МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 6

**Однотабличные запросы в SQL. Агрегатные функции.**

**Вариант 12.**

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Лаптев

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Я.С. Сергеева

Лабораторная работа защищена

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2021

Оглавление

[1. Словарь данных 3](#_Toc86751786)

[2. ER-диаграмма 3](#_Toc86751787)

[3. Информация о БД и содержащихся в таблицах данных 3](#_Toc86751788)

[4. Основные запросы и их результаты 4](#_Toc86751789)

[5. Приложение 10](#_Toc86751790)

# Словарь данных

Для первого задания словарем данных будет являться непосредственно таблица «Товары».

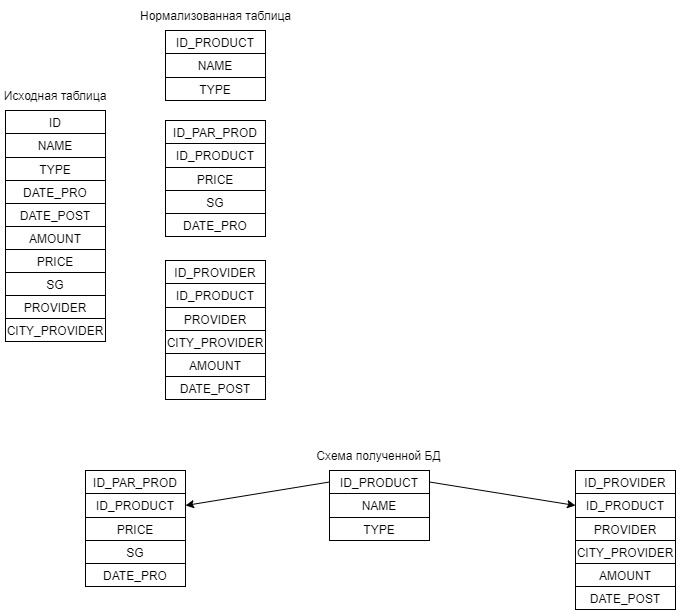
Изображение выглядит как текст

Автоматически созданное описание

*Схема 1. Таблица «Товары».*

В этой таблице каждый столбец является характеристикой определенного товара и не связан с какими-либо другими таблицами.

Для третьего задания словарем данных будет три взаимосвязанных таблицы «Товары», «Поставщики», «Параметры товаров».



*Схема 2. Взаимосвязь таблиц «Товары», «Поставщики», «Параметры товаров».*

Здесь основной таблицей будет являться таблица «Товары», от которой зависят, в свою очередь, таблицы «Поставщики» и «Параметры товаров». Как видно из схемы, последние две таблицы не зависят друг от друга. Соответственно, в таблице «Товары» хранится основная информация о товарах, в таблице «Поставщики» хранится информация, затрагивающая поставщиков товаров и их взаимодействия с товаром, а в таблице «Параметры товаров» хранится более конкретная информация о самих товарах.

# ER-диаграмма

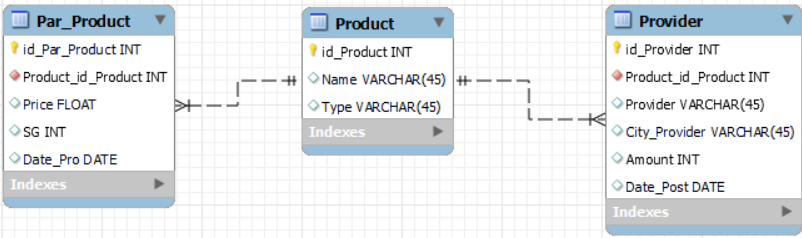
Для таблицы «Товары», предложенной в первом задании, ER-диаграмма выглядит следующим образом:

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 1. ER-диаграмма таблицы «Товары».*

Для нормализованной схемы БД, полученной в третьем задании, ER-диаграмма будет выглядеть так:



*Рис. 2. ER-диаграмма нормализованной схемы БД.*

# Информация о БД и содержащихся в таблицах данных

В первом задании БД представляет собой одну таблицу «Товары», в которой хранится набор данных для следующих столбцов: id\_Product, Name, Type, Date\_pro, Date\_post, Amount, Price, SG, Provider, City\_provider.

