магазина" from "Тип магазина"
where "Тип магазина" = shop_type);
END IF;

IF NOT EXISTS (SELECT 1 FROM "Магазин" WHERE "Название" =
shop_name) THEN
-- Если такой магазин не существует
    shop_code := (SELECT COUNT("ID типа магазина") from "Магазин");
ELSE
-- Если такой магазин существует
    shop_code := (SELECT "ID типа магазина" from "Магазин" WHERE
"Название" = shop_name);
END IF;

INSERT INTO "Магазин"("ID магазина", "ID типа магазина", "Название",
"Адрес", "Макс. объем заказа группы товаров")
VALUES (shop_code, shop_type_code, shop_name, address,
volume)
ON CONFLICT("Название") DO UPDATE
SET "ID типа магазина" = shop_type_code, "Название" = shop_name,
"Макс. объем заказа группы товаров" = volume;
RETURN QUERY
SELECT shop_code, shop_type_code;
END
$$;

```

Проверка:

```

SELECT * FROM "добавить магазин"('Восьмерочка', 'Парк победы', 800,
'Типермолл')

```

Добавлено новое значение

	ID магазина integer	ID Тип магазина integer
1	11	0

SELECT * FROM "добавить магазин"('Семерочка', 'Парк культуры', 200, 'Типермолл')

Значение уже находящееся в таблице перезаписано

	ID магазина integer	ID Тип магазина integer
1	0	0

Создание триггера обработки изменений в справочнике

"Справочник единиц измерения":

```
CREATE TABLE trig(
    operation      char(1) NOT NULL,
    time           timestamp NOT NULL,
    userid         text NOT NULL,
    issue          text NOT NULL
);
```

```
CREATE OR REPLACE FUNCTION process_trig() RETURNS TRIGGER AS
$trig$
```

```
    BEGIN
```

```
        IF (TG_OP = 'DELETE') THEN
```

```
            INSERT INTO trig SELECT 'D', now(), user, CAST(OLD.* AS text);
```

```
            RETURN OLD;
```

```
        ELSIF (TG_OP = 'UPDATE') THEN
```

```
            INSERT INTO trig SELECT 'U', now(), user, CAST(NEW.* AS text);
```

```
            RETURN NEW;
```

```
        END IF;
```

```
        RETURN NULL;
```

```
    END;
```



```
$trig$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trig  
AFTER UPDATE OR DELETE ON "Справочник единиц измерения"  
FOR EACH ROW EXECUTE PROCEDURE process_trig();
```

Пример записи в другой таблице при изменении записей в справочнике:

	operation character (1)	time timestamp without time zone	userid text	issue text
1	U	2022-03-26 01:48:02.693918	postgres	(6,паллет)

Аналогично для "Справочника товарных групп":

```
CREATE TABLE trig2(  
    operation      char(1) NOT NULL,  
    time           timestamp NOT NULL,  
    userid         text NOT NULL,  
    issue          text NOT NULL  
);
```

```
CREATE OR REPLACE FUNCTION process_trig2() RETURNS TRIGGER  
AS $trig2$  
BEGIN  
    IF (TG_OP = 'DELETE') THEN  
        INSERT INTO trig2 SELECT 'D', now(), user, CAST(OLD.* AS text);  
        RETURN OLD;  
    ELSIF (TG_OP = 'UPDATE') THEN  
        INSERT INTO trig2 SELECT 'U', now(), user, CAST(NEW.* AS text);  
        RETURN NEW;  
    END IF;  
    RETURN NULL;  
END;  
$trig2$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trig2  
AFTER UPDATE OR DELETE ON "Справочник товарных групп"  
FOR EACH ROW EXECUTE PROCEDURE process_trig2();
```

Пример записи в другой таблице при изменении записей в справочнике:

	operation character (1)	time timestamp without time zone	userid text	issue text
1	U	2022-03-26 01:58:37.83074	postgres	(8,Другие)

Создание пользователей и РПД

Создание роли “кладовщик”:

```
CREATE ROLE "кладовщик" WITH PASSWORD 'password1';
REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA "public"
FROM "кладовщик";
GRANT SELECT, UPDATE, INSERT, DELETE
ON TABLE
"Склад"
TO "кладовщик";
```

Проверка корректности работы:

