Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

Кафедра безпеки інформаційних систем і технологій

Лабораторна робота №3

*з навчальної дисципліни*

«Захист інформації в інформаційно-комунікаційних системах»

Виконав:

Студент групи КБ-31 Кравченко Є. М.

Перевірив:

Професор

Єсін В. І.

Харків – 2020 р.

**Тема:** «Проектирование базы данных на основе принципов нормализации»

**Цель работы:** ознакомление с методами проектирования реляционных баз данных на основе принципов нормализации (восходящее проектирование) в среде СУБД Access.

***1. Постановка задачи***

Спроектировать БД для организации работы предприятия, занимающегося продажей компьютерной техники, используя *восходящий* подход (*процесс нормализации* схем отношений).

Известны:

1. Реализуемые компанией товары, их характеристики, описание и цена.
2. Принцип оформления заказов клиентами (как оформляются заказы).
3. Модель предприятия, представленная в виде двух отношений реляционной модели данных (рис. 1).

**Товары**

**Заказы**

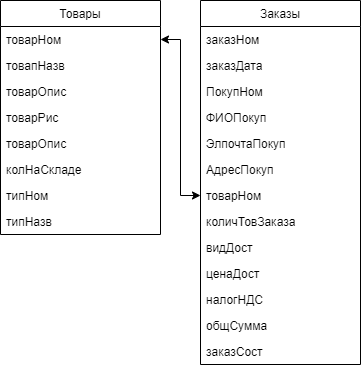
*Рис. 1. Начальная схема отношений*

Процесс нормализации завершить рассмотрением 3НФ.

***2. Нормализация БД***

*1-ый этап: 1 НФ → 2 НФ*

Из схемы организации работы фирмы на рис. 1 определим схему отношений:



*Рис. 2 – Начальная схема отношений со связями*

В атрибутах *«типНом»* и *«типНазв»* содержится информация о типах товаров, например, название типа − Процессоры (типНом = 1), к которому относятся конкретные товары − Intel 386, Intel 486, AMD, и т.д. Атрибут *«колНаСкладе»* содержит общее количество товаров с данным номером на складе.

В одном заказе может заказываться несколько товаров с разными номерами. Атрибут *«количТовЗаказа»* содержит количество товаров данного типа, заказанных в данном заказе.

В атрибутах *«ПокупНом», «ФИОПокуп», «ЭлпочтаПокуп», «адресПокуп»* содержится информация о покупателе − номер, ФИО, Электронная почта и адрес.

В атрибутах *«видДост», «ценаДост»* хранится информация о доставке − вид и цена доставки. Атрибут *«заказСост»* содержит состояние заказа, который может быть иметь значение: «Сделка совершена» , «Сделка отменена» (признак сделки: Kо=0,1)

Общая сумма заказа вычисляется по формуле:

*общСумма =* Kо*×(ТоварЦена\*количТов + налогНДС + ценаДост).*

На 1-м шаге проектирования форма 1НФ приводится к виду 2НФ. Для этого проверяем следующее определение:

*Отношение находится во второй нормальной форме (2НФ), если оно находится в первой нормальной форме и каждый его атрибут, не входящий в состав первичного ключа, характеризуется полной функциональной зависимостью от этого первичного ключа.*

Рассмотрим отношение *Заказы*, первичный ключ которого состоит из следующих атрибутов: *заказНом, заказДата, ПокупНом, ФИОПокуп, ЭлпочтаПокуп, адресПокуп, товарНом, количТовЗаказа, видДост, ценаДост, налогНДС, общСумма, заказСост*.