В третьем задании нормализованная схема БД состоит из трех связанных друг с другом таблиц.

Для таблицы Product определены следующие столбцы: id\_Product, Name, Type.

Для таблицы Par\_Product определены столбцы: id\_Par\_Product, Product\_id\_Product (foreign key), Price, SG, Date\_Pro.

Для таблицы Provider определены столбцы: id\_Provider, Product\_id\_Product (foreign key), Provider, City\_provider, Amount, Date\_Post.

Тестовые данные для заполнения всех таблиц приведены в приложении.

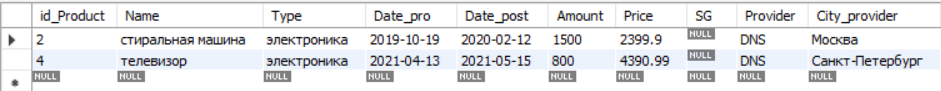
# Основные запросы и их результаты

И к первому, и к третьему заданию запросы были одинаковыми:

1. Вывести все товары в ценовом диапазоне от 850 до 5000, которые поставляет фирма "DNS".
2. Найти товары, которые поставленные более полугода назад.
3. Вывести список типов товаров и кол-во товаров данного типа.
4. Вывести название товара, поставщика и стоимость тех товаров, у которых истек срок годности.
5. Вывести все товары поставщиков из города Новосибирск, если дата производства у них позже 10.07.2021.
6. Вывести максимальную стоимость товаров определенного типа.
7. Вывести товары типа "молоко", у которого стоимость ниже среднего.
8. Вывести название товара и его тип для тех товаров, у которых между датой производства и датой поставки не прошло 10 дней.
9. Вывести список товаров поставленных осенью 2021 года и у которых название состоит из 5 букв.
10. Вывести первые 3 буквы названия товара, цену товара и день недели, в которую этот товар был поставлен.

Запросы и результаты запросов к первому заданию:

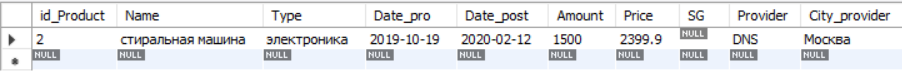
select \* from product where Price>=850 and Price<=5000 and Provider='DNS'



*Рис. 3. Результат запроса.*

select \* from product

where date\_format(current\_date, '%m.%d')-date\_format(Date\_post, '%m.%d')>6.0



*Рис. 4. Результат запроса.*

select Name, Amount from product where Type='молочная продукция'

Изображение выглядит как текст, стол

Автоматически созданное описание

*Рис. 5. Результат запроса.*

select Name, Provider, Price from product where current\_date-Date\_pro>SG

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 6. Результат запроса.*

select \* from product where City\_provider='Новосибирск' and Date\_pro>'2021-07-10'



*Рис. 7. Результат запроса.*

select max(Price) from product where Type='электроника'



*Рис. 8. Результат запроса.*

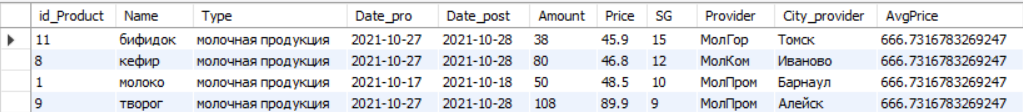
select \*

from (select \*,

avg(Price) over() AvgPrice

from product) as APrice

where Type='молочная продукция' and Price < AvgPrice



*Рис. 9. Результат запроса.*

select Name, Type from product

where date\_format(Date\_post, '%y%m%d') - date\_format(Date\_pro, '%y%m%d')<10

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 10. Результат запроса.*

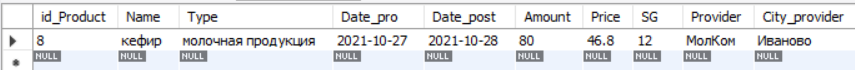
select \* from product

where (date\_format(Date\_post, '%m.%Y')='09.2021'

or date\_format(Date\_post, '%m.%Y')='10.2021'

or date\_format(Date\_post, '%m.%Y')='11.2021')

and char\_length(Name)=5



*Рис. 11. Результат запроса.*

select left(Name, 3), Price, dayname(Date\_pro) from product

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 12. Результат запроса.*

Запросы и результаты запросов к третьему заданию:

select id\_Product, Name, Type, Price, SG, Date\_Pro, Date\_Post, Amount, Provider, City\_Provider