```
SET ROLE "кладовщик";
SELECT * FROM "Склад"
```

8	SET ROLE "кладовщик";
9	SELECT * FROM "Склад"

	ID склада [PK] integer	ID типа склада integer	ID товарной группы integer	Бар-код integer
1	1	3	1	13457869
2	2	5	3	67345678
3	3	4	8	67508976
4	4	3	1	34678654
5	5	2	2	56324356
6	6	3	3	54324589
7	7	3	2	67548967
8	8	3	4	67946324
9	9	3	4	45327863

```
SET ROLE "кладовщик";  
SELECT * FROM "Магазин"
```

```
8 SET ROLE "кладовщик";  
9 SELECT * FROM "Магазин"
```

Data Output	Explain	Messages	Notifications
-------------	---------	----------	---------------

ERROR: permission denied for table Магазин SQL state: 42501			
--	--	--	--

Создание роли "логист":

```
CREATE ROLE "логист" WITH PASSWORD 'password2';  
REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA "public"  
FROM "логист";  
GRANT SELECT, UPDATE, INSERT, DELETE  
ON TABLE  
    "Заказ на поставку",  
    "Склад",  
    "Магазин"  
TO "логист";
```

Проверка корректности работы:

```
SET ROLE "логист";  
SELECT * FROM "Магазин"
```

```
20 SET ROLE "логист";
21 SELECT * FROM "Магазин"
```

Data Output Explain Messages Notifications

	ID магазина [PK] integer	ID типа магазина integer	Название character varying (256)	Адрес character varying (256)	Макс. объем заказа группы товаров integer
1	1	1	Подсолнух	Солнечная, 56	400
2	2	3	Ромашка	Бульвар Мира, 4к1	500
3	3	2	Василек	Проспект Мира, 67	350
4	4	2	Роза	Республиканская, 78	400
5	5	1	Лепесток	Ленина, 45к1	450
6	6	3	Гвоздика	Проспект Маршала Катукова, 56	600
7	7	2	Люттик	Пресненская набережная, 1	350
8	8	1	Лилия	Авиационная, 79	550

```
SET ROLE "логист";
SELECT * FROM "Тип магазина"
```

```
20 SET ROLE "логист";
21 SELECT * FROM "Тип магазина"
```

Data Output Explain Messages Notifications

ERROR: permission denied for table Тип магазина
SQL state: 42501

Создание роли "главный на смене":

```
CREATE ROLE "главный на смене" WITH PASSWORD 'password3';
REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA "public"
FROM "главный на смене";
GRANT SELECT, UPDATE, INSERT, DELETE
ON TABLE
  "Заказ на поставку",
  "Магазин"
TO "главный на смене";
```

Проверка корректности работы:

```
SET ROLE "главный на смене";
SELECT * FROM "Заказ на поставку"
```

```

33 SET ROLE "главный на смене";
34 SELECT * FROM "Заказ на поставку"

```

Data Output Explain Messages Notifications

	ID заказа [PK] integer	ID магазина integer	Бар-код integer	Количество integer	Дата поставки date
1	2	4	67548967	40	2022-04-03
2	3	3	13457869	70	2022-03-03
3	4	2	67345678	30	2022-05-03
4	5	7	67946324	40	2022-07-03
5	6	8	34678654	150	2022-10-03
6	7	6	67508976	50	2022-11-03
7	8	9	56324356	20	2022-06-03
8	9	10	45327863	60	2022-05-03
9	10	1	54324589	90	2022-12-03

```

SET ROLE "главный на смене";
SELECT * FROM "Склад"

```

```

33 SET ROLE "главный на смене";
34 SELECT * FROM "Склад"

```

Data Output Explain Messages Notifications

```

ERROR: permission denied for table Склад
SQL state: 42501

```

Создание представлений для пользователей

```

CREATE VIEW "предприятие"."Изделия и спецификации"
AS
SELECT "название изделия", "описание спецификации"
FROM "предприятие"."изделия"
INNER JOIN "предприятие"."спецификации"

```