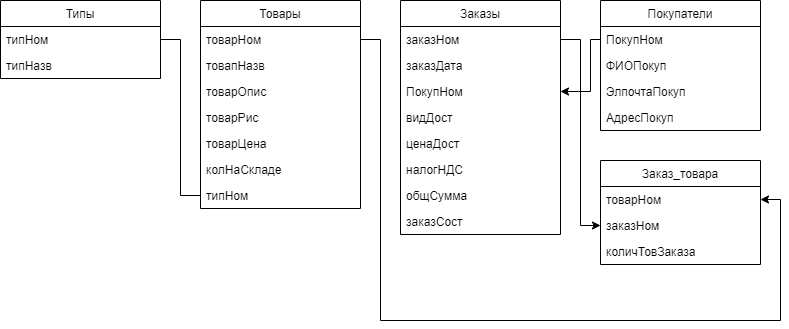
В качестве потенциальных ключей могут быть следующие наборы атрибутов:

*заказНом, ПокупНом, товарНом.*

В отношении *Заказы*, кроме зависимости от первичного ключа, существуют следующие функциональные зависимости от части первичного ключа:

*общСумма.*

откуда следует, что отношение *Заказы* не находится во второй нормальной форме (существуют неполные (частичные) функциональные зависимости). Поэтому отношение *Заказы* необходимо привести ко второй нормальной форме. Для чего произведем декомпозицию отношения *Заказы* на следующие отношения: Покупатели, Заказы, Заказ\_товара:



*Рис. 3 – Схема отношений Покупатели, Заказы, Заказ\_товара*

Схема процесса преобразования отношений из 1НФ во 2НФ приведена на рис. 4.

**1НФ**

**2НФ**

**Товары**

**Товары**

**Заказы**

**Заказы**

**Покупатели**

**Заказ\_товара**

*Рис. 4 – Схема процесса преобразования отношений из 1НФ во 2НФ*

*2-ой этап: 2 НФ → 3 НФ*

На 2-м шаге проектирования форма 2НФ приводится к виду 3НФ. Для этого проверяется следующее определение:

*Отношение находится в третьей нормальной форме (3НФ), если оно находится в первой и во второй нормальных формах и не имеет атрибутов, не входящих в состав первичного ключа, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.*

Рассмотрим функциональные зависимости (ФЗ) отношени *Товары.* Атрибут *типНазв* напрямую не зависит от первичного ключа *товарНом*, а непосредственно зависит от его составного атрибута *типНом*, следовательно, здесь имеет место транзитивная ФЗ *типНом ­– типНазв* Другими словами, название типа товара на самом деле является характеристикой не товра, а типа товара, к которому он относится. Следовательно, отношение *Товары* не находится в третьей нормальной форме, и его необходимо к ней привести.

Для этого проведем декомпозицию отношения *Товары* :



*Рис. 5 – Схема отношений Типы, Товары*

При этом отношение *Товары* – основная таблица, а отношение *Типы* – подчиненная таблица.

Таблицы *Типы, Товары, Заказы, Покупатели* уже находятся в 3НФ, так как не имеют транзитивных зависимостей.

Схема процесса преобразования отношений из 2НФ в 3НФ приведена на рис. 6.