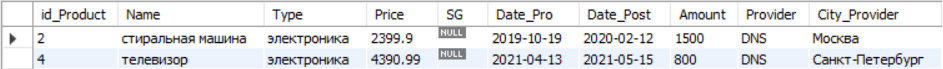
from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product

and Price>=850 and Price<=5000

and Provider='DNS'



*Рис. 13. Результат запроса.*

select id\_Product, Name, Type, Price, SG, Date\_Pro, Date\_Post, Amount, Provider, City\_Provider

from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product

and date\_format(current\_date, '%m.%d')-date\_format(Date\_post, '%m.%d')>6.0

*Рис. 14. Результат запроса.*

select Name, Amount from product, provider

where product.id\_Product=provider.Product\_id\_Product and Type='молочная продукция'

Изображение выглядит как текст, стол

Автоматически созданное описание

*Рис. 15. Результат запроса.*

select Name, Provider, Price

from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product

and current\_date-Date\_pro>SG

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 16. Результат запроса.*

select id\_Product, Name, Type, Price, SG, Date\_Pro, Provider, City\_Provider, Amount, Date\_Post

from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product

and City\_provider='Новосибирск' and Date\_pro>'2021-07-10'



*Рис. 17. Результат запроса.*

select max(Price)

from product, par\_product where Type='электроника'



*Рис. 18. Результат запроса.*

select \*

from (select id\_Product, Name, Type, Price, SG, Date\_Pro, Provider, City\_Provider, Amount, Date\_Post,

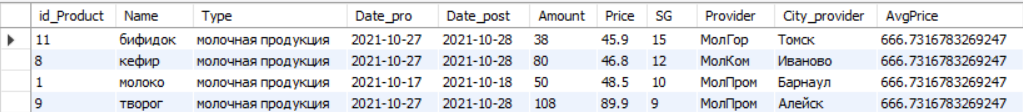
avg(Price) over() AvgPrice

from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product) as APrice

where Type='молочная продукция' and Price < AvgPrice



*Рис. 19. Результат запроса.*

select Name, Type

from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product

and date\_format(Date\_post, '%y%m%d') - date\_format(Date\_pro, '%y%m%d')<10

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 20. Результат запроса.*

select id\_Product, Name, Type, Price, SG, Date\_Pro, Provider, City\_Provider, Amount, Date\_Post

from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product

and (date\_format(Date\_post, '%m.%Y')='09.2021'

or date\_format(Date\_post, '%m.%Y')='10.2021'

or date\_format(Date\_post, '%m.%Y')='11.2021')

and char\_length(Name)=5



*Рис. 21. Результат запроса.*

select left(Name, 3), Price, dayname(Date\_Pro)

from product, provider, par\_product

where product.id\_Product=provider.Product\_id\_Product

and product.id\_Product=par\_product.Product\_id\_Product

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 22. Результат запроса.*

# Приложение

Скрипт для формирования структуры БД и наполнения таблицы «Товары» тестовыми данными:

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema products

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `products` DEFAULT CHARACTER SET utf8 ;

USE `products` ;

-- -----------------------------------------------------

-- Table `products`.`Product`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `products`.`Product` (

`id\_Product` INT NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(45) NULL,