```
ON "изделия"."код спецификации" = "спецификации"."код спецификации";
```

1. Кладовщик

Найти магазин по определенному заказу

```
CREATE VIEW "Магазин и заказ"  
AS
```

```
SELECT "Название" FROM "Заказ на поставку"
```

```
INNER JOIN "Магазин" ON "Магазин"."ID магазина" = "Заказ на поставку"."ID магазина"
```

```
WHERE "ID заказа" = '2'
```

```
SET ROLE "кладовщик";
```

```
SELECT * FROM "Магазин и заказ"
```

17	SET ROLE "кладовщик";
18	SELECT * FROM "Магазин и заказ"
<div>Data Output Explain Messages Notifications</div>	
	<div>Название</div> <div>character varying (256)</div>
1	Роза

2. Сотрудник отдела логистики

Получить адреса магазинов сегодняшней поставки

```
CREATE VIEW "Адреса и поставки"  
AS
```

```
SELECT "Адрес" FROM "Магазин"
```

```
INNER JOIN "Заказ на поставку" ON "Заказ на поставку"."ID  
магазина" = "Магазин"."ID магазина"
```

```
WHERE "Дата поставки" = '2022-02-03'
```

```
SET ROLE "логист";
```

```
SELECT * FROM "Адреса и поставки"
```

```
30 SET ROLE "логист";
31 SELECT * FROM "Адреса и поставки"
```

Data Output Explain Messages Notifications

	Адрес character varying (256)	
1	Ленина, 45к1	

3. Главный на смене

Найти даты поставок в определенный магазин

```
CREATE VIEW "Поставки и магазин"
AS
```

```
SELECT "Дата поставки" FROM "Заказ на поставку"
```

```
INNER JOIN "Магазин" ON "Магазин"."ID магазина" = "Заказ на
поставку"."ID магазина"
```

```
WHERE "Название" = 'Лютик'
```

```
SET ROLE "главный на смене";
```

```
SELECT * FROM "Поставки и магазин"
```

```
40 SET ROLE "главный на смене";
41 SELECT * FROM "Поставки и магазин"
```

Data Output Explain Messages Notifications

	Дата поставки date	
1	2022-07-03	

Настройка шифрования атрибута

```
CREATE EXTENSION IF NOT EXISTS pgcrypto;
ALTER TABLE "Справочник товарных групп" ADD COLUMN "crypto"
bytea;
UPDATE "Справочник товарных групп" SET "crypto" = encrypt("Тип
товарной группы"::bytea, 'password'::bytea, 'aes');
```

```
SELECT "ID товарной группы", "Тип товарной группы", "crypto"::text
FROM "Справочник товарных групп";
```

Data Output	Explain	Messages	Notifications
ID товарной группы [PK] smallint	Тип товарной группы character varying (64)	crypto text	
1	1 Молочно-масляные	\xab9fdaff15e90bcd4efde690ce8c64c2f3df38a288ab05476660eddabba1e91	
2	2 Хлебобулочные изделия	\x6bde17a64d6827f6b533bfc2403fd4d74eb226b7acfc382f2d874b716460351e30e80fce99d6c62842b0b15f7baf73	
3	3 Плоды и овощи	\x734675f6c2276ab019c8f99ce79b5904a30e785814a64353c10a3d57ecc159b8	
4	4 Кондитерские	\x9378a81b3ed9b6592fd00c7c7f324a2eab839b621d2a24fe1d0bdd878dd0a3b4	
5	5 Безалкогольные напитки	\x21cfcce4130a78704ea73109b4267093c8db40cbfaa140c81484e81a7003899447ae52279ccf31af60714a53e77f1	
6	6 Мясные и колбасные изделия	\x32e9bcb6f1ad02f1b34351691bde60b783e193aef68e18e21d6b9d35099e71f46955bcd49bb78667a450fd281986f	
7	7 Табачные	\xf95a018ddf4dac7911aa54b0e55452b8b6154b61b544aa448e59596254e21046	
8	8 Другие	\x25a7b45865501197d7d4d3138c322d07	

