Лабораторная работа 2.

Тема: «Создание таблиц в MS ACCESS средствами языка SQL»

Цель работы:

* получить навыки создания таблиц средствами языка SQL;
  1. создание полей с необходимыми типами данных;
  2. задание необходимых ограничений;
* получить навыки создания первичных и внешних ключей.
* Формирование схемы базы данных

Описания структуры таблиц:

Таблица 1-  **НАЧИНКИ**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_начинки | Счетчик |  | Ключ |
| Сорт\_начинки | Текстовый | 20 | Обязательное, индексированное- совпадения не  допускаются |

Таблица 2- **КОНФЕТЫ**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_конфеты | Текстовый | 5 | Ключ |
| Название | Текстовый | 20 | Обязательное |
| Сорт шоколада | Текстовый | 20 | Значение по умолчанию: «Горький» |
| Сорт ореха | Текстовый | 20 | Значение по умолчанию: «Нет» |
| Начинка | Числовой | Длинное  целое | Значение по умолчанию: 0 |
| Описание\_конфеты | Поле MEMO |  | Необязательное |
| Цена | Денежный |  | Число десятичных знаков: 2 |
| Картинка | объект OLE |  | Необязательное |

Таблица 3- **ЗАКАЗЧИКИ**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_заказчика | Счетчик |  | Ключ |
| Фамилия | Текстовый | 20 | Обязательное |
| Имя | Текстовый | 12 | Обязательное |
| Отчество | Текстовый | 12 | Необязательное |
| Адрес | Текстовый | 50 | Обязательное |
| Город | Текстовый | 20 | Значение по умолчанию: «Санкт-Петербург» |
| Район | Текстовый | 20 | Обязательный |
| Страна | Текстовый | 20 | Значение по умолчанию: «Россия» |
| Почтовый индекс | Текстовый | 6 |  |
| Телефон | Текстовый | 15 |  |
| Кредитная\_карта | Числовой | Длинное целое | Значение по умолчанию: 0 |
| Номер кредитной карты | Текстовый | 14 | Индексированное: Совпадения не допускаются |

Таблица 4 - **НАБОРЫ**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_набора | Текстовый | 8 | Ключ |
| Название набора | Текстовый | 20 | Обязательное |
| Вес набора | Числовой |  | Одинарное с плавающей точкой |
| Описание\_набора | Поле MEMO |  | Необязательное |
| Цена\_набора | Денежный |  |  |
| На складе | Числовой | Целое | Необязательное |

Таблица 5 - **СОТРУДНИКИ**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_сотрудника | Счетчик | Длинное целое | Ключ |
| Фамилия | Текстовый | 20 | Обязательное |
| Имя | Текстовый | 20 | Обязательное |
| Отчество | Текстовый | 20 | Необязательное |
| Должность | Текстовый | 40 | Обязательное |
| Телефон | Текстовый | 13 | Необязательное |
| Отдел | Текстовый | 20 | Обязательное |
| Дата\_рождения | Дата/время |  |  |
| Дата\_приема | Дата/время |  |  |

Таблица 6 - ТИПЫ КРЕДИТНЫХ КАРТ

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_кредитной карты | Счетчик |  | Ключ |
| Название карты | Текстовый | 20 | Обязательное |

Таблица 7 - ТРАНСПОРТ

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_компании | Счетчик |  | Ключ |
| Транспортная\_компания | Текстовый | 25 | Обязательное, Совпадения не допускаются |
| Представитель | Текстовый | 20 |  |
| Контактный \_телефон | Текстовый | 13 | Необязательное |
| Тариф | Денежный |  | Обязательное, число десятичных знаков 2 |

Таблица 8 - ПОДРОБНОСТИ НАБОРА

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_конфеты | Текстовый | 5 | Составной Ключ, Индексированное — Да(Допускаются совпадения) |
| Код\_набора | Текстовый | 8 |
| Количество | Числовой | Целое | Значение по умолчанию: 0 |

Таблица 9 - **ЗАКАЗЫ**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код\_заказа | Числовой | Длинное целое | Ключ |
| Код\_заказчика | Числовой | Длинное целое | Индексированное — Да(Допускаются совпадения) |
| Код\_трансп\_компании | Числовой | Длинное целое | Индексированное – Да(Допускаются совпадения) |
| Доставка | Логический |  | Да/Нет |
| Дата\_заказа | Дата/время |  |  |
| Дата\_оплаты | Дата/время |  |  |
| Номер счета | Числовой | Целое | Индексированное-Да(Совпадения не допускаются) |

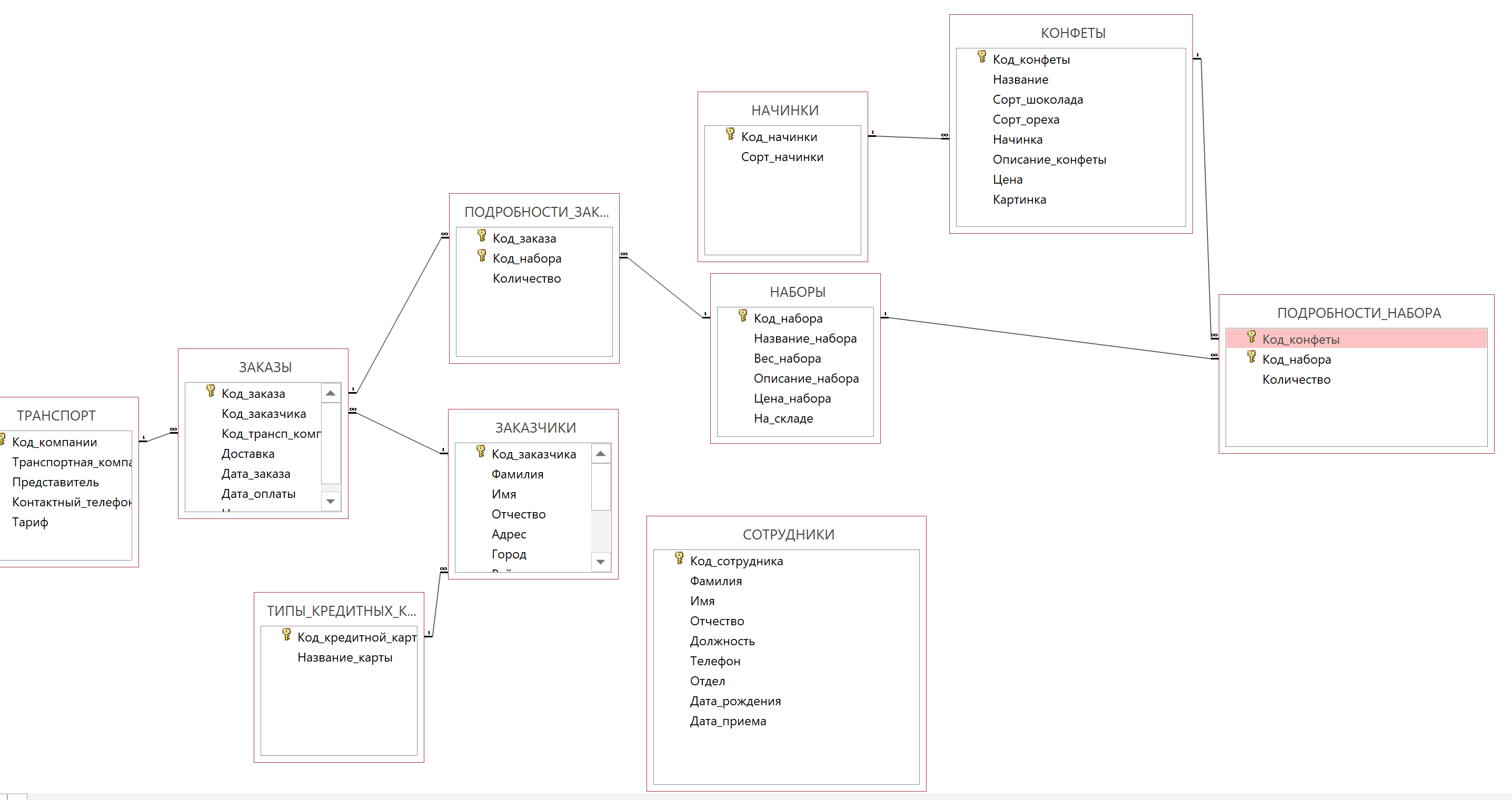
Таблица 10 - ПОДРОБНОСТИ ЗАКАЗА

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип | Размер | Другие свойства |
| Код заказа | Числовой | Длинное целое | Составной ключ, Индексированное -Да(Допускаются совпадения) |
| Код\_набора | Текстовый | 8 |
| Количество | Числовой | Целое | Значение по умолчанию: 0 |

Запросы:

|  |  |
| --- | --- |
| Название запроса | Листинг запроса на создание |
| Создание Начинки | CREATE TABLE НАЧИНКИ  (  Код\_начинки COUNTER(1,1) PRIMARY KEY,  Сорт\_начинки VARCHAR(20) NOT NULL  ); |
| Создание Конфеты | CREATE TABLE КОНФЕТЫ  (  Код\_конфеты VARCHAR(5) PRIMARY KEY,  Название VARCHAR(20) NOT NULL,  Сорт\_шоколада VARCHAR(20) DEFAULT горький,  Сорт\_ореха VARCHAR(20) DEFAULT нет,  Начинка INTEGER DEFAULT 0,  Описание\_конфеты VARCHAR(255),  Цена DECIMAL(18, 2),  Картинка IMAGE,  FOREIGN KEY(Начинка)  REFERENCES НАЧИНКИ(Код\_начинки)  ); |
| Создание Заказчики | CREATE TABLE ЗАКАЗЧИКИ  (  Код\_заказчика COUNTER(1, 1) PRIMARY KEY,  Фамилия VARCHAR(20) NOT NULL,  Имя VARCHAR(12) NOT NULL,  Отчество VARCHAR(12),  Адрес VARCHAR(50) NOT NULL,  Город VARCHAR(20) DEFAULT "Санкт-Петербург",  Район VARCHAR(20),  Страна VARCHAR(20) DEFAULT "Россия",  Почтовый\_индекс VARCHAR(6),  Телефон VARCHAR(15),  Кредитная\_карта Integer DEFAULT 0,  Номер\_кредитной\_карты VARCHAR(14) UNIQUE  ); |
| Создание Наборы | CREATE TABLE Наборы  (  Код\_набора VARCHAR(8) PRIMARY KEY,  Название\_набора VARCHAR(20) NOT NULL,  Вес\_набора REAL,  Описание\_набора MEMO,  Цена\_набора DECIMAL(18, 2),  На\_складе INTEGER  ); |
| Создание Сотрудники | CREATE TABLE СОТРУДНИКИ  (  Код\_сотрудника COUNTER(1, 1) PRIMARY KEY,  Фамилия VARCHAR(20) NOT NULL,  Имя VARCHAR(20) NOT NULL,  Отчество VARCHAR(20),  Должность VARCHAR(40) NOT NULL,  Телефон VARCHAR(20),  Отдел VARCHAR(20) NOT NULL,  Дата\_рождения DATETIME,  Дата\_приема DATETIME  ); |
| Создание Типы кредитных карт | CREATE TABLE ТИПЫ\_КРЕДИТНЫХ\_КАРТ(  Код\_кредитной\_карты COUNTER(1, 1) PRIMARY KEY,  Название\_карты VARCHAR(20)  ); |
| Создание Транспорт | CREATE TABLE ТРАНСПОРТ  (  Код\_компании COUNTER(1,1) PRIMARY KEY,  Транспортная\_компания VARCHAR(25) NOT NULL UNIQUE,  Представитель VARCHAR(20),  Контактный\_телефон VARCHAR(13),  Тариф DECIMAL(18, 2)  ); |
| Создание Подробности набора | CREATE TABLE ПОДРОБНОСТИ\_НАБОРА  (  Код\_конфеты VARCHAR(5),  Код\_набора VARCHAR(8),  Количество INTEGER DEFAULT 0,  PRIMARY KEY (Код\_конфеты, Код\_набора),  FOREIGN KEY (Код\_конфеты) REFERENCES КОНФЕТЫ (Код\_конфеты),  FOREIGN KEY (Код\_набора) REFERENCES НАБОРЫ (Код\_набора)  ); |
| Создание Заказы | CREATE TABLE ЗАКАЗЫ  (  Код\_заказа INTEGER PRIMARY KEY,  Код\_заказчика INTEGER,  Код\_трансп\_компании INTEGER,  Доставка BIT,  Дата\_заказа DATETIME,  Дата\_оплаты DATETIME,  Номер\_счета INTEGER UNIQUE,  FOREIGN KEY (Код\_заказчика) REFERENCES ЗАКАЗЧИКИ (Код\_заказчика),  FOREIGN KEY (Код\_трансп\_компании) REFERENCES ТРАНСПОРТ (Код\_компании)  ); |
| Создание Подробности заказа | CREATE TABLE ПОДРОБНОСТИ\_ЗАКАЗА  (  Код\_заказа INTEGER,  Код\_набора VARCHAR(8),  Количество INTEGER DEFAULT 0,  PRIMARY KEY(Код\_заказа, Код\_набора),  FOREIGN KEY (Код\_заказа)  REFERENCES ЗАКАЗЫ (Код\_заказа),  FOREIGN KEY (Код\_набора)  REFERENCES НАБОРЫ (Код\_набора)  ); |
| Добавить связь тип данных | ALTER TABLE ЗАКАЗЧИКИ ADD  CONSTRAINT FKKRED FOREIGN KEY (Кредитная\_карта)  REFERENCES ТИПЫ\_КРЕДИТНЫХ\_КАРТ(Код\_кредитной\_карты); |

Схема базы данных:



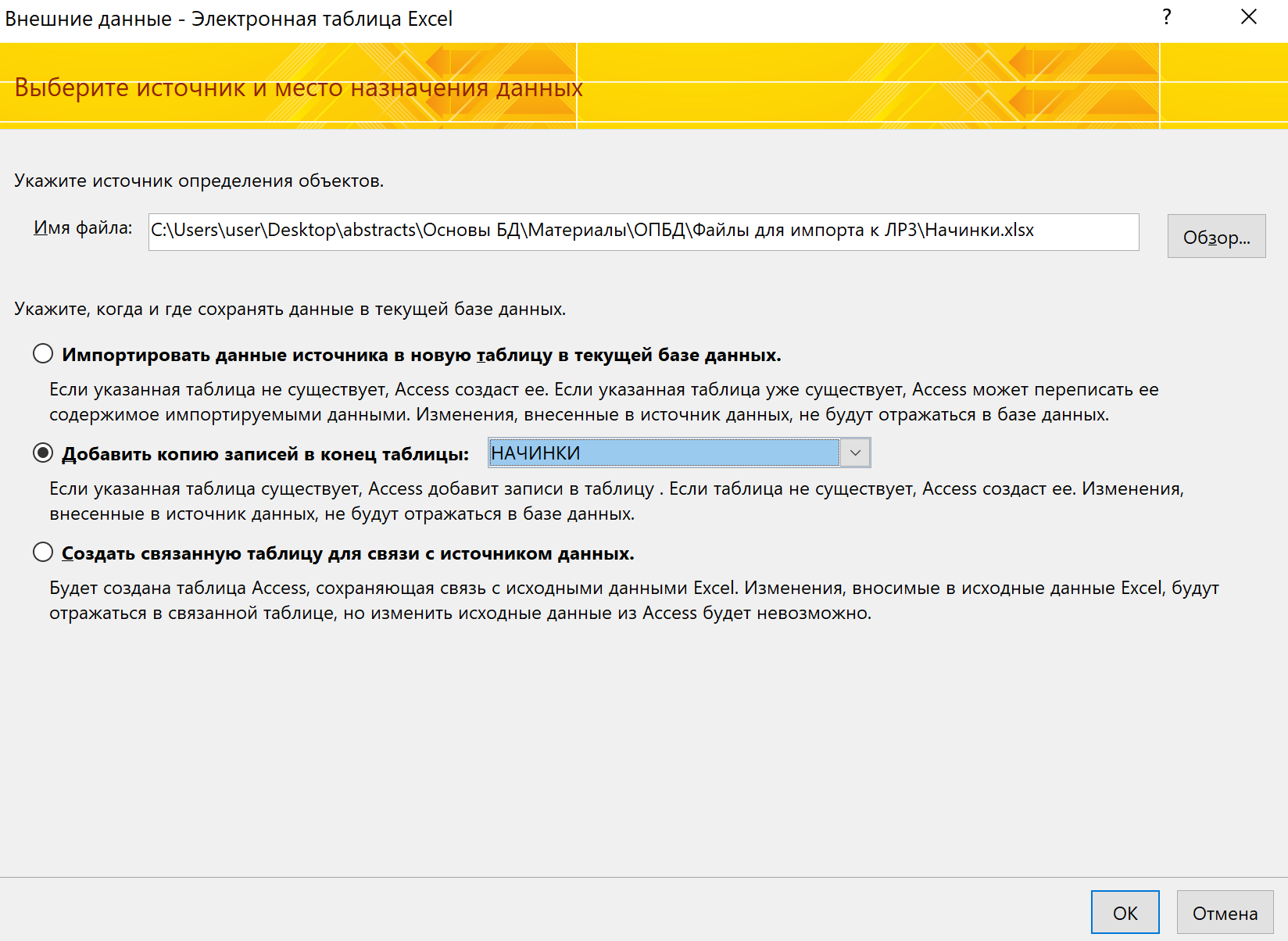
**Лабораторная работа 3**

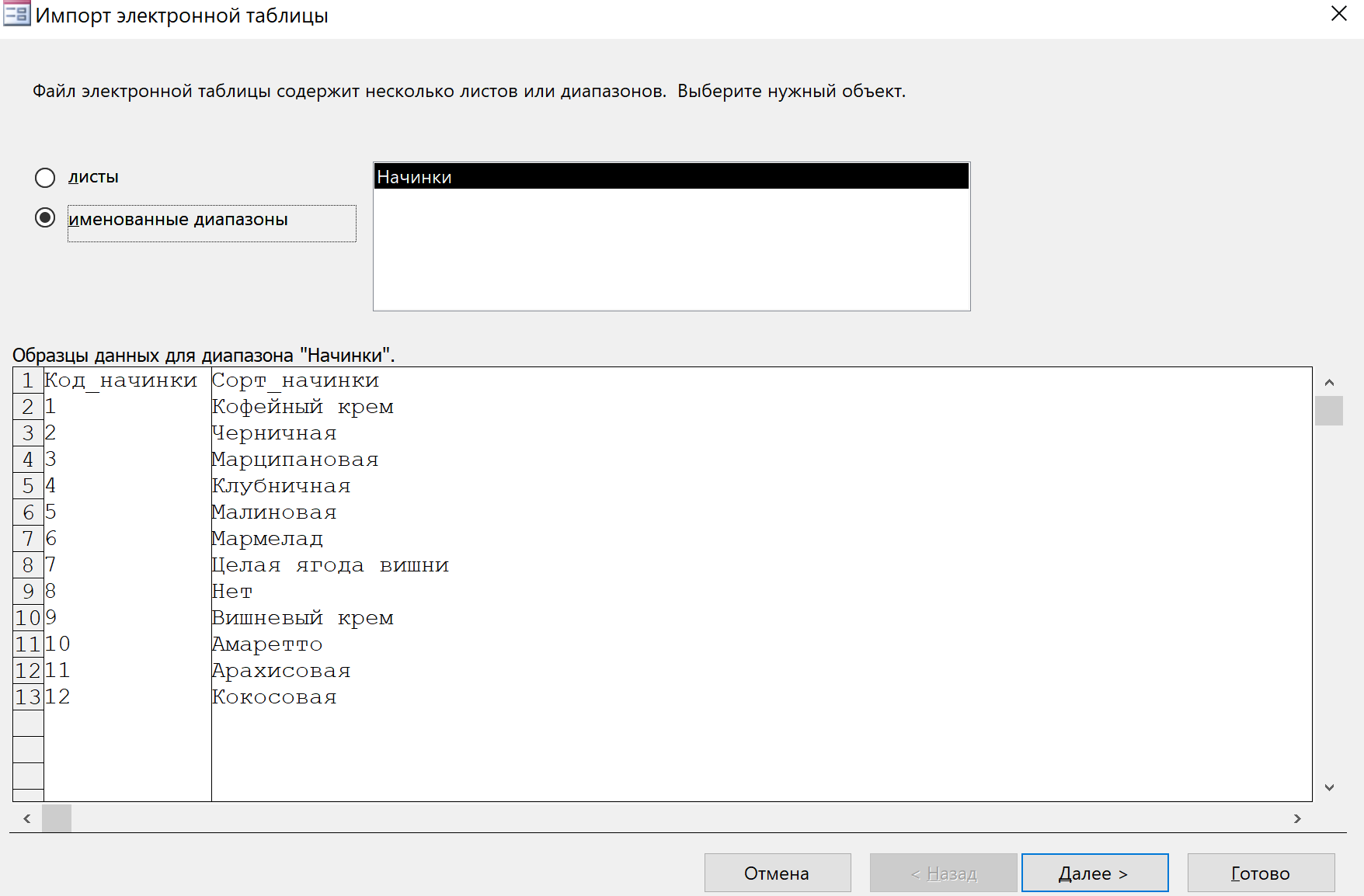
**Заполнение таблиц. Импорт данных в таблицы**

**Цель работы**: получить навыки заполнения таблиц

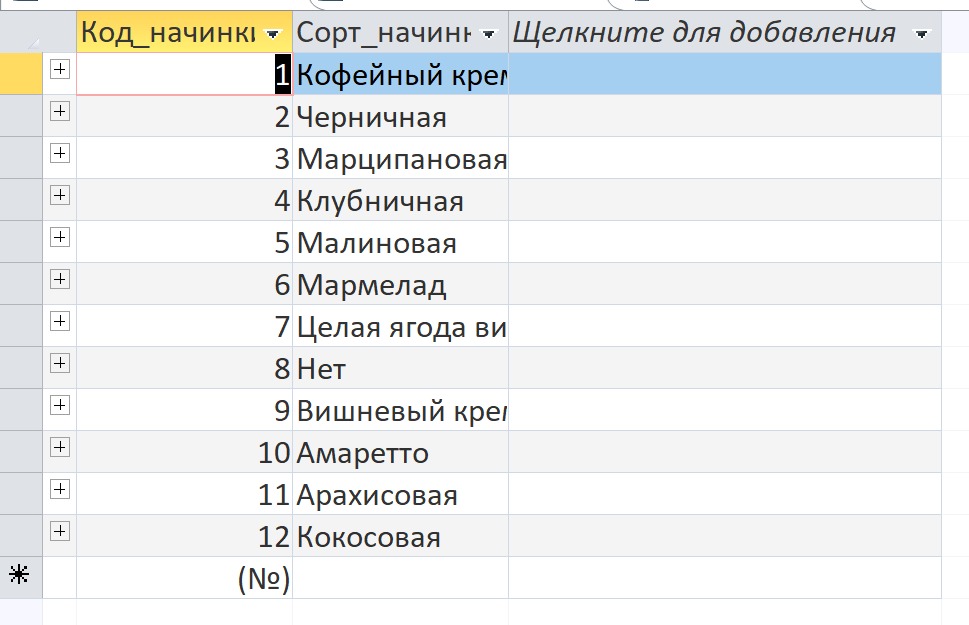
* путем непосредственного ввода данных в таблицы;
* импортом данных из внешних файлов;
* заполнения таблиц средствами языка SQL.

Импорт начинок:

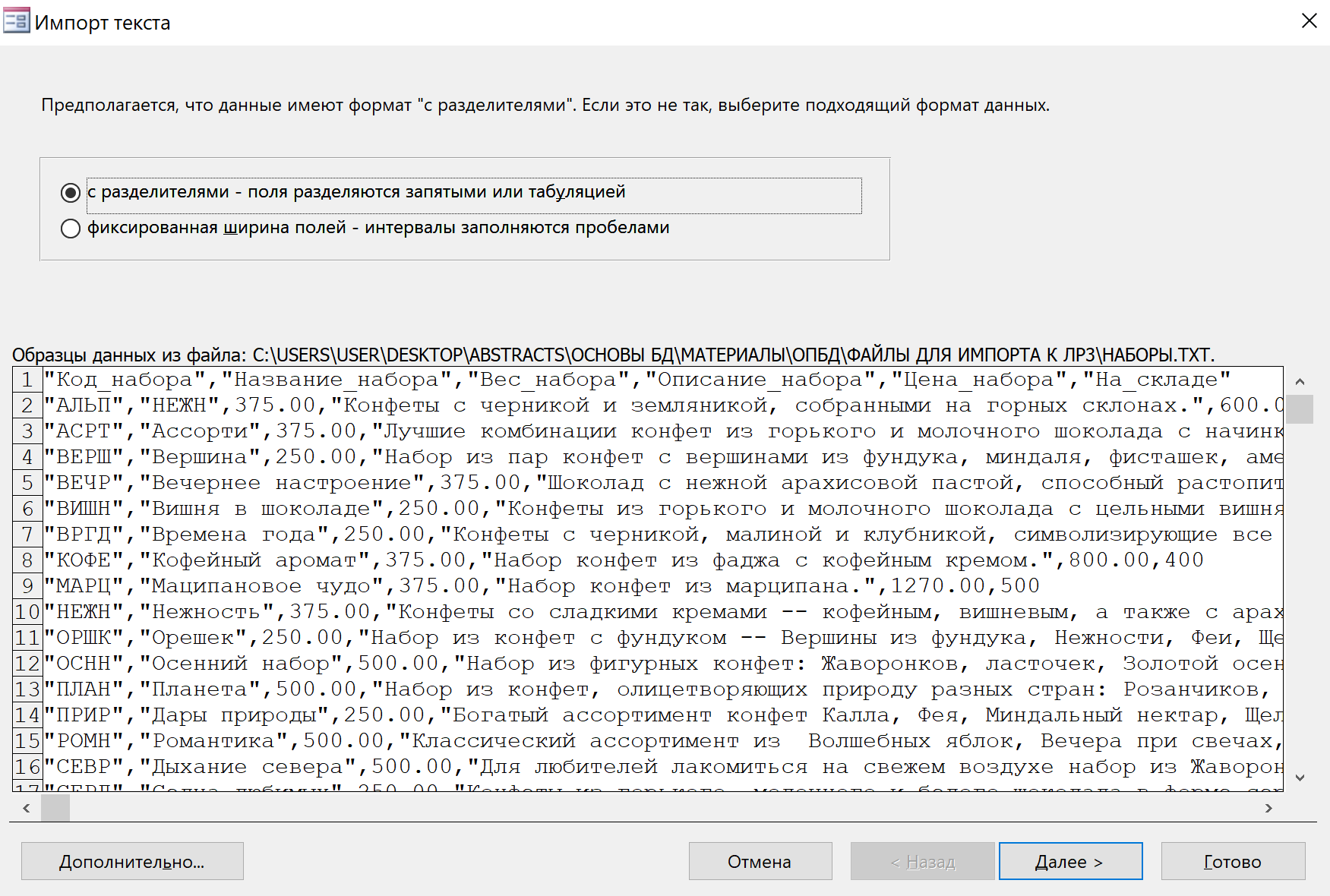


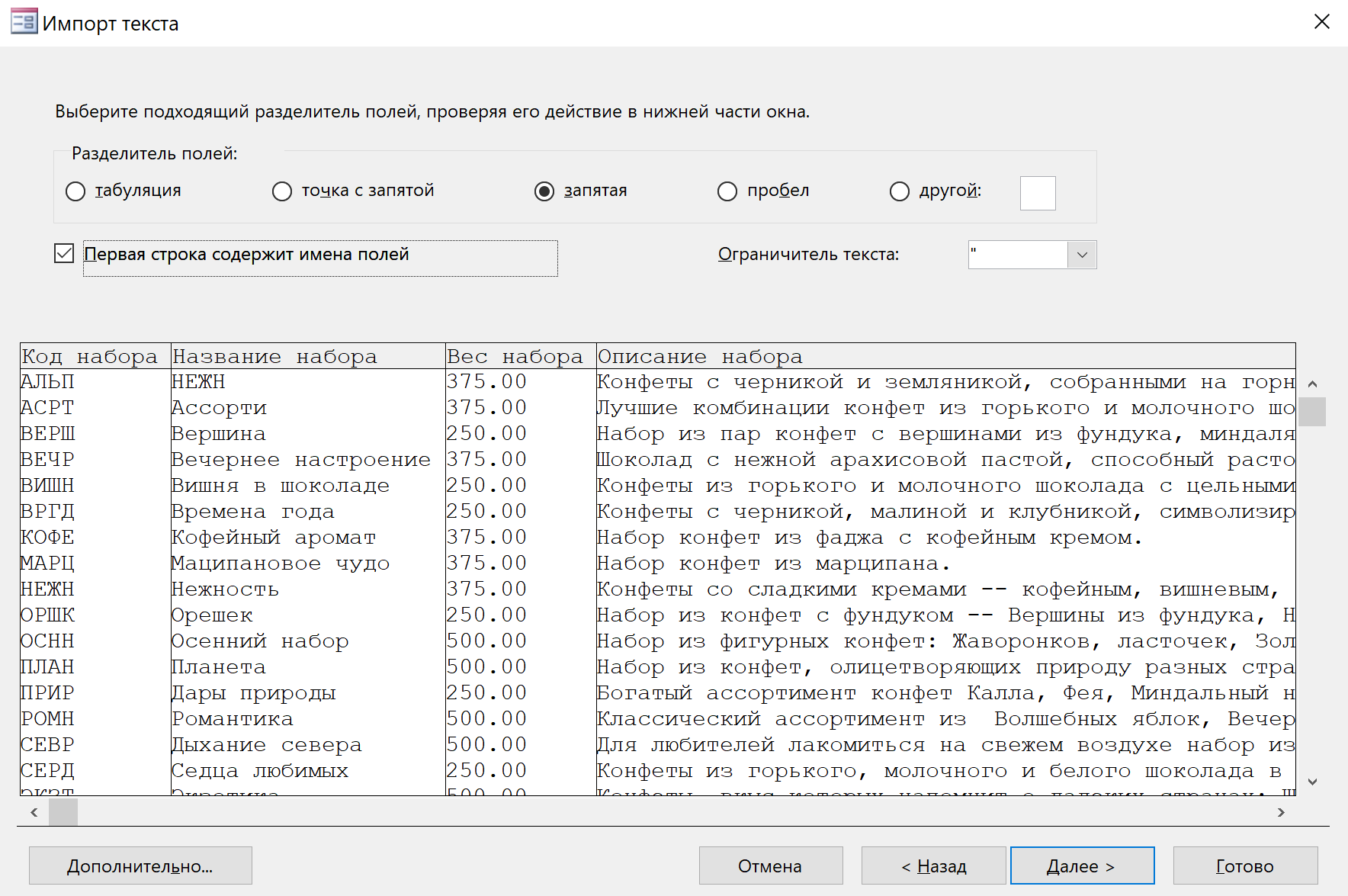


Итог импорта:

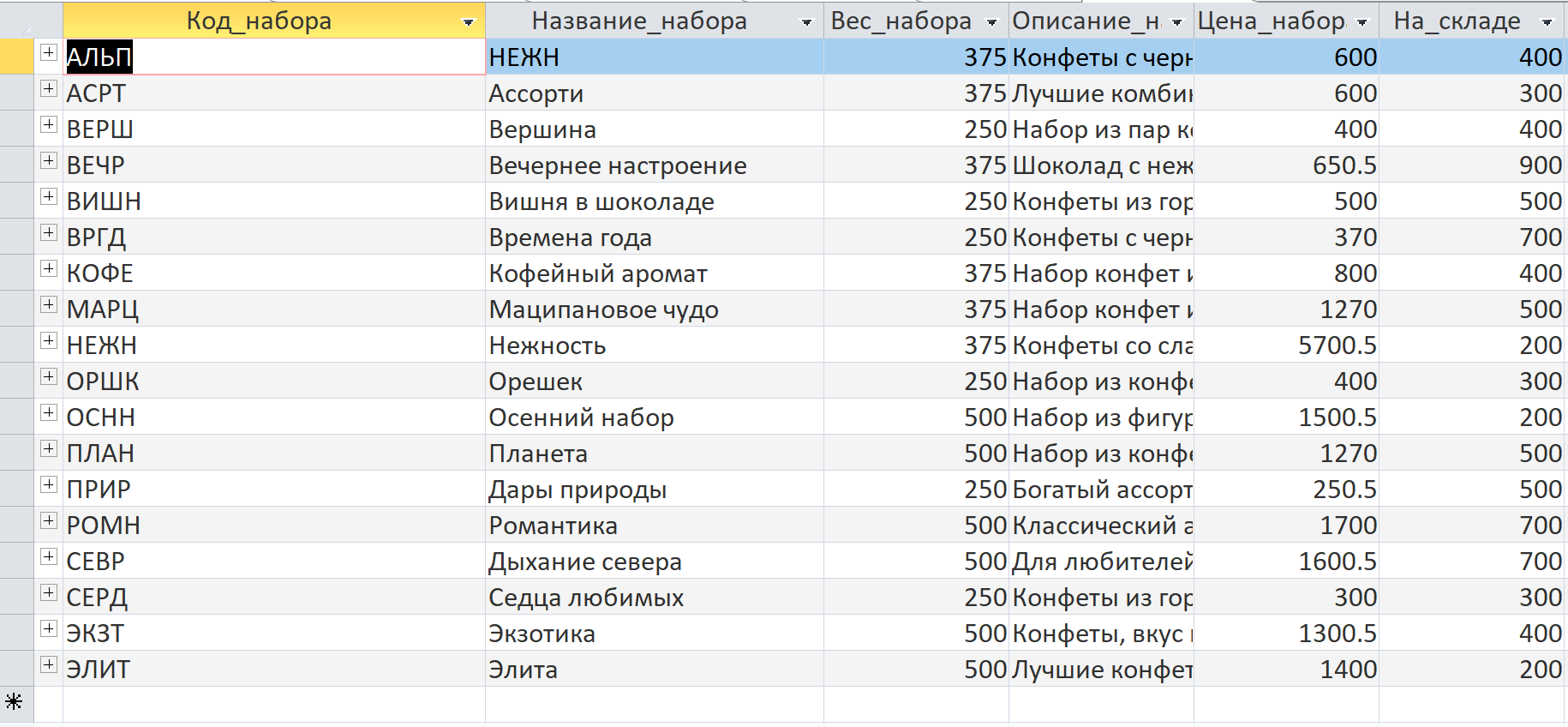


Импорт наборы:

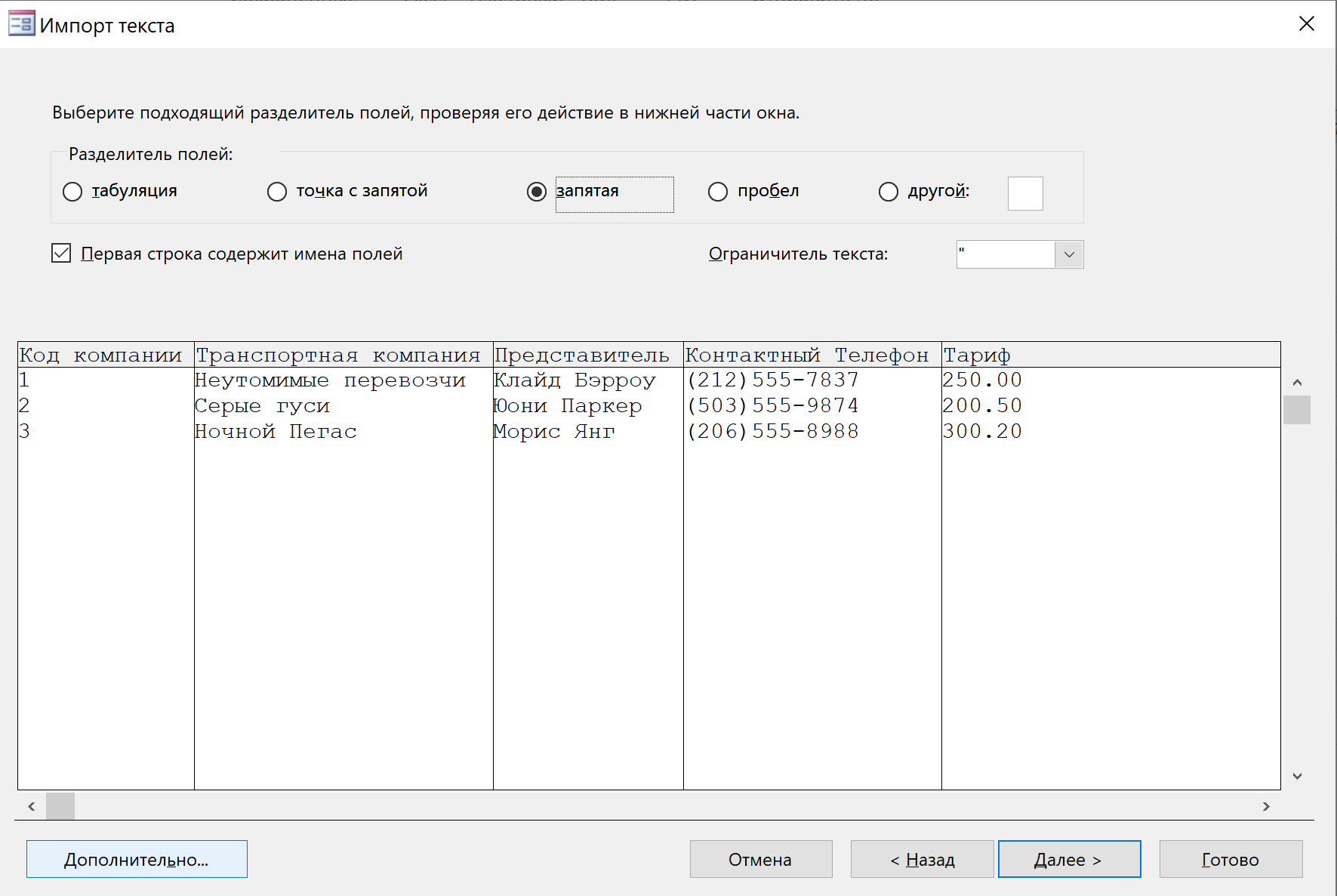




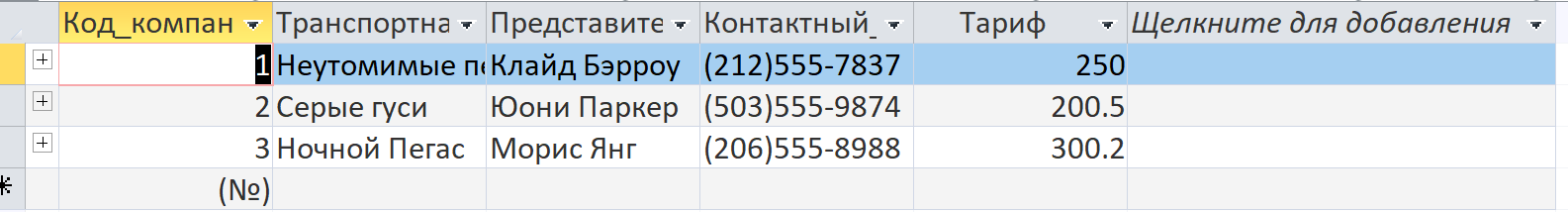
Итог импорта:



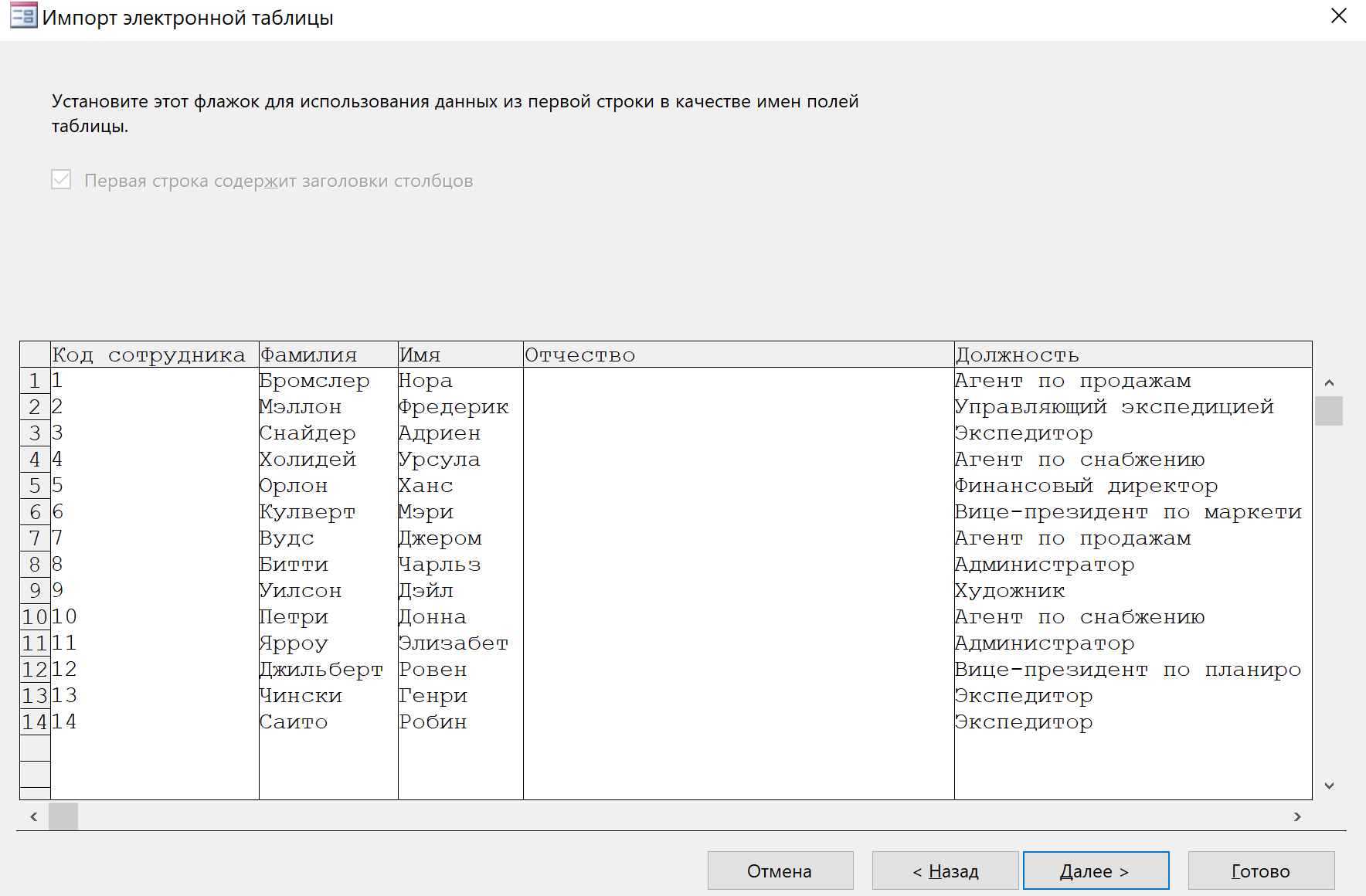
Импорт транспорта:



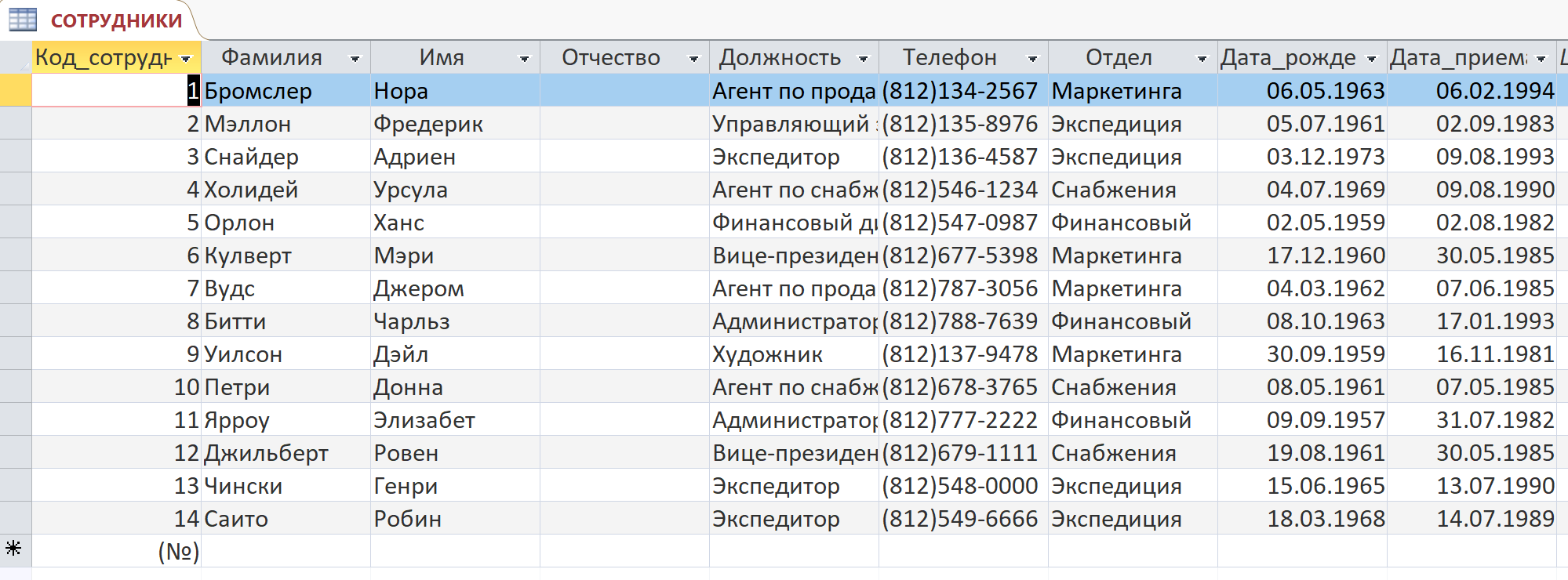
Итог импорта:



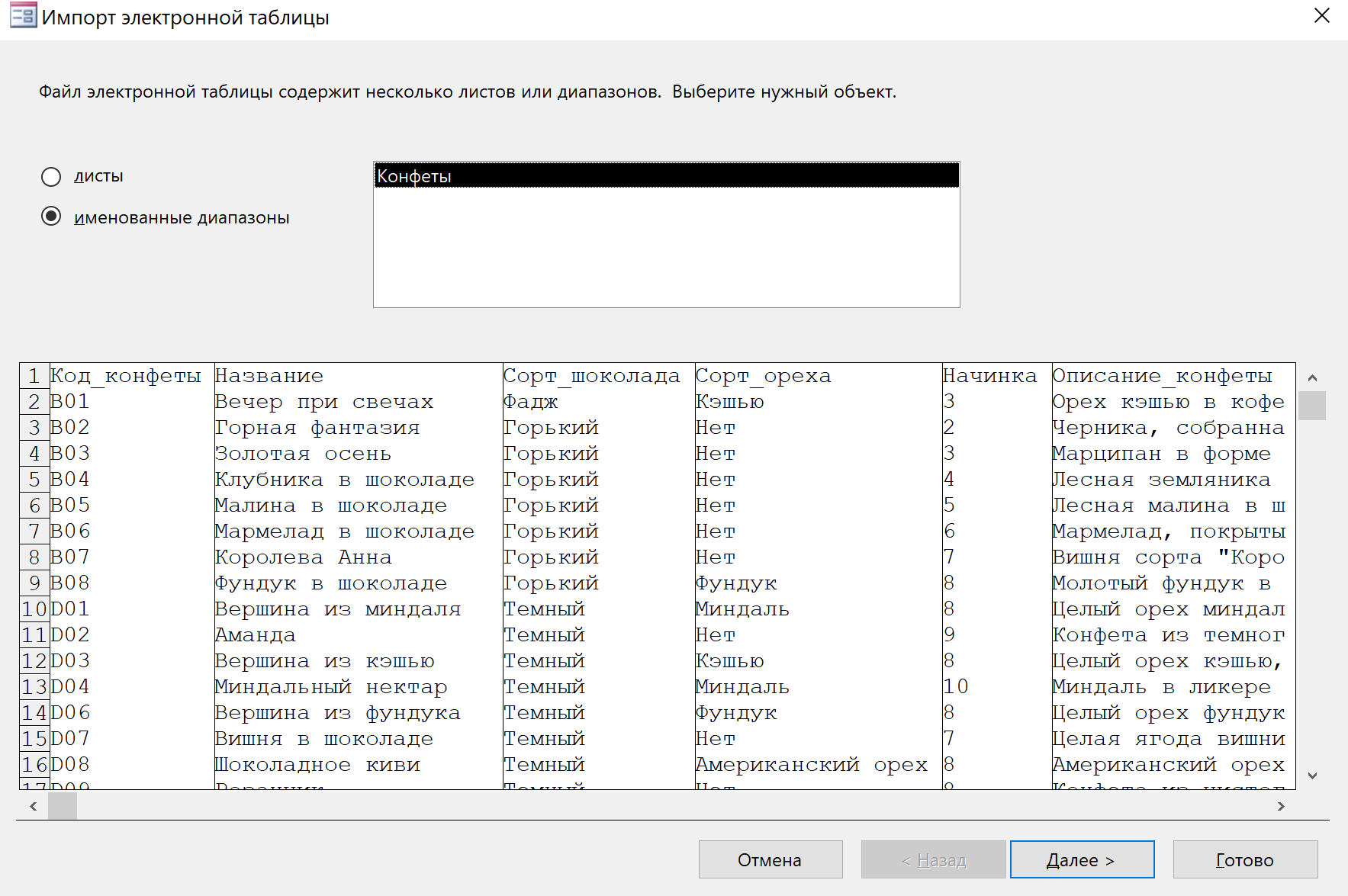
Импорт сотрудников:



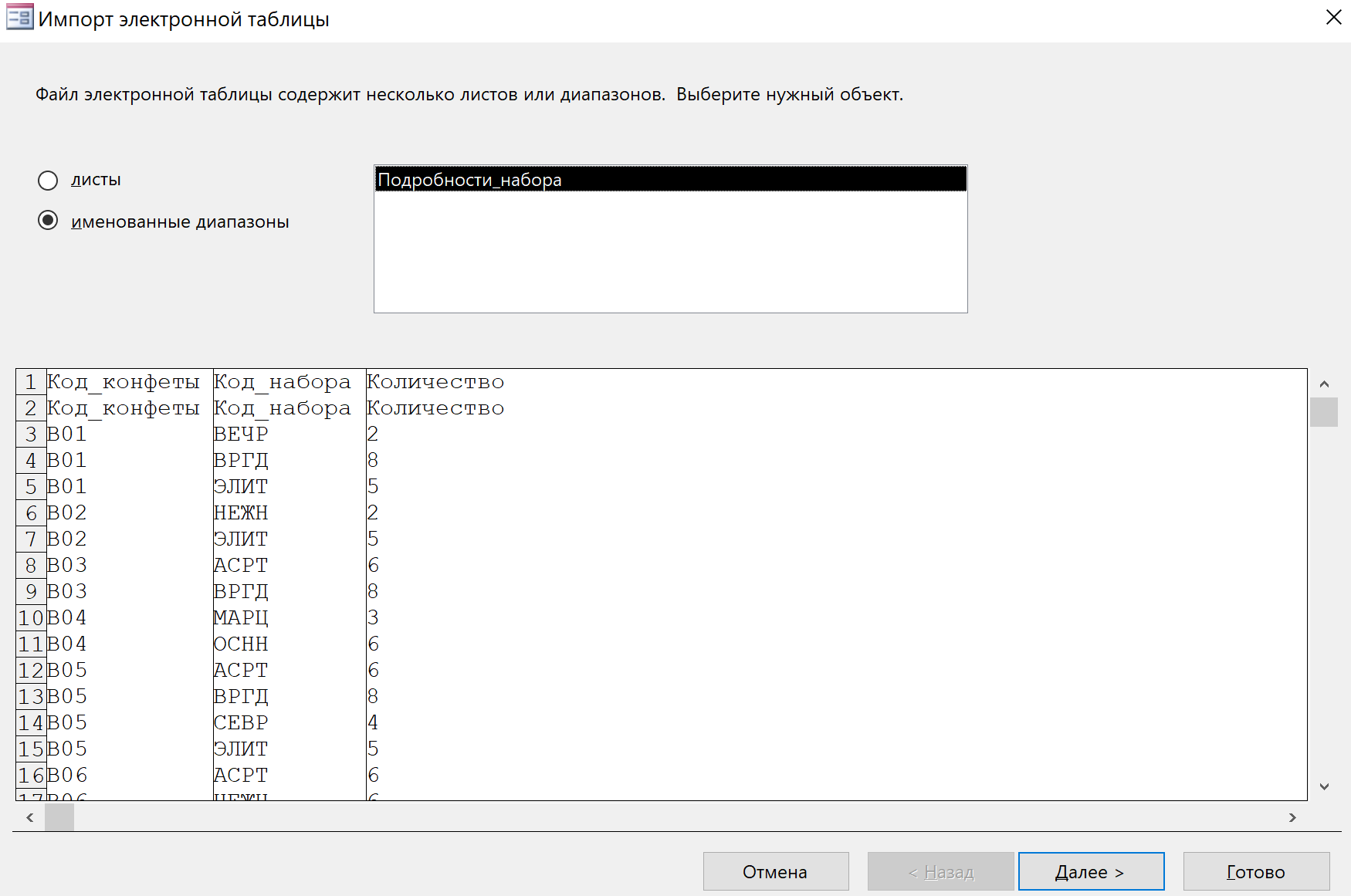
Итог импорта:



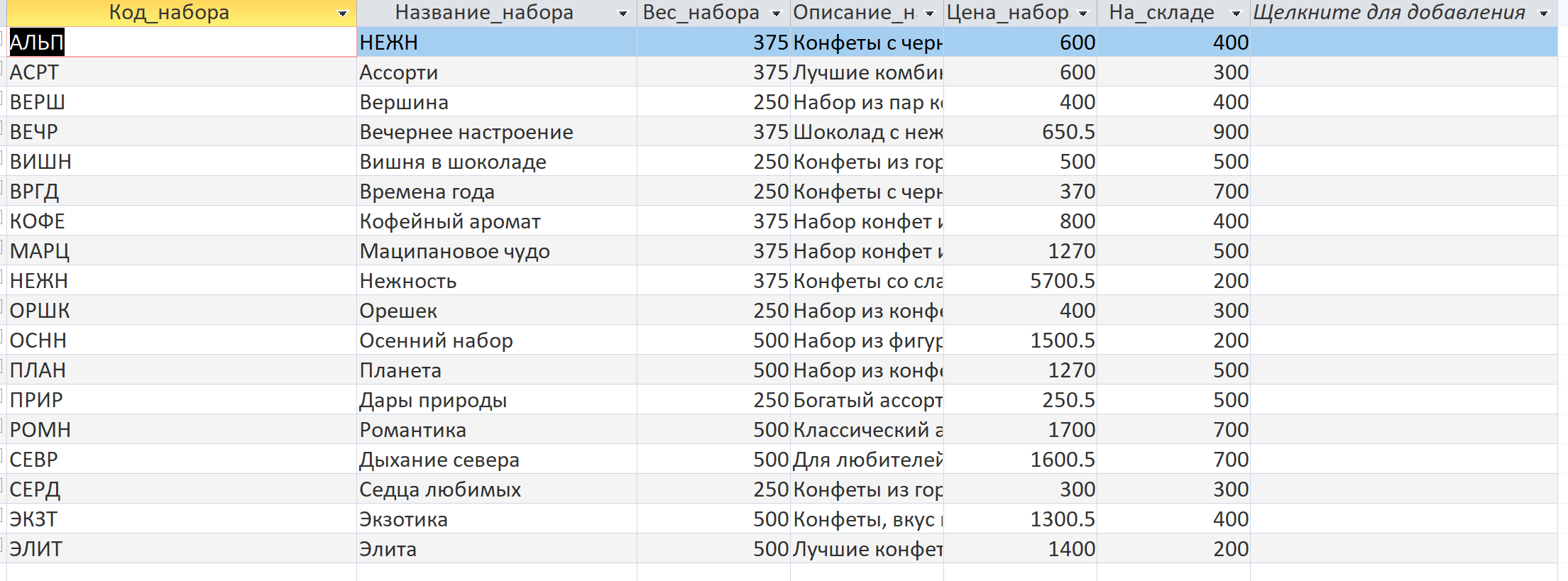
Импорт Конфеты:



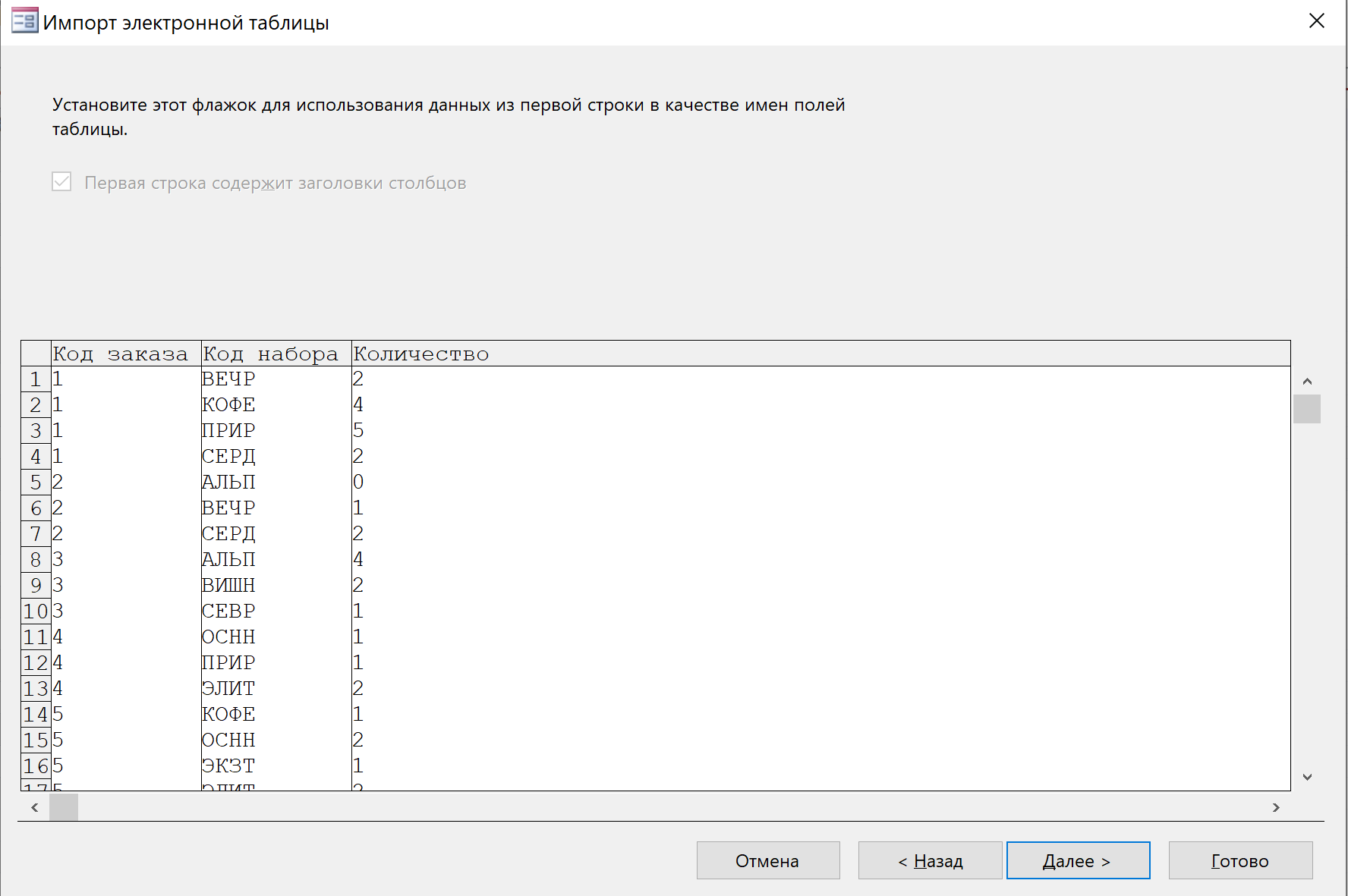
Импорт набора:



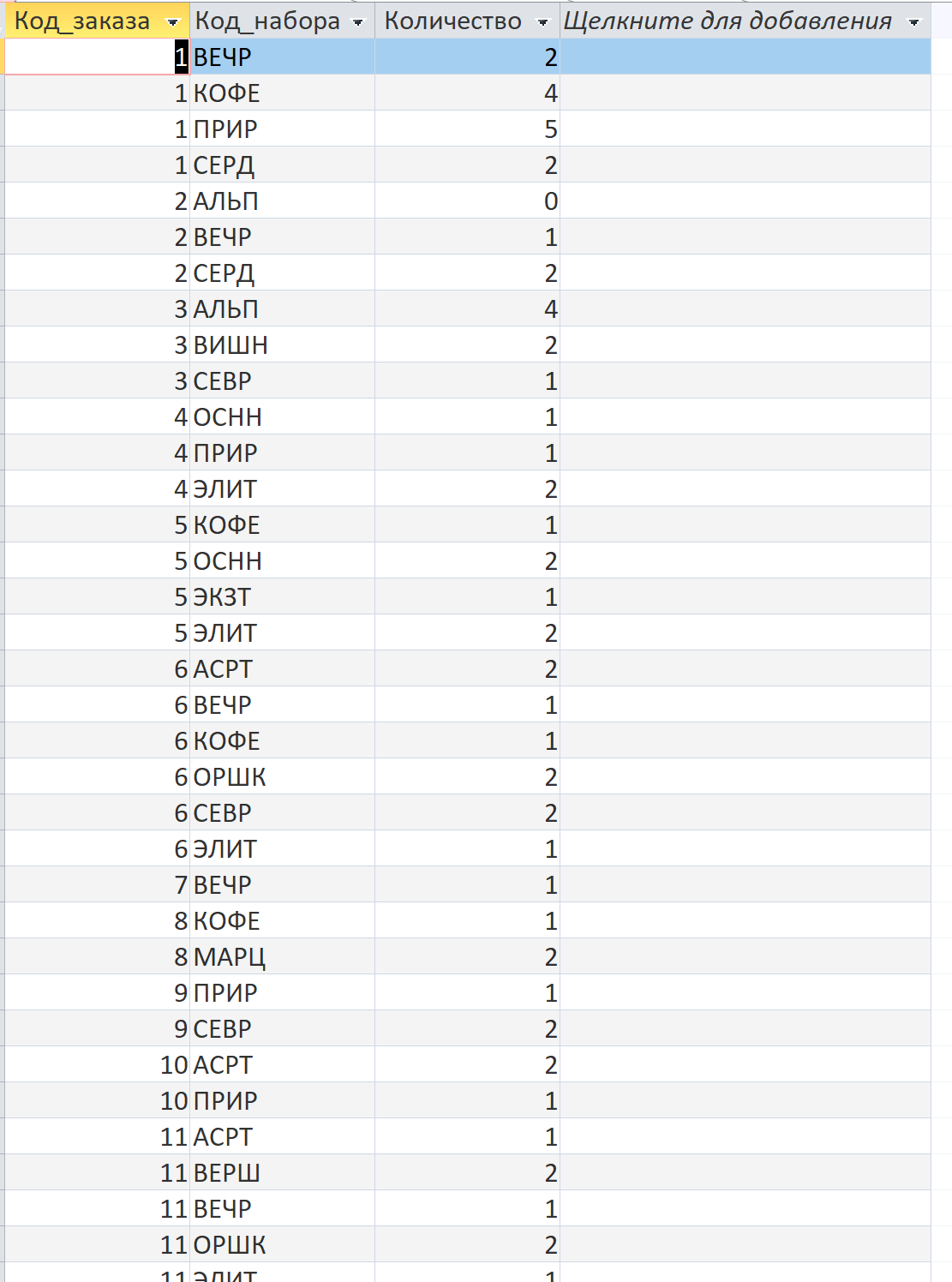
Итог импорта:



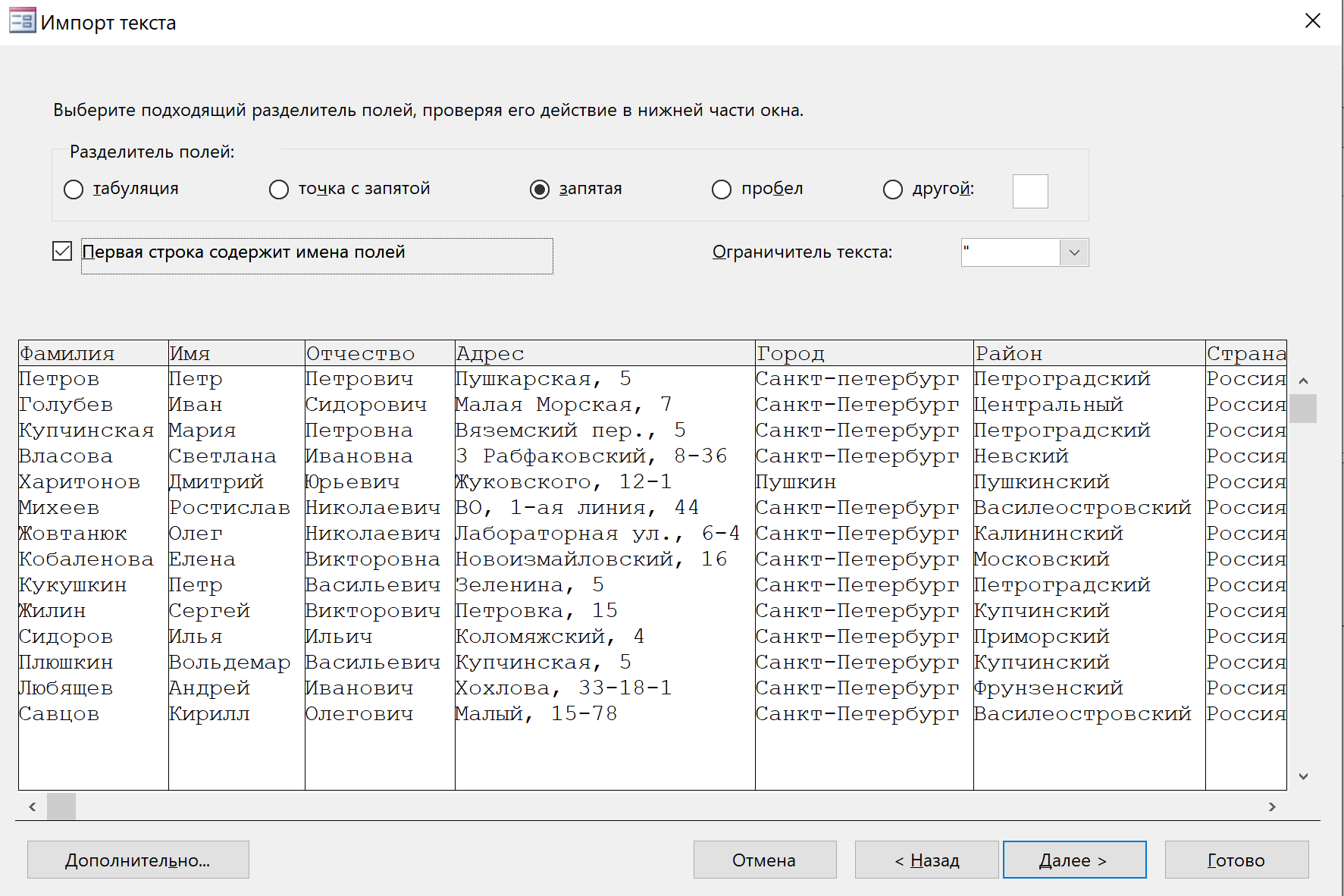
Импорт подробностей заказа:



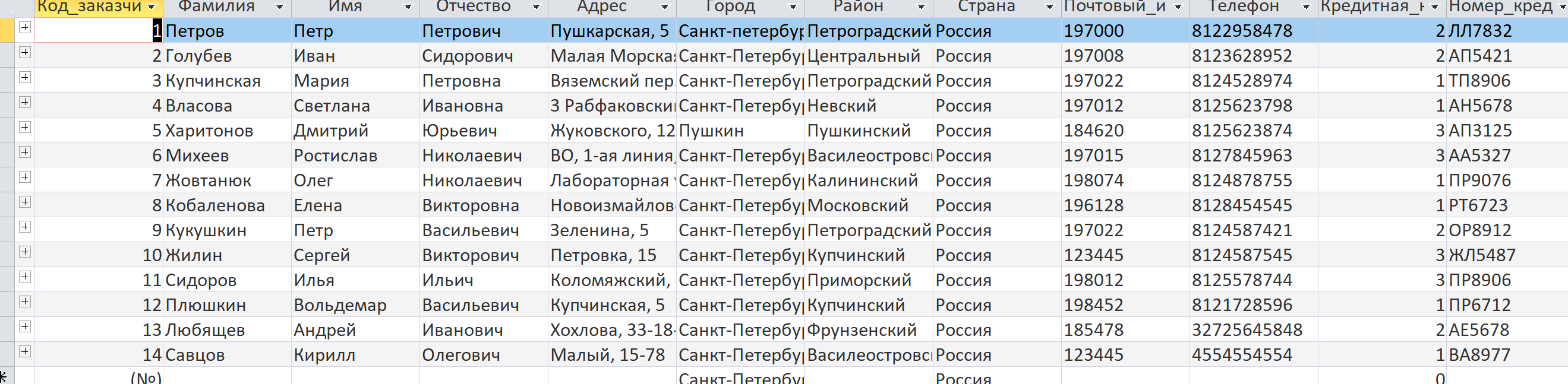
Итог импорта:



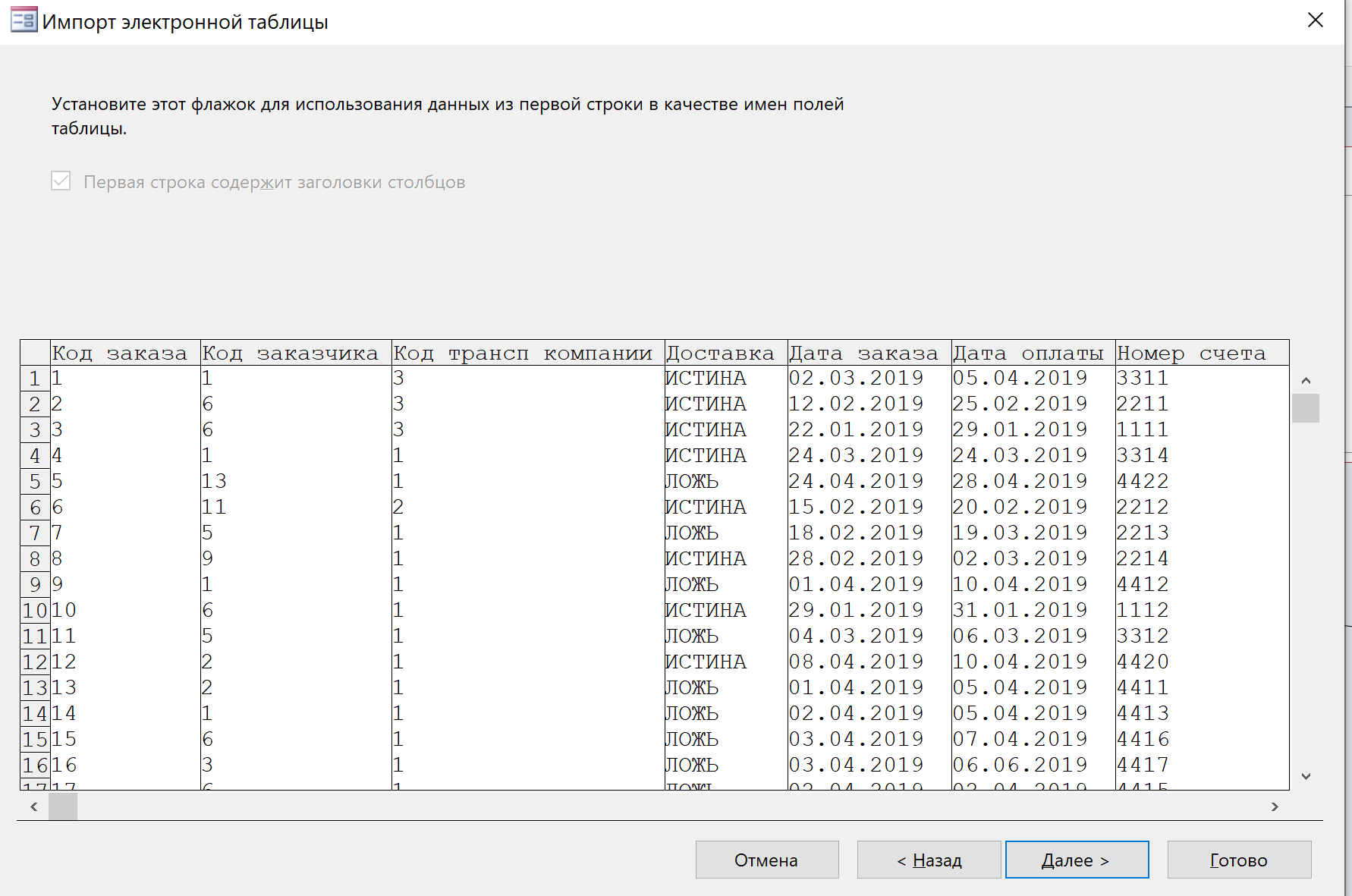
Импорт Заказчики:



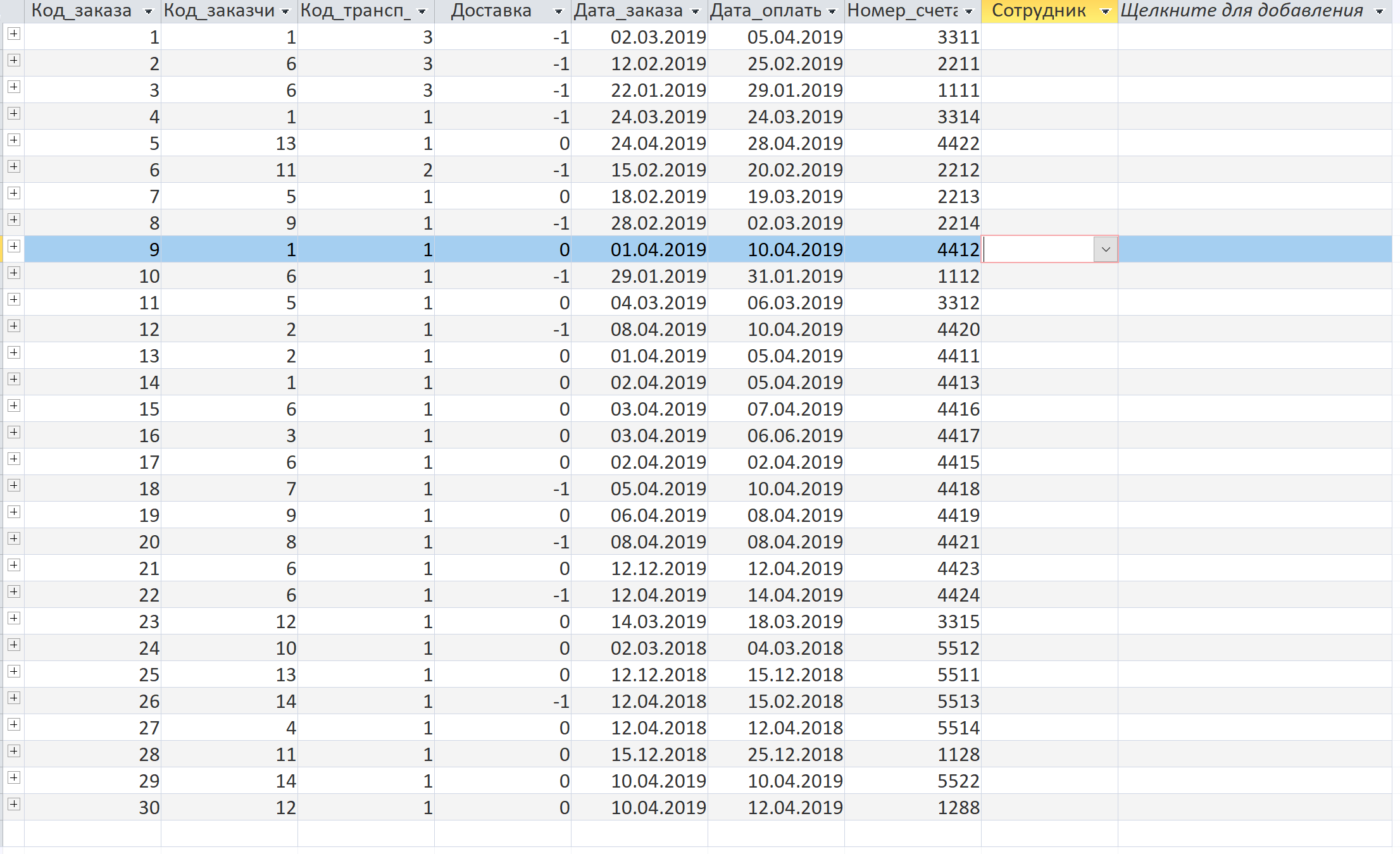
Итог импорта:



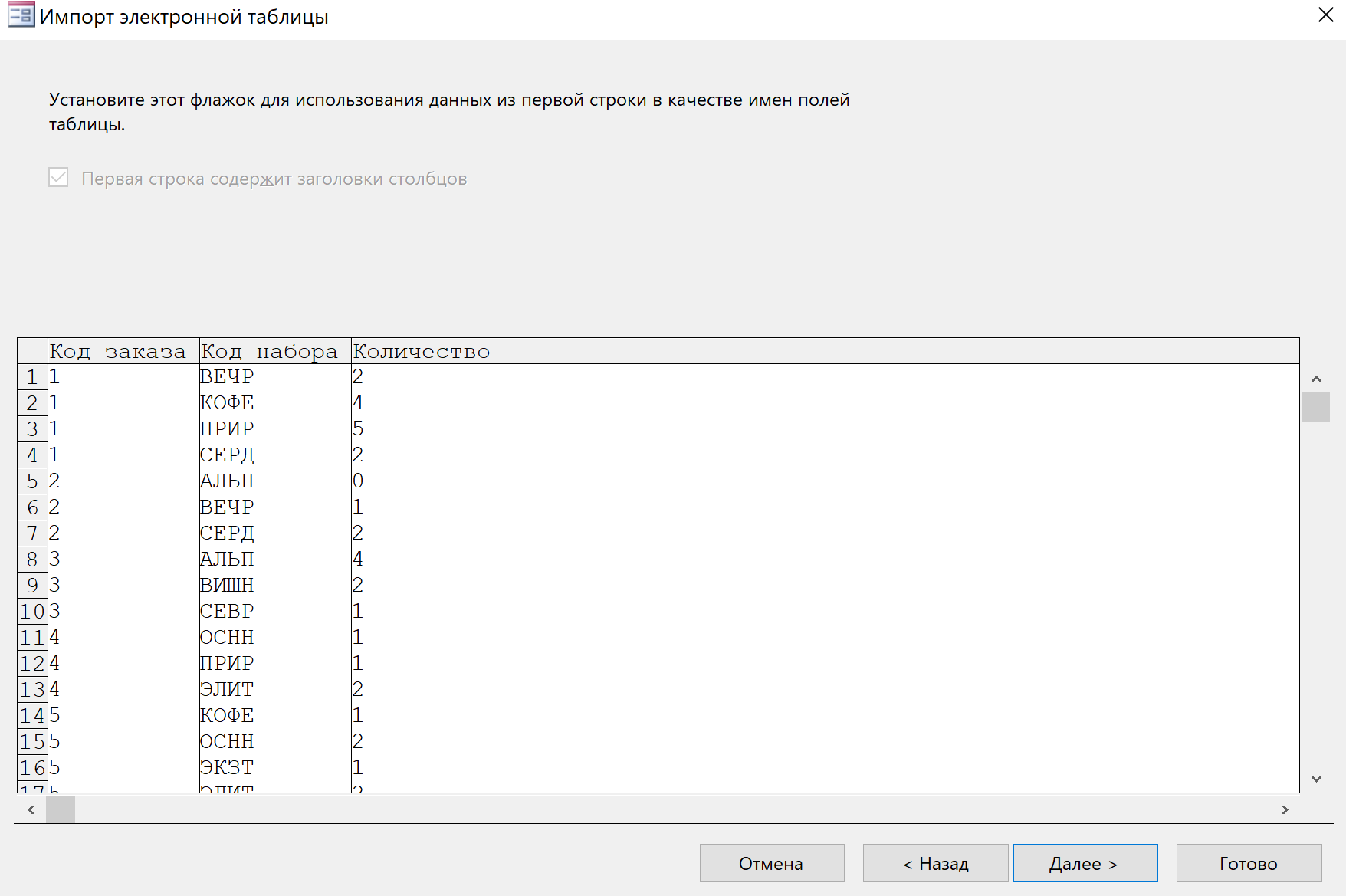
Импорт Заказы:



Итог импорта:



Импорт Подробности заказа:



Итог импорта:

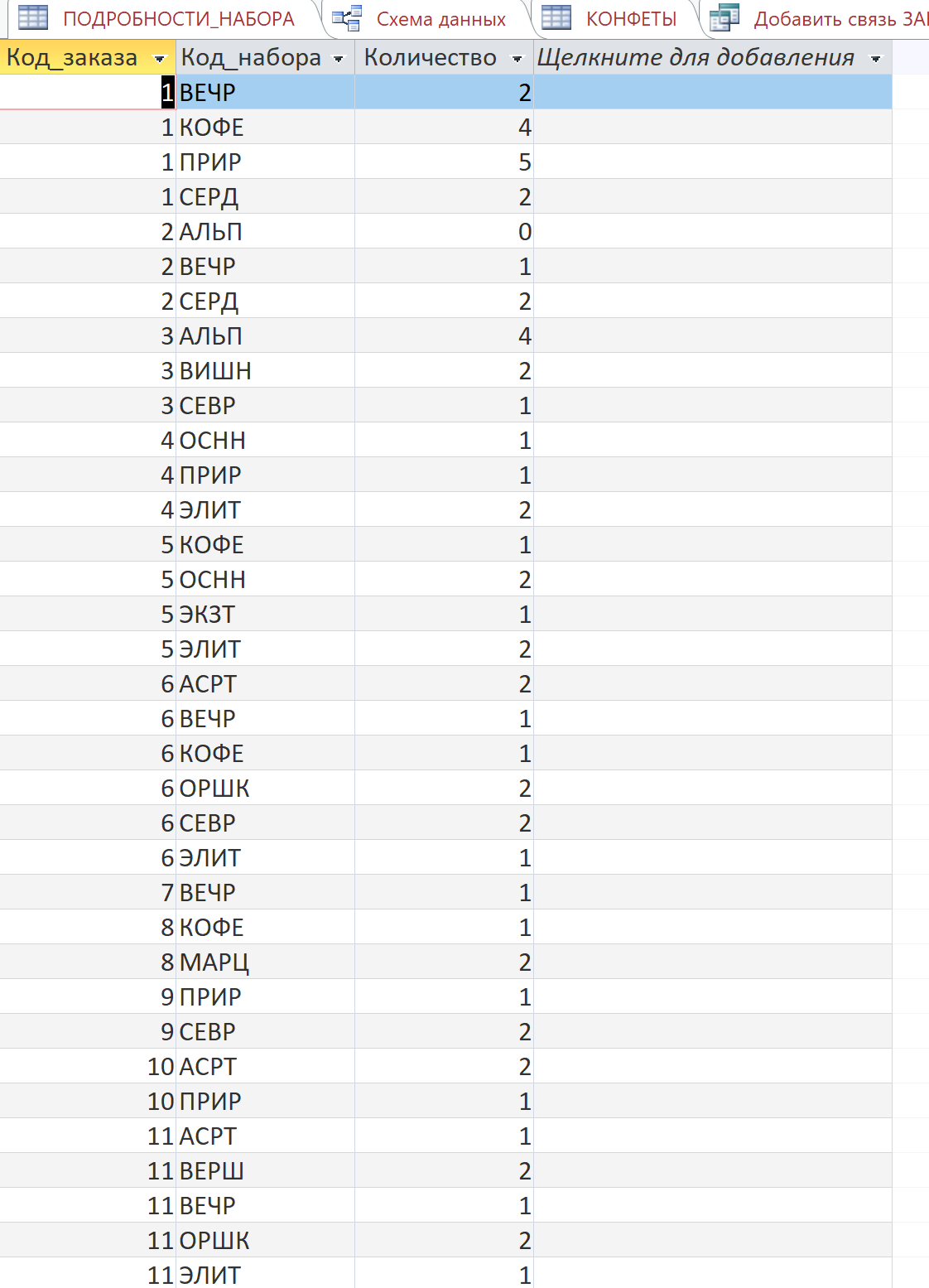
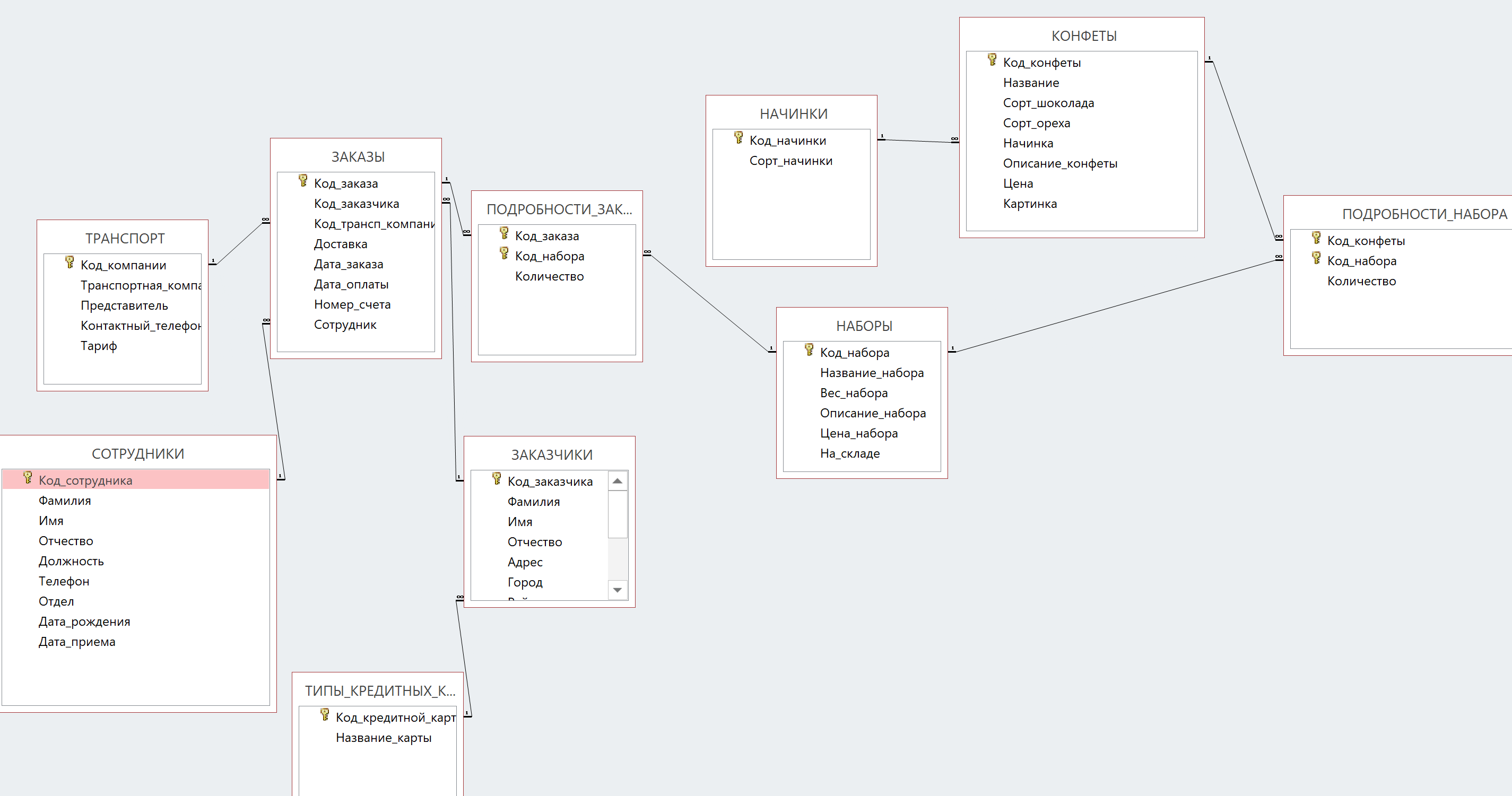