**Заказ\_товара**

**2НФ**

**3НФ**

**Товары**

**Товары**

**Заказы**

**Заказы**

**Покупатели**

**Заказ\_товара**

**Типы**

**Покупатели**

*Рис. 3 – Схема процесса преобразования отношений из 2НФ в 3НФ*

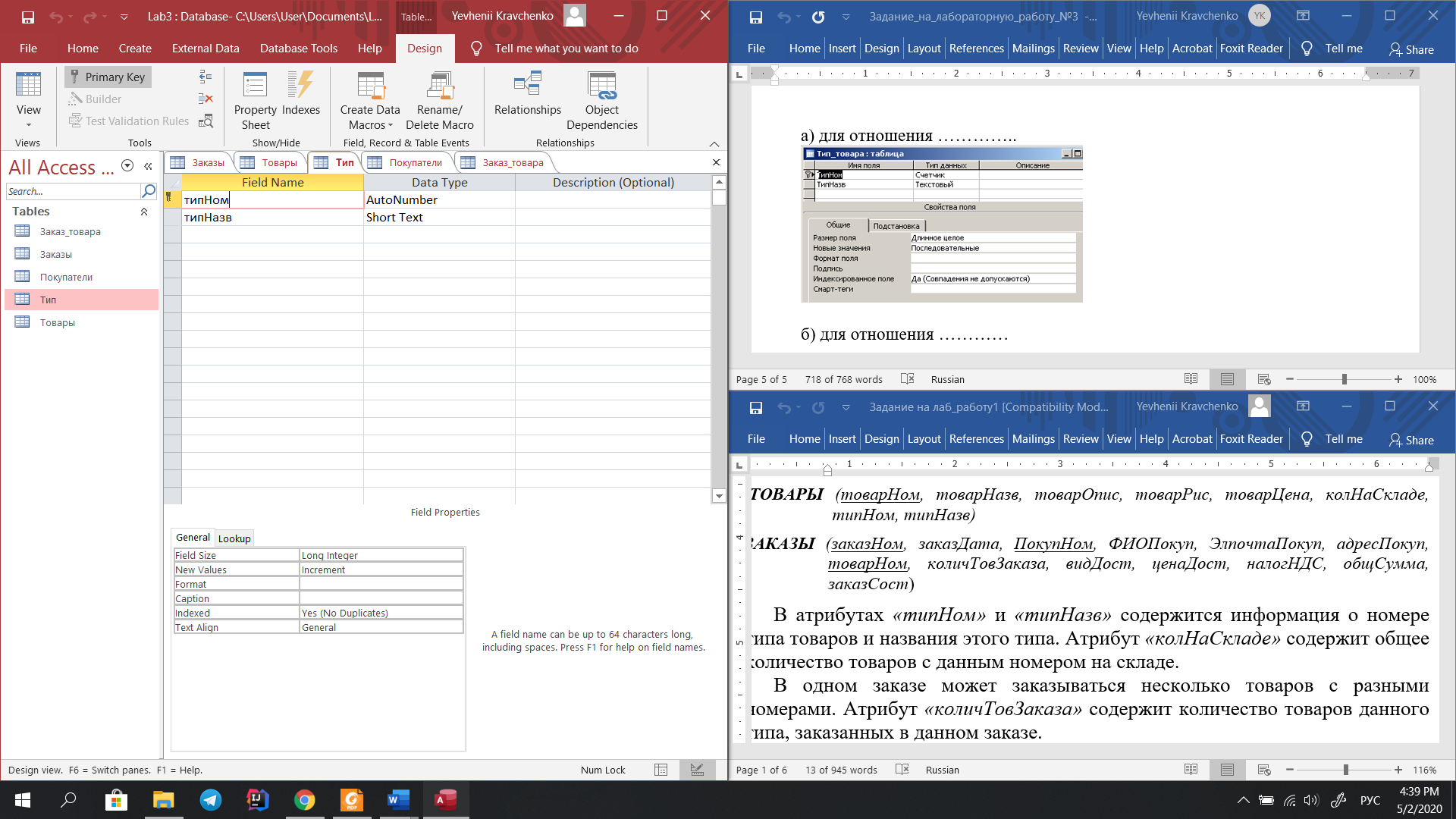
На практике достаточно часто процесс проектирования реляционных баз данных заканчивается на третьей нормальной форме. По условию задачи, мы тоже ограничимся третьей нормальной формой.