```
CREATE OR REPLACE VIEW "Crypto" AS
SELECT "ID товарной группы", "Тип товарной группы",
convert_from(decrypt("crypto", 'password'::bytea, 'aes'),'SQL_ASCII')
FROM "Справочник товарных групп";
SELECT * FROM "Crypto";
```

	ID товарной группы smallint	Тип товарной группы character varying (64)	convert_from text
1	1	Молочно-масляные	Молочно-масляные
2	2	Хлебобулочные изделия	Хлебобулочные изделия
3	3	Плоды и овощи	Плоды и овощи
4	4	Кондитерские	Кондитерские
5	5	Безалкогольные напитки	Безалкогольные напитки
6	6	Мясные и колбасные изделия	Мясные и колбасные изделия
7	7	Табачные	Табачные
8	8	Другие	Другие

```
SET ROLE "кладовщик";
SELECT * FROM "Crypto";
```



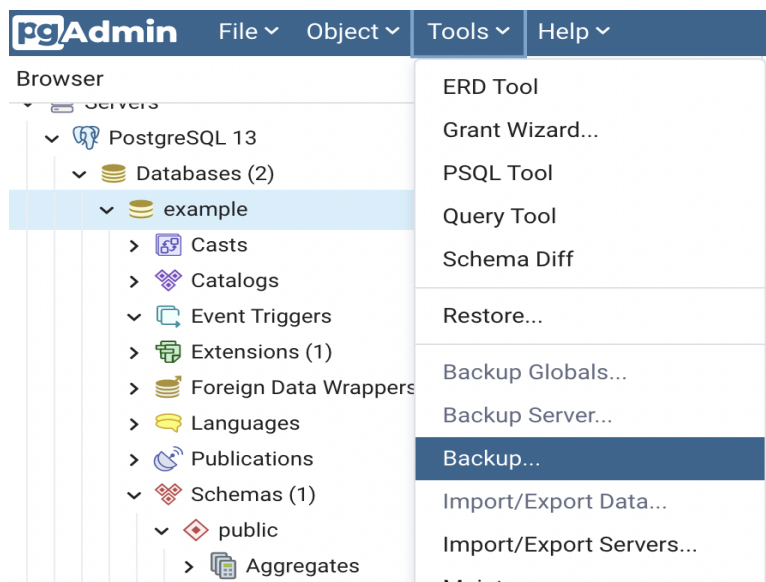
```
12 SET ROLE "кладовщик";
13 SELECT * FROM "Crypto";
```

Data Output Explain Messages Notifications

ERROR: permission denied for view Crypto
SQL state: 42501

Резервное копирование данных

Для того, чтобы сделать резервное копирование данных необходимо выбрать необходимую БД, затем в Tools выбрать Backup



В появившемся окне указать указать Filename

Backup (Database: example)

General

Data/Objects

Options

Filename

Format

Custom

Compression ratio

Encoding

Select an item...

