## Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

## «Московский государственный технический университет имени Н.Э.Баумана

**(национальный исследовательский университет)» (МГТУ им.Н.Э.Баумана)**

**ФАКУЛЬТЕТ «Робототехника и комплексная автоматизация» КАФЕДРА «Системы автоматизированного проектирования»**

# ОТЧЁТ

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ**

# Вариант 1. Резервирование товаров

ПО ДИСЦИПЛИНЕ: «Базы данных»

Студент **РК6-48Б Кочура Д.И.**

(Фамилия И.О.)

Проверил **\_ \_ \_ \_**

(Подпись, дата) (Фамилия И.О.)

*2022г.*

## Содержание задания

***Вариант 1. Резервирование Товаров***

На складе хранятся *товары*, о которых известно шифр товара, его название, единица измерения, цена единицы, установленная поставщиком при доставке товара.

Товары со склада продаются *покупателям*, о которых известно уникальный номер, название, адрес, контактный телефон, банковские реквизиты (БИК, номер счета, название банка), дата заключения контракта на поставку товаров со склада.

Прежде чем покупатель сможет получить товар, он обязан его зарезервировать, сделав *заказ*. Заказ должен быть оплачен в течение 3-х дней. После оплаты заказа, товар отпускается со склада.

В заказе указывается уникальный номер заказа, номер покупателя, дата составления заказа, статус заказа, максимально допустимая дата оплаты и фактическая дата оплаты. Возможные значения статуса заказа:

1. Заказ составлен, но не оплачен
2. Заказ оплачен
3. Заказ просрочен и аннулирован

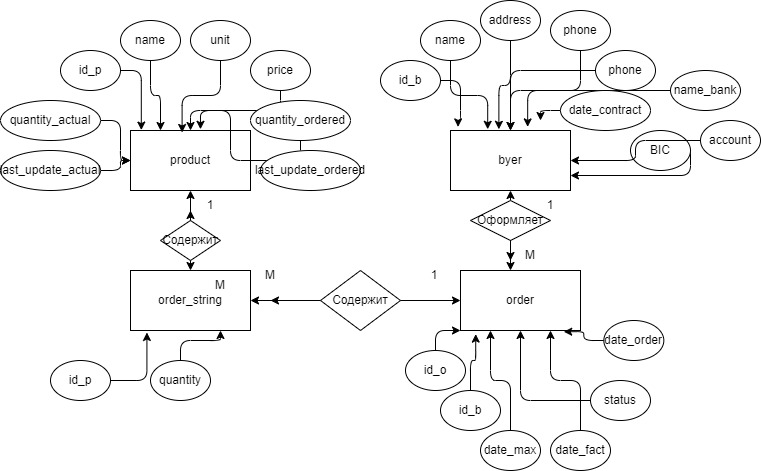
В одном заказе можно зарезервировать произвольное количество товаров. Для этого формируются *строки заказа.* Каждая строка содержит шифр товара и его количество.