`Type` VARCHAR(45) NULL,

`Date\_pro` DATE NULL,

`Date\_post` DATE NULL,

`Amount` INT NULL,

`Price` FLOAT NULL,

`SG` INT NULL,

`Provider` VARCHAR(45) NULL,

`City\_provider` VARCHAR(45) NULL,

PRIMARY KEY (`id\_Product`),

UNIQUE INDEX `id\_Product\_UNIQUE` (`id\_Product` ASC) VISIBLE)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (1, 'молоко', 'молочная продукция', '2021-10-17 00:00:00', '2021-10-18 00:00:00', 50, 48.50, 10, 'МолПром', 'Барнаул');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (2, 'стиральная машина', 'электроника', '2019-10-19 00:00:00', '2020-02-12 00:00:00', 1500, 2399.90, NULL, 'DNS', 'Москва');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (3, 'пенталгин', 'лекарства', '2021-07-23 00:00:00', '2021-07-28 00:00:00', 15, 159.90, 950, 'ФармСтанд', 'Новосибирск');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (4, 'телевизор', 'электроника', '2021-04-13 00:00:00', '2021-05-15 00:00:00', 800, 4390.99, NULL, 'DNS', 'Санкт-Петербург');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (5, 'ветчина', 'мясные изделия', '2021-09-26 00:00:00', '2021-09-28 00:00:00', 23, 312.50, 30, 'МясКомбКем', 'Кемерово');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (6, 'анальгин', 'лекарства', '2021-07-08 00:00:00', '2021-07-13 00:00:00', 5, 90.90, 900, 'ФармФэйм', 'Новосибирск');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (7, 'аспирин', 'лекарства', '2021-06-20 00:00:00', '2021-06-25 00:00:00', 34, 85.60, 960, 'ХелфФарм', 'Владивосток');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (8, 'кефир', 'молочная продукция', '2021-10-27 00:00:00', '2021-10-28 00:00:00', 80, 46.80, 12, 'МолКом', 'Иваново');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (9, 'творог', 'молочная продукция', '2021-10-27 00:00:00', '2021-10-28 00:00:00', 108, 89.90, 9, 'МолПром', 'Барнаул');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (10, 'колбаса копченая', 'мясные изделия', '2021-10-19 00:00:00', '2021-10-21 00:00:00', 18, 78.90, 24, 'КрасМясКом', 'Краснодар');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (11, 'бифидок', 'молочная продукция', '2021-10-27 00:00:00', '2021-10-28 00:00:00', 38, 45.90, 15, 'МолГор', 'Томск');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE, DATE\_PRO, DATE\_POST, AMOUNT, PRICE, SG, PROVIDER, CITY\_PROVIDER) VALUES (12, 'торт', 'кондитерские изделия', '2021-10-14 00:00:00', '2021-10-15 00:00:00', 54, 250.99, 7, 'Вилли Вонка', 'Липецк');

COMMIT WORK;

Скрипт для формирования структуры БД и наполнения ее тестовыми данными для нормализованной схемы данных:

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema normal\_products

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `normal\_products` DEFAULT CHARACTER SET utf8 ;

USE `normal\_products` ;

-- -----------------------------------------------------

-- Table `normal\_products`.`Product`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `normal\_products`.`Product` (

`id\_Product` INT NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(45) NULL,

`Type` VARCHAR(45) NULL,

PRIMARY KEY (`id\_Product`),

UNIQUE INDEX `idProduct\_UNIQUE` (`id\_Product` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `normal\_products`.`Provider`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `normal\_products`.`Provider` (

`id\_Provider` INT NOT NULL AUTO\_INCREMENT,

`Product\_id\_Product` INT NOT NULL,

`Provider` VARCHAR(45) NULL,

`City\_Provider` VARCHAR(45) NULL,

`Amount` INT NULL,

`Date\_Post` DATE NULL,

PRIMARY KEY (`id\_Provider`),

UNIQUE INDEX `id\_Provider\_UNIQUE` (`id\_Provider` ASC) VISIBLE,

INDEX `fk\_Provider\_Product1\_idx` (`Product\_id\_Product` ASC) VISIBLE,

CONSTRAINT `fk\_Provider\_Product1`

FOREIGN KEY (`Product\_id\_Product`)

REFERENCES `normal\_products`.`Product` (`id\_Product`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `normal\_products`.`Par\_Product`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `normal\_products`.`Par\_Product` (