Number of jobs

Role name

Select an item...

Please provide a filename.

Close

Reset

Backup

После чего нажать на Backup

Backup (Database: example)

General

Data/Objects

Options

Filename

/Users/dianarymkulova/Documents/example

Format

Custom

Compression ratio

Encoding

Select an item...

Number of jobs

Role name

Select an item...

Close

Reset

Backup

Спустя время появится уведомление об успешном завершении копирования

Backing up an object on the server

Backing up an object on the server 'PostgreSQL 13 (localhost:5432)' from database 'example'

Sat Mar 26 2022 03:49:23 GMT+0300 (Москва, стандартное время)

0.79 seconds

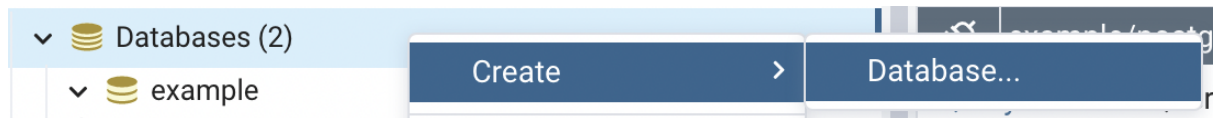
More details...

Stop Process

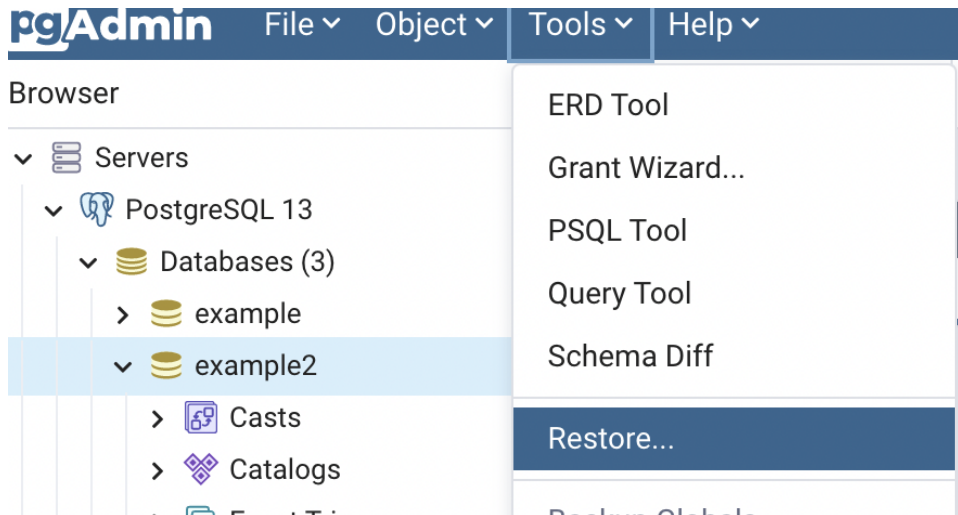
✓

Successfully completed.

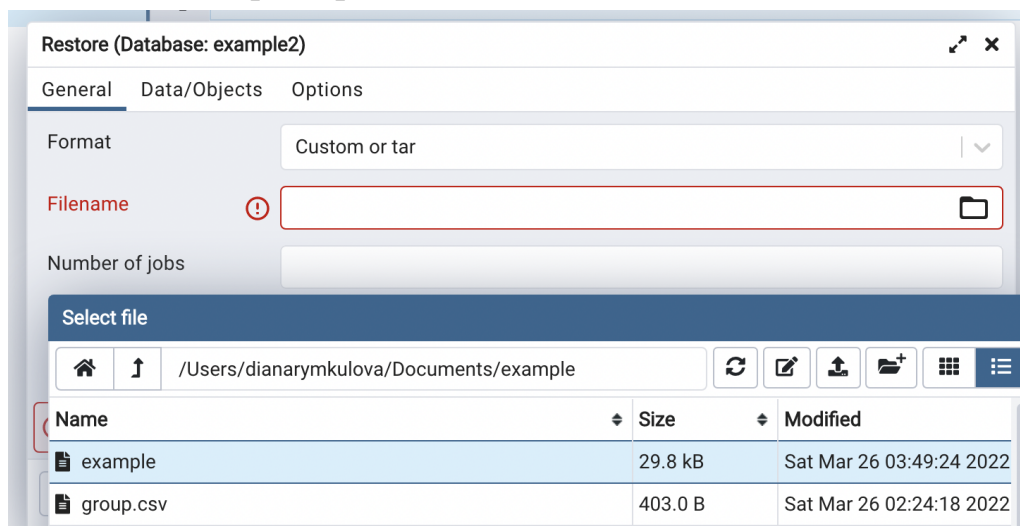
Для того, чтобы восстановить БД необходимо создать новую БД (либо обновить старую, если она сохранилась)



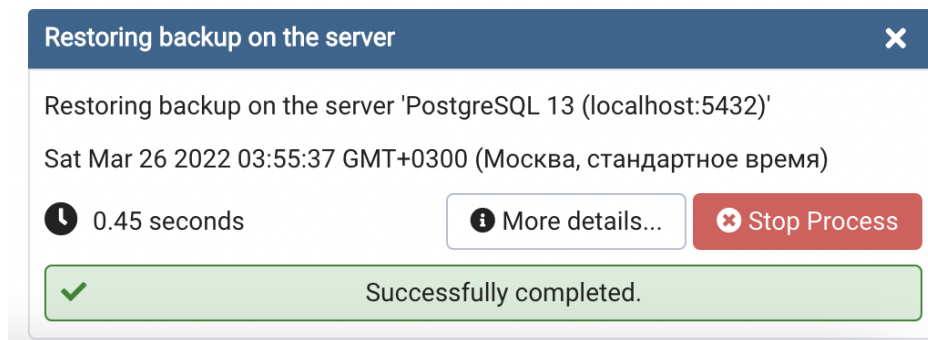
Выбрать Restore



После чего выбрать файл дампа



Нажать Restore, после успешного восстановления появится уведомление об этом, и можно приступать к дальнейшей работе



Приложение 1.

CREATE TABLE "Справочник товарных групп"