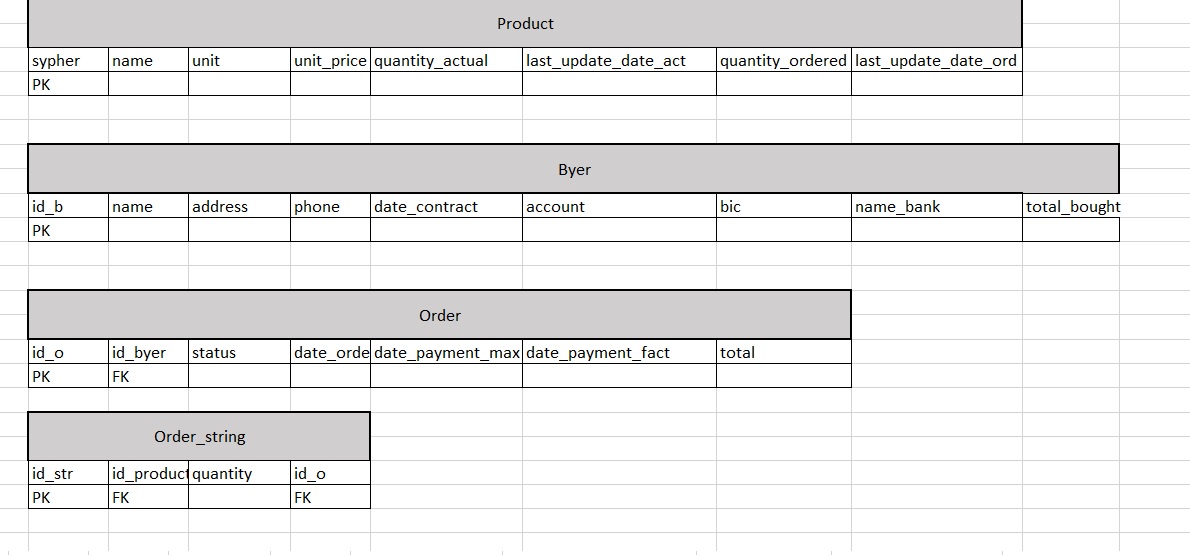
По истечении 3-х дней просроченные заказы аннулируются с проставлением в них соответствующего статуса.

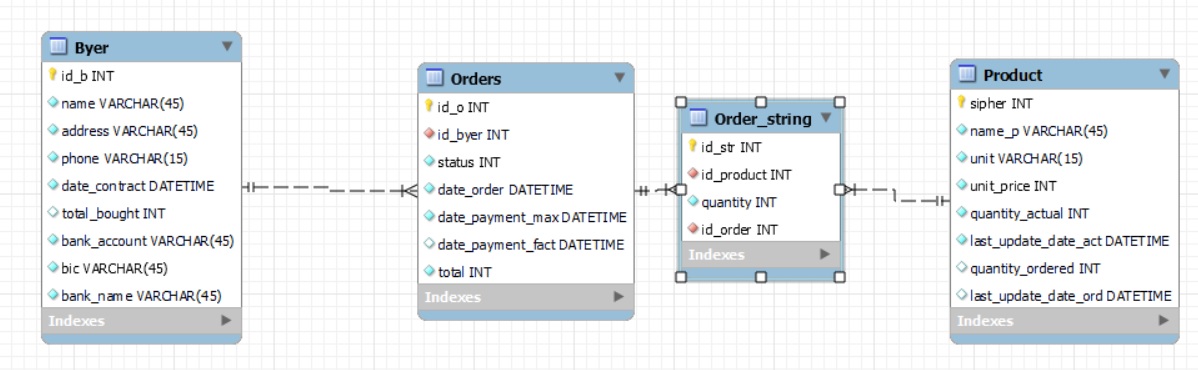
Для правильного учета фактически имеющегося на складе товара для *товаров* предусмотрены следующие атрибуты: количество фактически имеющегося на складе товара, дата его последнего обновления, количество заказанного, но не оплаченного товара, и дата его последнего обновления.

**Лабораторная работа №1**

**Реализация**

Инфологическая модель

Логическая модель



**Простые запросы**

1. Показать все сведения о покупателях, заключивших договора в марте 2020 года.

**SELECT \* FROM byer**

**WHERE YEAR(date\_contract)=2020 AND MONTH(date\_contract)=3;**

1. Показать все сведения о заказах, созданных за последние 60 дней.

**SELECT \* FROM orders**

**WHERE (DATEDIFF(CURRENT\_DATE(), date\_order))<60;**

1. Показать, сколько заказов из числа созданных за последние 5 дней, находятся в каждом из возможных статусов.

**SELECT state, count(id\_o) AS amount**

**FROM orders WHERE ((DATEDIFF(CURRENT\_DATE(), date\_order))<=5)**

**GROUP BY state;**

1. Показать всех покупателей, у которых номера телефонов имеют в префиксе 495.

**SELECT \* FROM byer WHERE phone LIKE "\_495%";**

1. Показать общую стоимость неоплаченных заказов для каждого покупателя.