`id\_Par\_Product` INT NOT NULL AUTO\_INCREMENT,

`Product\_id\_Product` INT NOT NULL,

`Price` FLOAT NULL,

`SG` INT NULL,

`Date\_Pro` DATE NULL,

PRIMARY KEY (`id\_Par\_Product`),

UNIQUE INDEX `id\_Par\_Product\_UNIQUE` (`id\_Par\_Product` ASC) VISIBLE,

INDEX `fk\_Par\_Product\_Product\_idx` (`Product\_id\_Product` ASC) VISIBLE,

CONSTRAINT `fk\_Par\_Product\_Product`

FOREIGN KEY (`Product\_id\_Product`)

REFERENCES `normal\_products`.`Product` (`id\_Product`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (1, 'молоко', 'молочная продукция');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (2, 'стиральная машина', 'электроника');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (3, 'пенталгин', 'лекарства');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (4, 'телевизор', 'электроника');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (5, 'ветчина', 'мясные изделия');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (6, 'анальгин', 'лекарства');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (7, 'аспирин', 'лекарства');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (8, 'кефир', 'молочная продукция');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (9, 'творог', 'молочная продукция');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (10, 'колбаса копченая', 'мясные изделия');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (11, 'бифидок', 'молочная продукция');

INSERT INTO PRODUCT (ID\_PRODUCT, NAME, TYPE) VALUES (12, 'торт', 'кондитерские изделия');

COMMIT WORK;

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (1, 1, '2021-10-18 00:00:00', 50, 'МолПром', 'Барнаул');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (2, 2, '2020-02-12 00:00:00', 1500, 'DNS', 'Москва');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (3, 3, '2021-07-28 00:00:00', 15, 'ФармСтанд', 'Новосибирск');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (4, 4, '2021-05-15 00:00:00', 800, 'DNS', 'Санкт-Петербург');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (5, 5, '2021-09-28 00:00:00', 23, 'МясКомбКем', 'Кемерово');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (6, 6, '2021-07-13 00:00:00', 5, 'ФармФэйм', 'Новосибирск');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (7, 7, '2021-06-25 00:00:00', 34, 'ХелфФарм', 'Владивосток');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (8, 8, '2021-10-28 00:00:00', 80, 'МолКом', 'Иваново');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (9, 9, '2021-10-28 00:00:00', 108, 'МолПром', 'Алейск');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (10, 10, '2021-10-21 00:00:00', 18, 'КрасМясКом', 'Краснодар');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (11, 11, '2021-10-28 00:00:00', 38, 'МолГор', 'Томск');

INSERT INTO PROVIDER (ID\_PROVIDER, PRODUCT\_ID\_PRODUCT, DATE\_POST, AMOUNT, PROVIDER, CITY\_PROVIDER) VALUES (12, 12, '2021-10-15 00:00:00', 54, 'Вилли Вонка', 'Липецк');

COMMIT WORK;

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (1, 1, '2021-10-17 00:00:00', 48.50, 10);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (2, 2, '2019-10-19 00:00:00', 2399.90, NULL);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (3, 3, '2021-07-23 00:00:00', 159.90, 950);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (4, 4, '2021-04-13 00:00:00', 4390.99, NULL);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (5, 5, '2021-09-26 00:00:00', 312.50, 30);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (6, 6, '2021-07-08 00:00:00', 90.90, 900);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (7, 7, '2021-06-20 00:00:00', 85.60, 960);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (8, 8, '2021-10-27 00:00:00', 46.80, 12);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (9, 9, '2021-10-27 00:00:00', 89.90, 9);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (10, 10, '2021-10-19 00:00:00', 78.90, 24);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (11, 11, '2021-10-27 00:00:00', 45.90, 15);

INSERT INTO PAR\_PRODUCT (ID\_PAR\_PRODUCT, PRODUCT\_ID\_PRODUCT, DATE\_PRO, PRICE, SG) VALUES (12, 12, '2021-10-14 00:00:00', 250.99, 7);

COMMIT WORK;