***3. Проектирование БД в среде Microsoft Access***

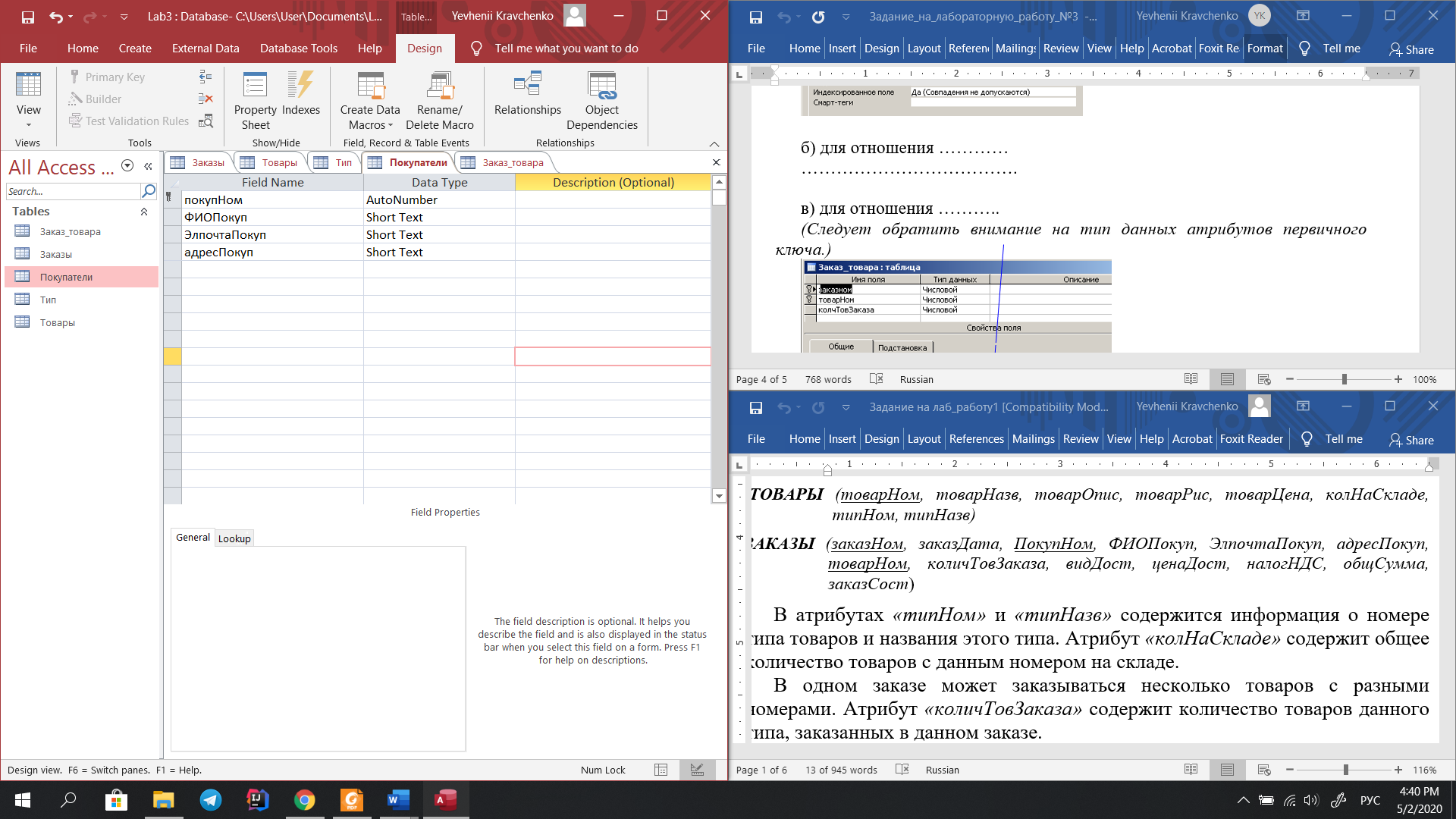
Теперь занесем эти нормализованные отношения в среду базы данных Access, определив при этом типы данных атрибутов этих отношений.

Типы данных для каждого отношения приведены в следующих таблицах:

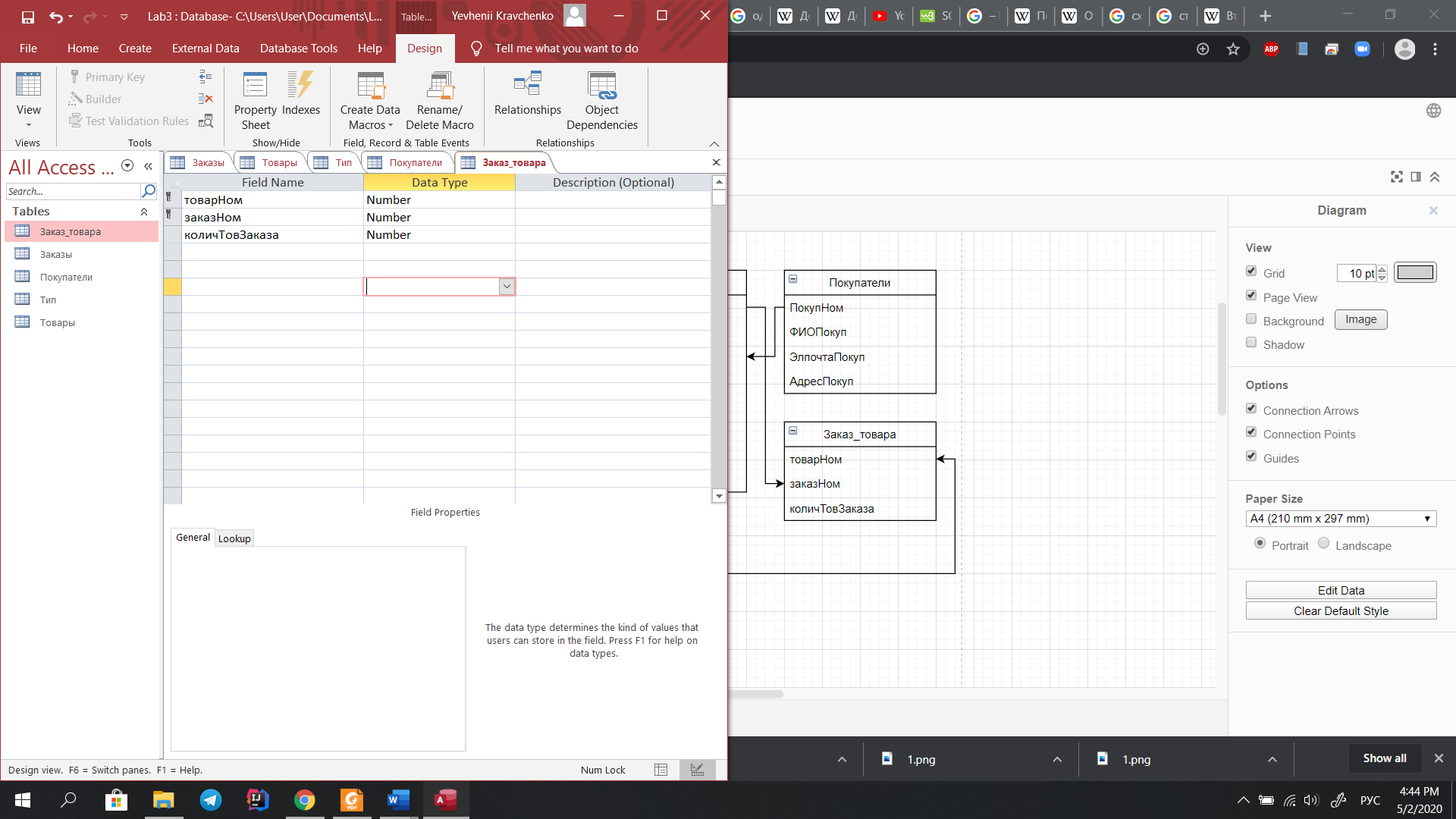
а) для отношения Типы



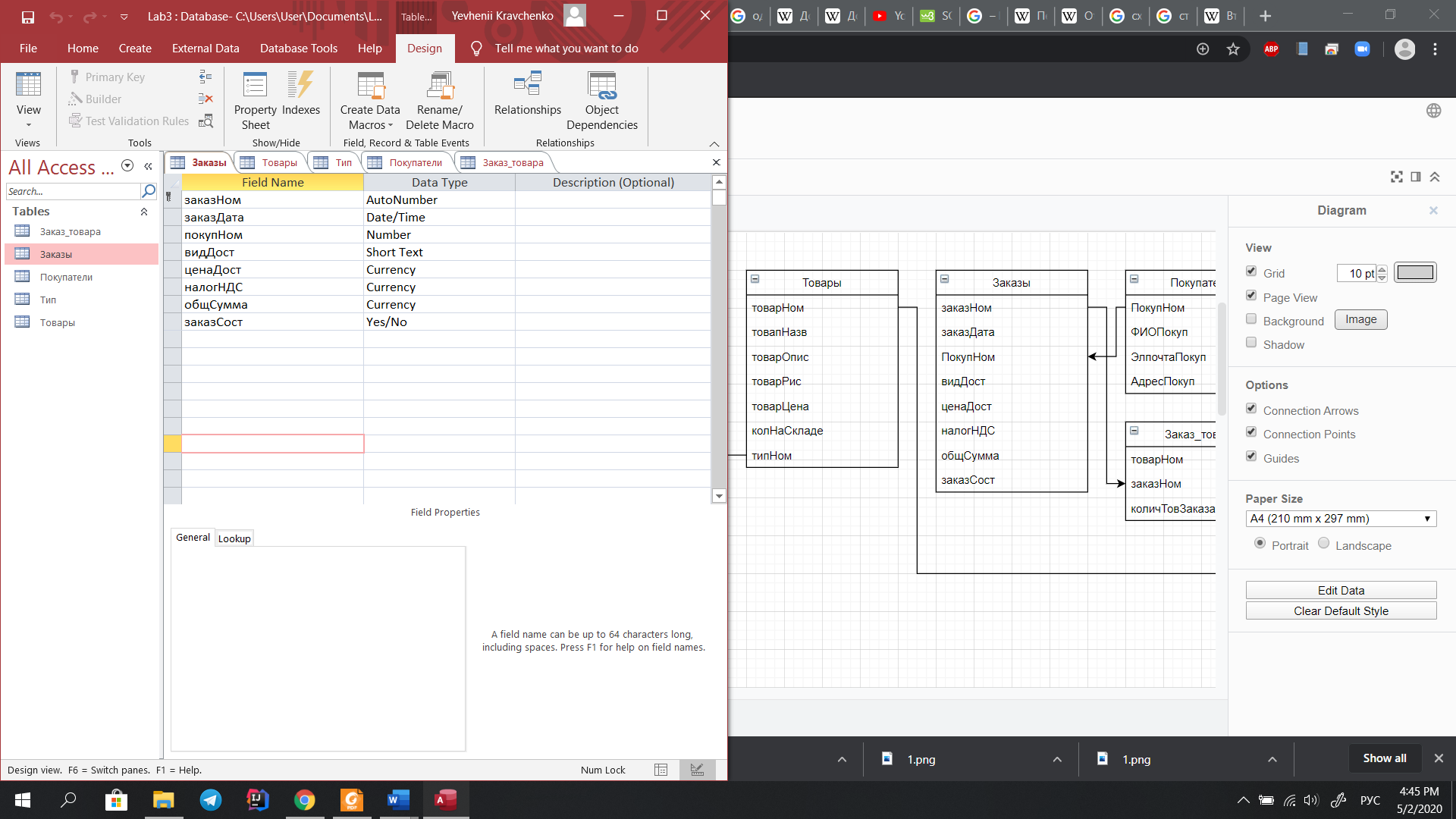
б) для отношения Покупатели



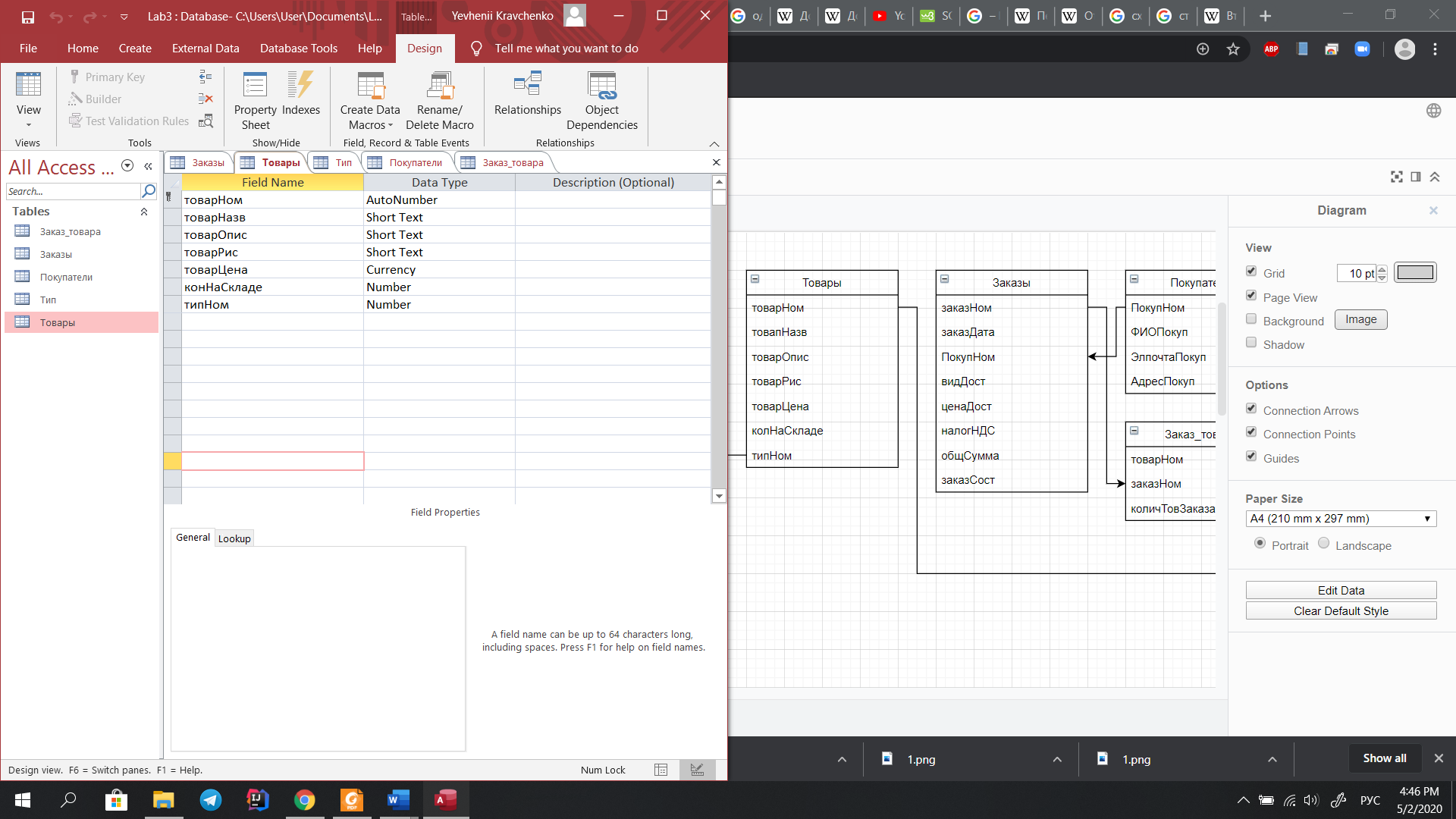
в) для отношения Заказ\_товара



г) для отношения Заказы



д) для отношения Товары