Схема:



Лабораторная работа 4

**Создание запросов к БД**

**Цель работы:** получить навыки в создании запросов следующих типов:

- однотабличные запросы на выборку;

- многотабличные запросы на выборку;

- перекрестные запросы;

- запросы-изменение.

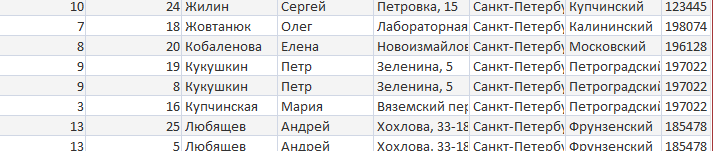
Таблица запросов

|  |  |  |
| --- | --- | --- |
| Список заказчиков |  | SELECT Район, Фамилия, Имя, Отчество, Адрес, Телефон, Почтовый\_индекс  FROM ЗАКАЗЧИКИ  Order by Район, Фамилия; |
| Список\_заказчиков\_К \_М |  | SELECT Район, Фамилия, Имя, Отчество, Адрес, Телефон, Почтовый\_индекс  FROM ЗАКАЗЧИКИ  Where Фамилия Between 'К' and 'М'  ORDER BY Район, Фамилия; |
| Фамилии\_на \_ов |  | SELECT Фамилия  FROM ЗАКАЗЧИКИ  Where Фамилия Like '%ов'; |
| Конфеты\_Условие\_И |  | SELECT Код\_конфеты, Название, Сорт\_шоколада, Сорт\_ореха, Цена  FROM КОНФЕТЫ  Where Сорт\_шоколада='Молочный' AND Конфеты.Цена > 30; |
| Конфеты\_Условие\_ИЛИ |  | SELECT Код\_конфеты, Название, Сорт\_шоколада, Сорт\_ореха, Цена  FROM КОНФЕТЫ  Where Сорт\_шоколада='Молочный' OR Конфеты.Цена > 30; |
| Заказы\_и\_Заказчики |  | SELECT Заказчики.Код\_заказчика, Заказы.Код\_заказа, Заказчики.Фамилия, Заказчики.Имя, Заказчики.Адрес, Заказчики.Город, Заказчики.Район, Заказчики.Почтовый\_индекс, Заказы.Дата\_заказа, Заказы.Дата\_оплаты, Заказы.Код\_трансп\_компании, Заказы.Доставка, Заказы.Номер\_счета, Заказчики.Кредитная\_карта, Заказчики.Номер\_кредитной\_карты, Заказчики.Телефон  FROM ЗАКАЗЧИКИ Inner JOIN ЗАКАЗЫ ON ЗАКАЗЧИКИ.Код\_заказчика = ЗАКАЗЫ.Код\_заказчика  ORDER BY Заказчики.Фамилия; |
| Заказчики\_из\_заданного\_района |  | SELECT \*  FROM ЗАКАЗЧИКИ  where Район Like [Введите начало названия района:] & '%'; |
| Состав\_и\_стоимость\_набора |  | SELECT Наборы.Название\_набора, КОНФЕТЫ.Название, КОНФЕТЫ.Цена, ПОДРОБНОСТИ\_НАБОРА.Количество, Цена\*Количество\*(1+0.2) AS Стоимость  FROM КОНФЕТЫ INNER JOIN (Наборы INNER JOIN ПОДРОБНОСТИ\_НАБОРА ON Наборы.Код\_набора = ПОДРОБНОСТИ\_НАБОРА.Код\_набора) ON КОНФЕТЫ.Код\_конфеты = ПОДРОБНОСТИ\_НАБОРА.Код\_конфеты  ORDER BY Наборы.Код\_набора; |
| Итог\_по\_Наборам |  | SELECT Наборы.Название\_набора, Sum(Цена\*Количество\*(1+0.2)) AS Стоимость  FROM КОНФЕТЫ INNER JOIN (Наборы INNER JOIN ПОДРОБНОСТИ\_НАБОРА ON Наборы.Код\_набора = ПОДРОБНОСТИ\_НАБОРА.Код\_набора) ON КОНФЕТЫ.Код\_конфеты = ПОДРОБНОСТИ\_НАБОРА.Код\_конфеты  GROUP BY Наборы.Название\_набора, Наборы.Код\_набора  ORDER BY Наборы.Код\_набора; |
| Итог\_по\_наборам\_с\_кол\_типов\_конфет |  | SELECT Наборы.Название\_набора, Count(Наборы.Название\_набора) AS Количество\_типов\_конфет, Sum(Цена\*Количество\*(1+0.2)) AS Стоимость  FROM КОНФЕТЫ INNER JOIN (Наборы INNER JOIN ПОДРОБНОСТИ\_НАБОРА ON Наборы.Код\_набора = ПОДРОБНОСТИ\_НАБОРА.Код\_набора) ON КОНФЕТЫ.Код\_конфеты = ПОДРОБНОСТИ\_НАБОРА.Код\_конфеты  GROUP BY Наборы.Название\_набора, Наборы.Код\_набора, Наборы.Название\_набора  ORDER BY Наборы.Код\_набора; |
| Итоговая\_стоимость\_набора |  | SELECT Наборы.Название\_набора, Sum(ПОДРОБНОСТИ\_НАБОРА.Количество) AS [Sum-Количество], Sum(Конфеты.Цена\*ПОДРОБНОСТИ\_НАБОРА.Количество)\*(1+0.2) AS Стоимость  FROM КОНФЕТЫ INNER JOIN (Наборы INNER JOIN ПОДРОБНОСТИ\_НАБОРА ON Наборы.Код\_набора = ПОДРОБНОСТИ\_НАБОРА.Код\_набора) ON КОНФЕТЫ.Код\_конфеты = ПОДРОБНОСТИ\_НАБОРА.Код\_конфеты  GROUP BY Наборы.Название\_набора  ORDER BY Наборы.Название\_набора; |
| Фамилия\_группировка |  | SELECT ЗАКАЗЧИКИ.Фамилия, Наборы.Название\_набора, Наборы.Цена\_набора, Sum(ПОДРОБНОСТИ\_ЗАКАЗА.Количество) AS Количество\_наборов, ЗАКАЗЧИКИ.Код\_заказчика  FROM ЗАКАЗЧИКИ INNER JOIN (Наборы INNER JOIN (ЗАКАЗЫ INNER JOIN ПОДРОБНОСТИ\_ЗАКАЗА ON ЗАКАЗЫ.Код\_заказа = ПОДРОБНОСТИ\_ЗАКАЗА.Код\_заказа) ON Наборы.Код\_набора = ПОДРОБНОСТИ\_ЗАКАЗА.Код\_набора) ON ЗАКАЗЧИКИ.Код\_заказчика = ЗАКАЗЫ.Код\_заказчика  GROUP BY ЗАКАЗЧИКИ.Фамилия, Наборы.Название\_набора, Наборы.Цена\_набора, ЗАКАЗЧИКИ.Код\_заказчика  HAVING Наборы.Название\_набора="Ассорти" Or Наборы.Название\_набора="Орешек"  ORDER BY ЗАКАЗЧИКИ.Код\_заказчика; |
| Фамилия\_именование\_группировка |  | SELECT Фамилия, Название\_набора, Цена\_набора, Количество\_наборов, Sum([Цена\_набора]\*[Количество\_наборов]) AS Сумма  FROM Фамилия\_группировка\_все\_наборы  GROUP BY Фамилия\_группировка\_все\_наборы.Фамилия, Фамилия\_группировка\_все\_наборы.Название\_набора, Фамилия\_группировка\_все\_наборы.Цена\_набора, Фамилия\_группировка\_все\_наборы.Количество\_наборов; |
| Конфеты в наборе -перекрестный |  | TRANSFORM Sum(Подробности\_набора.Количество) AS [Sum-Количество]  SELECT Наборы.Название\_набора, Sum(Подробности\_набора.Количество) AS [Общее количество]  FROM Наборы INNER JOIN (Конфеты INNER JOIN Подробности\_набора ON Конфеты.Код\_конфеты=Подробности\_набора.Код\_конфеты) ON Наборы.Код\_набора = Подробности\_набора.Код\_набора  GROUP BY Наборы.Название\_набора  PIVOT Конфеты.Название; |
| Конфеты\_с\_орехами\_ | Запрос на изменение | SELECT Код\_конфеты, Название, Сорт\_шоколада, Сорт\_ореха, Начинка, Описание\_конфеты, Цена, Картинка  Into КОНФЕТЫ\_С\_ОРЕХАМИ  FROM КОНФЕТЫ  WHERE КОНФЕТЫ.Сорт\_ореха<>"Нет"  ORDER BY КОНФЕТЫ.Название; |
| Конфеты\_удаление | Запрос на удаление | DELETE Цена  FROM КОНФЕТЫ\_С\_ОРЕХАМИ  Where Цена>30; |
| Конфеты\_новая\_цена | Запрос на изменение | Update КОНФЕТЫ\_С\_ОРЕХАМИ SET Цена = Цена\*2; |

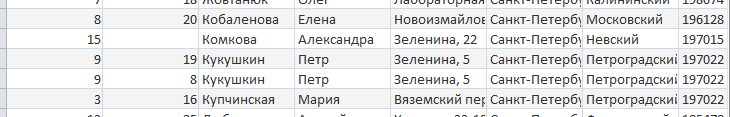
Задание 2.

Многотабличные запросы на выборку

1) Комковой нет:

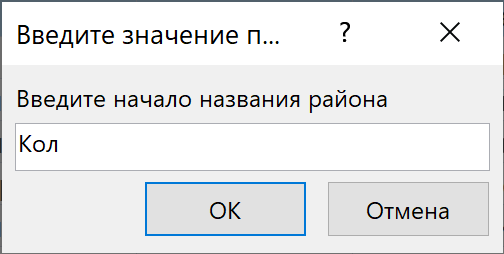


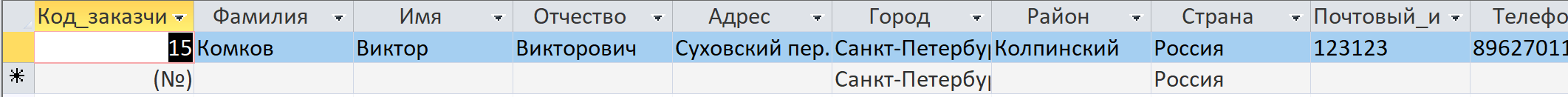
2) Left join:



Задание 3

Параметрические запросы



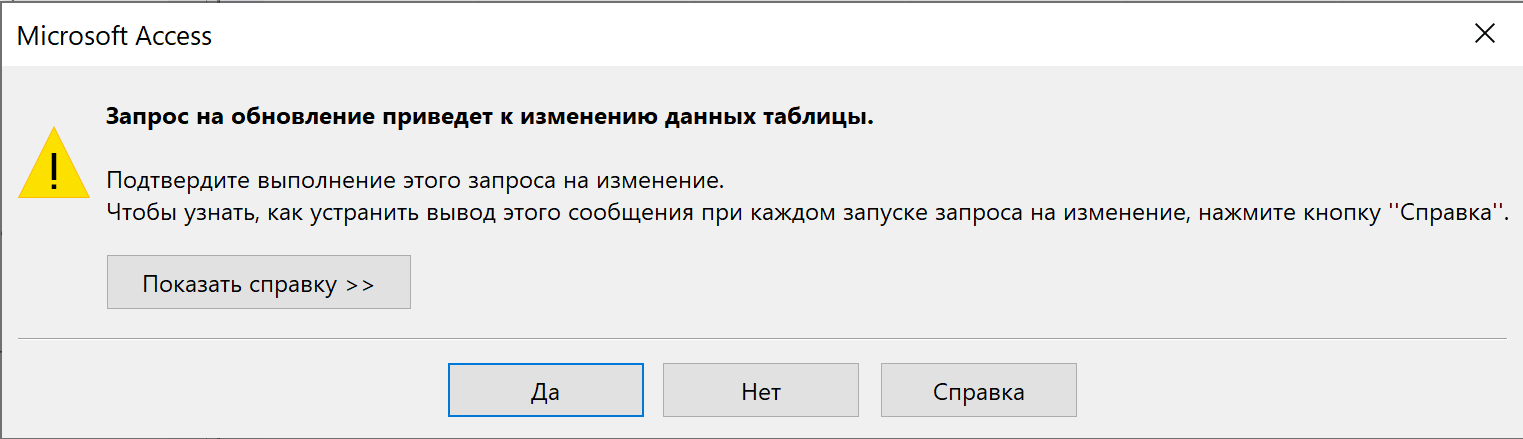


Запрос на обновление записей:

UPDATE Конфеты\_с\_орехами SET Цена = Цена \* 2;