**SELECT id\_byer, sum(total) FROM orders WHERE state = 1 GROUP BY id\_byer ORDER BY total;**

1. Показать максимальную стоимость оплаченного заказа для каждого покупателя в 2020 году.

**SELECT id\_byer, max(total) FROM orders WHERE state = 2 AND YEAR(date\_order) = 2022 GROUP BY id\_byer;**

**Лабораторная работа №2**

**Сложные запросы**

1. Выдать данные о покупках со склада в марте 2020 по форме: Шифр покупателя, название покупателя, название товара, общее количество купленного товара.

**SELECT id\_b, byer.name, product.name\_p, sum(quantity)**

**FROM byer JOIN orders ON orders.id\_byer=byer.id\_b**

**JOIN order\_string ON order\_string.id\_order=orders.id\_o**

**JOIN product ON product.sipher=order\_string.id\_product**

**WHERE MONTH(orders.date\_order) = 3 AND YEAR(orders.date\_order) = 2022 AND orders.state = 2**

**GROUP BY product.name\_p, byer.name;**

1. Покажите шифры и названия товаров, которые покупатель с именем ХХХ покупал в апреле 2020 года.

**SELECT DISTINCT product.sipher, product.name\_p FROM product**

**JOIN order\_string ON order\_string.id\_product=product.sipher**

**JOIN orders ON order\_string.id\_order=orders.id\_o**

**JOIN byer ON orders.id\_byer=byer.id\_b**

**WHERE MONTH(orders.date\_order) = 3 AND YEAR(orders.date\_order) = 2022**

**AND byer.name="Maxim";**

1. Покажите все сведения о самом дорогом товаре на складе.

**SELECT \* FROM product where unit\_price = (select max(unit\_price) from product);**

1. Показать все сведения о покупателях, заключивших долгосрочные договора, но ни разу не сделавших заказ (с помощью левостороннего соединения).

**SELECT byer.\***

**FROM byer LEFT JOIN orders ON orders.id\_byer=byer.id\_b**

**where id\_o is null;**

1. Показать все сведения о покупателях, ни разу не сделавших заказ на покупку товаров в марте 2020 года.

**SELECT byer.\* FROM byer**

**LEFT JOIN (select \* from orders where year(date\_order)=2022 and month(date\_order)=3) as ss**

**ON ss.id\_byer=byer.id\_b**

**where id\_o is null;**

1. Показать все сведения о покупателе, сделавшем наибольшее количество покупок в 2020 году (с использованием view).

**CREATE VIEW bestbyer as**

**SELECT id\_b, name, address, phone, date\_contract, total\_bought, bank\_account, bic, bank\_name, count(id\_o) AS "Count\_of\_orders" FROM byer**

**JOIN orders ON id\_byer = id\_b**

**WHERE year(date\_order) = 2022 group by id\_b;**

**select \* from bestbyer where Count\_of\_orders = (select max(Count\_of\_orders) from bestbyer);**

**Лабораторная работа №3**

**Триггеры и процедуры**

Разработать триггер, который будет срабатывать при изменении статуса заказа. В триггере реализовать увеличение суммы ранее купленных заказов для соответствующего покупателя, если заказ поменял статус с 1 на 2.

**CREATE TRIGGER upd\_state AFTER UPDATE ON Orders FOR EACH ROW**

**BEGIN**

**DECLARE ss INT;**

**DECLARE olds INT;**

**IF new.state = 2 and old.state=1 THEN**

**SELECT total\_bought FROM byer**

**JOIN orders ON byer.id\_b=NEW.id\_byer AND id\_o=OLD.id\_o INTO olds;**

**UPDATE byer**

**SET total\_bought = olds+NEW.total**

**WHERE id\_b=NEW.id\_byer;**

**END IF;**

**END**

Разработать процедуру, которая будет принимать на вход дату. В процедуре реализовать аннулирование всех заказов, для которых истек срок оплаты.

**CREATE PROCEDURE date\_check(dat DATE)**

**BEGIN**

**UPDATE orders**

**SET state = 3**

**WHERE DATE(date\_payment\_max) < DATE(dat) AND state = 1;**

**end**