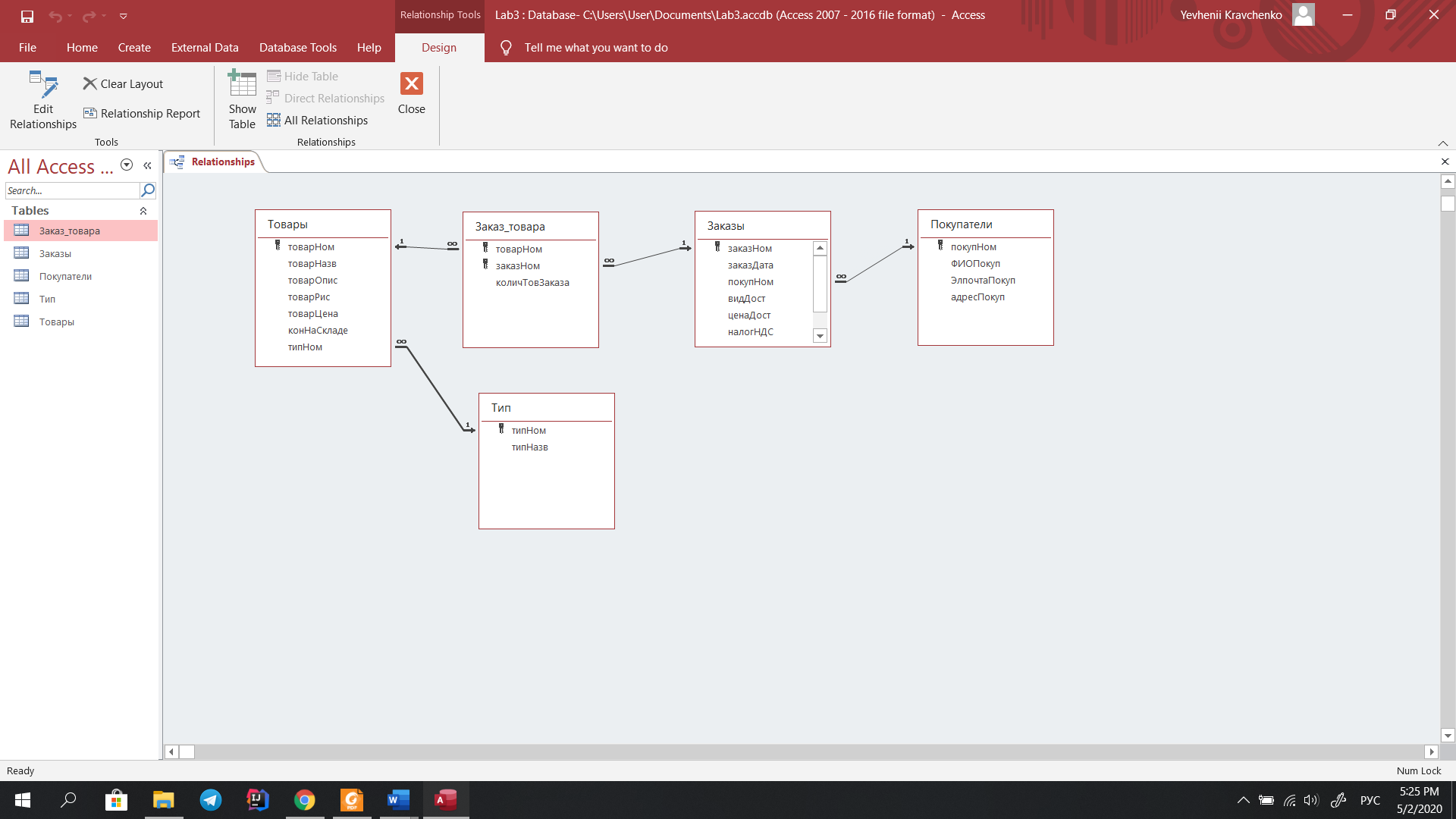
После чего необходимо определить связи между этими отношениями с учетом ограничений предметной области.

В результате анализа, приходим к тому, что между отношениями имеют место следующие связи:

* «Заказы» и «Заказ\_товар» (связь 1:\*);
* «Заказы» и «Заказ\_товар» (связь 1:\*);
* «Типы» и «Товары» (связь 1:\*);
* «Покупатели» и «Заказы» (связь 1:\*).

Для того чтобы выполнить требования целостности данных и требования предметной области (бизнес-правила), ″пропишем″ эти связи в базу данных.

Искомая схема данных имеет следующий вид:



Теперь можно приступать к занесению информации в спроектированные таблицы.

**Выводы**

В этой лабораторной работе я провел ознакомление с методами проектирования реляционных баз данных на основе принципов нормализации (восходящее проектирование) в среде СУБД Access.