```
(  
  "ID товарной группы" smallint PRIMARY KEY,  
  "Тип товарной группы" varchar(64) UNIQUE NOT NULL  
);
```

CREATE TABLE "Справочник единиц измерения"

```
(  
  "ID единицы измерения" integer PRIMARY KEY,  
  "Тип единицы измерения" varchar(256) UNIQUE NOT NULL  
);
```

CREATE TABLE "Тип склада"

```
(  
  "ID типа склада" integer PRIMARY KEY,  
  "Тип склада" varchar(256) UNIQUE NOT NULL  
);
```

CREATE TABLE "Тип магазина"

```
(  
  "ID типа магазина" integer PRIMARY KEY,  
  "Тип магазина" varchar(256) UNIQUE NOT NULL  
);
```

CREATE TABLE "Товар"

```
(  
  "Бар-код" integer PRIMARY KEY,  
  "Название" varchar(256) UNIQUE NOT NULL,  
  "ID товарной группы" integer NOT NULL REFERENCES "Справочник  
товарных групп" ON DELETE RESTRICT ON UPDATE CASCADE,  
  "ID единицы измерения" integer NOT NULL REFERENCES "Справочник  
единиц измерения" ON DELETE RESTRICT ON UPDATE CASCADE,
```

```

    "Объем страхового запаса" integer NOT NULL DEFAULT 1 CHECK
("Объем страхового запаса" > 0),
    "Цена" integer NOT NULL DEFAULT 1 CHECK ("Цена" > 0)
);
CREATE TABLE "Склад"
(
    "ID склада" integer PRIMARY KEY,
    "ID типа склада" integer NOT NULL REFERENCES "Тип склада" ON
DELETE RESTRICT ON UPDATE CASCADE,
    "ID товарной группы" integer NOT NULL REFERENCES "Справочник
товарных групп" ON DELETE RESTRICT ON UPDATE CASCADE,
    "Бар-код" integer NOT NULL REFERENCES "Товар" ON DELETE
RESTRICT ON UPDATE CASCADE
);
CREATE TABLE "Магазин"
(
    "ID магазина" integer PRIMARY KEY,
    "ID типа магазина" integer NOT NULL REFERENCES "Тип магазина" ON
DELETE RESTRICT ON UPDATE CASCADE,
    "Название" varchar(256) UNIQUE NOT NULL,
    "Адрес" varchar(256) UNIQUE NOT NULL,
    "Макс. объем заказа группы товаров" integer NOT NULL DEFAULT 1
CHECK ("Макс. объем заказа группы товаров" > 0)
);
CREATE TABLE "Заказ на поставку"
(
    "ID заказа" integer PRIMARY KEY,
    "ID магазина" integer NOT NULL REFERENCES "Магазин" ON DELETE
RESTRICT ON UPDATE CASCADE,
    "Бар-код" integer NOT NULL REFERENCES "Товар" ON DELETE
RESTRICT ON UPDATE CASCADE,
    "Количество" integer NOT NULL DEFAULT 1 CHECK ("Количество" >
0),
    "Дата поставки" date NOT NULL
);
CREATE TABLE "Заказ на поставку_Склад"
(

```

```
"ID заказа" integer REFERENCES "Заказ на поставку" ON DELETE  
RESTRICT ON UPDATE CASCADE,  
"ID склада" integer REFERENCES "Склад" ON DELETE RESTRICT ON  
UPDATE CASCADE,  
PRIMARY KEY ("ID заказа", "ID склада")  
);
```