Таблица до обновления:





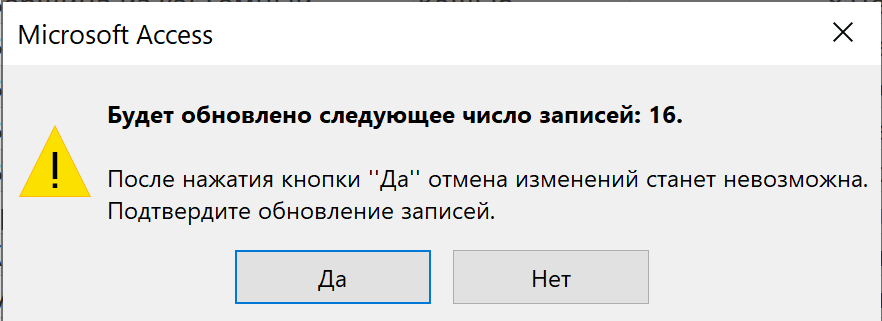


Таблица после обновления:



**Отчет по контрольной работе.**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Задание | Листинг |
| КР6\_1 | Перекрёстный | Создать перекрёстный запрос, отображающий по месяцам, сколько договоров заключил каждый сотрудник | TRANSFORM COUNT(СОТРУДНИКИ.Код\_сотрудника)  SELECT Фамилия  FROM СОТРУДНИКИ INNER JOIN ЗАКАЗЫ ON СОТРУДНИКИ.КОД\_СОТРУДНИКА = ЗАКАЗЫ.Сотрудник  GROUP BY СОТРУДНИКИ.Фамилия  PIVOT MONTHNAME(MONTH(ДАТА\_ЗАКАЗА)); |
| КР6\_2 | На выборку | Создать запрос вычисляющий общее количество заказанных наборов | SELECT НАБОРЫ.Название\_набора, SUM(Количество) AS [Количество заказанных наборов]  FROM ПОДРОБНОСТИ\_ЗАКАЗА INNER JOIN НАБОРЫ ON НАБОРЫ.Код\_набора = ПОДРОБНОСТИ\_ЗАКАЗА.Код\_Набора  GROUP BY НАБОРЫ.Название\_набора; |
| КР6\_3 | На выборку | Вывести информацию о сотруднике с минимальным стажем работы | SELECT Код\_сотрудника, Фамилия, Имя, DateDiff("d", Дата\_приема, Date()) AS [Дней с принятия на работу]  FROM СОТРУДНИКИ  WHERE DateDiff("d", Дата\_приема, Date()) = (SELECT MIN(DateDiff("d", Дата\_приема, Date())) FROM СОТРУДНИКИ); |
| КР6\_4 | На выборку | Вывести в алфавитном порядке фамилии заказчиков с инициалами (в одном поле) не пользующихся услугами транспортными компаний | SELECT (ФАМИЛИЯ & " " & MID(ИМЯ, 1, 1) & "." & MID(ОТЧЕСТВО, 1, 1) & ".") AS Инициалы  FROM ЗАКАЗЧИКИ INNER JOIN ЗАКАЗЫ ON ЗАКАЗЫ.Код\_Заказчика = ЗАКАЗЧИКИ.Код\_Заказчика  WHERE Доставка = -1  GROUP BY (ФАМИЛИЯ & " " & MID(ИМЯ, 1, 1) & "." & MID(ОТЧЕСТВО, 1, 1) & ".")  ORDER BY (ФАМИЛИЯ & " " & MID(ИМЯ, 1, 1) & "." & MID(ОТЧЕСТВО, 1, 1) & "."); |
| КР6\_5\_1 | На создание таблицы | Создайте новую таблицу «конфеты с фисташками», содержащую список конфет с фисташками | SELECT \* INTO [конфеты с фисташками]  FROM КОНФЕТЫ  WHERE Сорт\_ореха = "Фисташки"; |
| КР6\_5\_2 | Обновление таблицы | Создайте запрос, который увеличит на 15 процентов цену таких конфет в созданной таблице. | UPDATE [конфеты с фисташками] SET Цена = Цена \* 1.15; |
| КР7\_6 | Выборка | Вывести по месяцам рождения фамилию\_имя\_отчество(в одном поле) сотрудника, место его работы и должность | SELECT MONTHNAME(MONTH(Дата\_рождения)) AS Месяц\_рождения, (Фамилия & "\_" & Имя & "\_" & Отчество) AS фамилия\_имя\_отчетство, Отдел, Должность  FROM СОТРУДНИКИ  ORDER BY MONTH(Дата\_рождения); |

**Лабораторная работа 5**

**Создание форм в MS ACCESS**

Цель работы: 1. Получить навыки создания форм в MS ACCESS с помощью мастера: форма в один столбец ; ленточная форма; составная форма;

2. Создание и редактирование форм в режиме конструктора: простой формы ; составной (главной и подчиненной) формы.

Способы создания форм:

1) С помощью функциональной кнопки

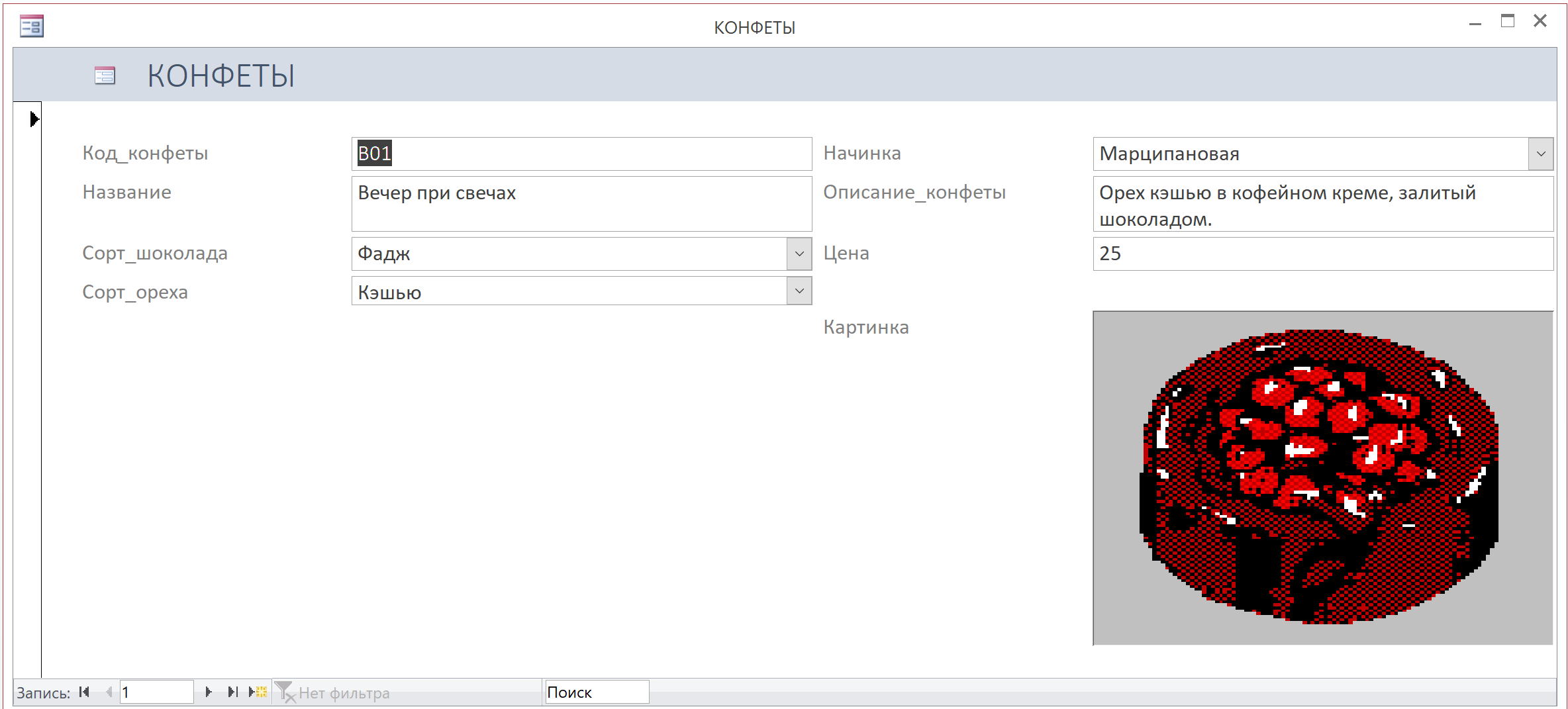
2) Через «Мастер форм»

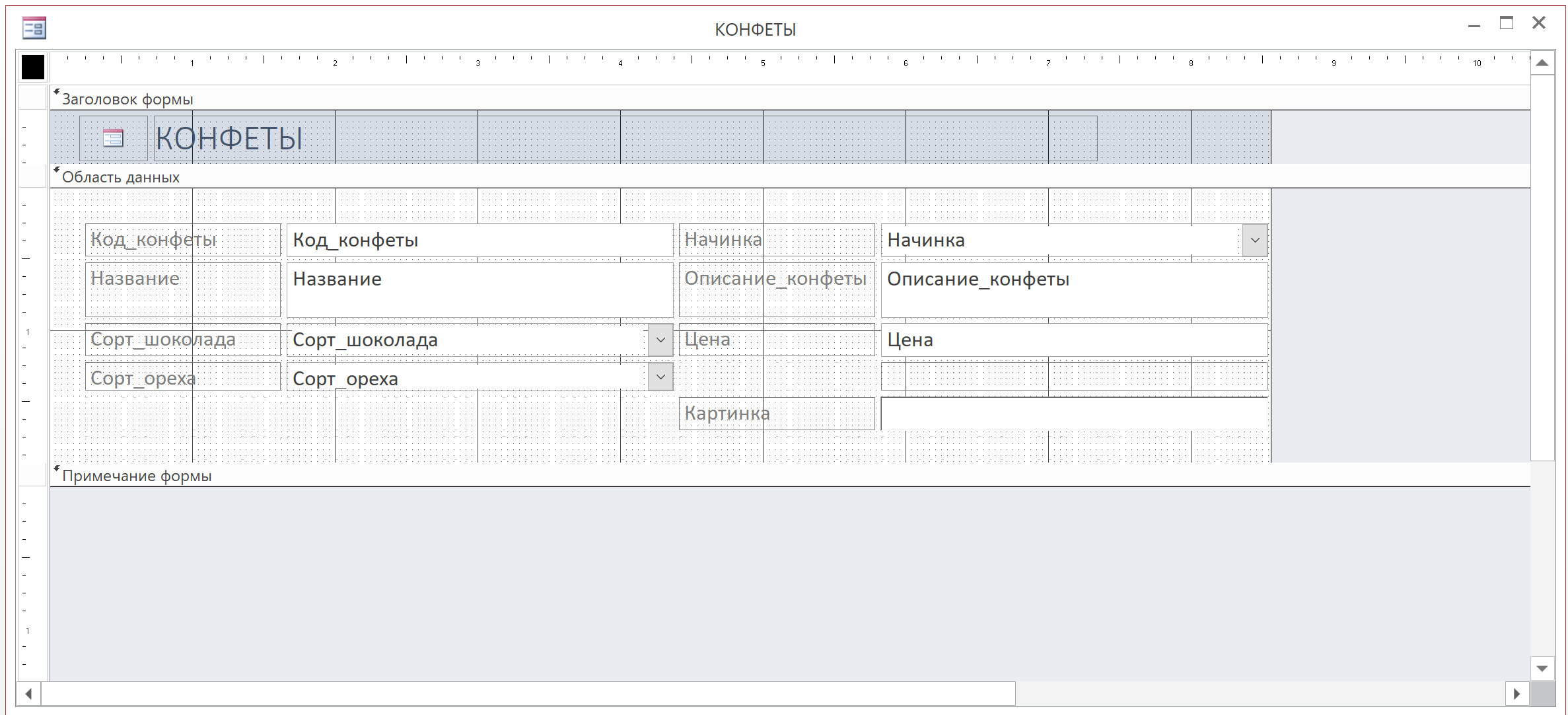
3) Через функциональную кнопку «Несколько элементов»

|  |  |  |  |
| --- | --- | --- | --- |
| Название формы | Способ создания | Источник данных | Вычисляемые поля |
| КОНФЕТЫ | Через функциональную кнопку | Таблица «КОНФЕТЫ» | Нет |
| ТРАНСПОРТ | Мастер | Таблица ТРАНСПОРТ | Нет |
| ЗАКАЗЧИК И ЕГО ЗАКАЗЫ | С помощью мастера | Таблица ЗАКАЗЧИКИ | Поле «Стоимость»,  «Общая стоимость» |
| Заказы\_подчиненная | Через функциональную кнопку «Несколько элементов» | Запрос «Заказы\_подчиненный» | «Стоимость»,  «Общая стоимость» |
| ЗАКАЗЧИКИ | Мастер | Таблица «ЗАКАЗЧИКИ» | Нет |
| ПОДРОБНОСТИ\_ЗАКАЗА | Мастер | Запрос:  SELECT ПОДРОБНОСТИ\_ЗАКАЗА.Код\_заказа, ПОДРОБНОСТИ\_ЗАКАЗА.Код\_набора, Наборы.Название\_набора, ПОДРОБНОСТИ\_ЗАКАЗА.Количество, Наборы.Цена\_набора, [Количество]\*[Цена\_набора] AS Стоимость FROM Наборы INNER JOIN ПОДРОБНОСТИ\_ЗАКАЗА ON Наборы.Код\_набора = ПОДРОБНОСТИ\_ЗАКАЗА.Код\_набора; | «Стоимость», «Общая стоимость» |
| Оформление заказа | Через функциональную кнопку | Таблица «ЗАКАЗЫ» | «Цена заказа» |
| НАБОРЫ | Мастер | Таблица «Наборы» | «Себестоимость набора» |
| НАБОРЫ ПОДЧИНЕННАЯ | Через функциональную кнопку | Запрос:  SELECT [ПОДРОБНОСТИ\_НАБОРА].[Код\_набора], [ПОДРОБНОСТИ\_НАБОРА].[Код\_конфеты], [ПОДРОБНОСТИ\_НАБОРА].[Количество], [КОНФЕТЫ].[Название], [КОНФЕТЫ].[Сорт\_шоколада], [КОНФЕТЫ].[Сорт\_ореха], [КОНФЕТЫ].[Начинка], [НАЧИНКИ].[Сорт\_начинки], [КОНФЕТЫ].[Цена] FROM (НАЧИНКИ INNER JOIN КОНФЕТЫ ON НАЧИНКИ.Код\_начинки=КОНФЕТЫ.Начинка) INNER JOIN | «стоимость конфет» |

Скриншоты:

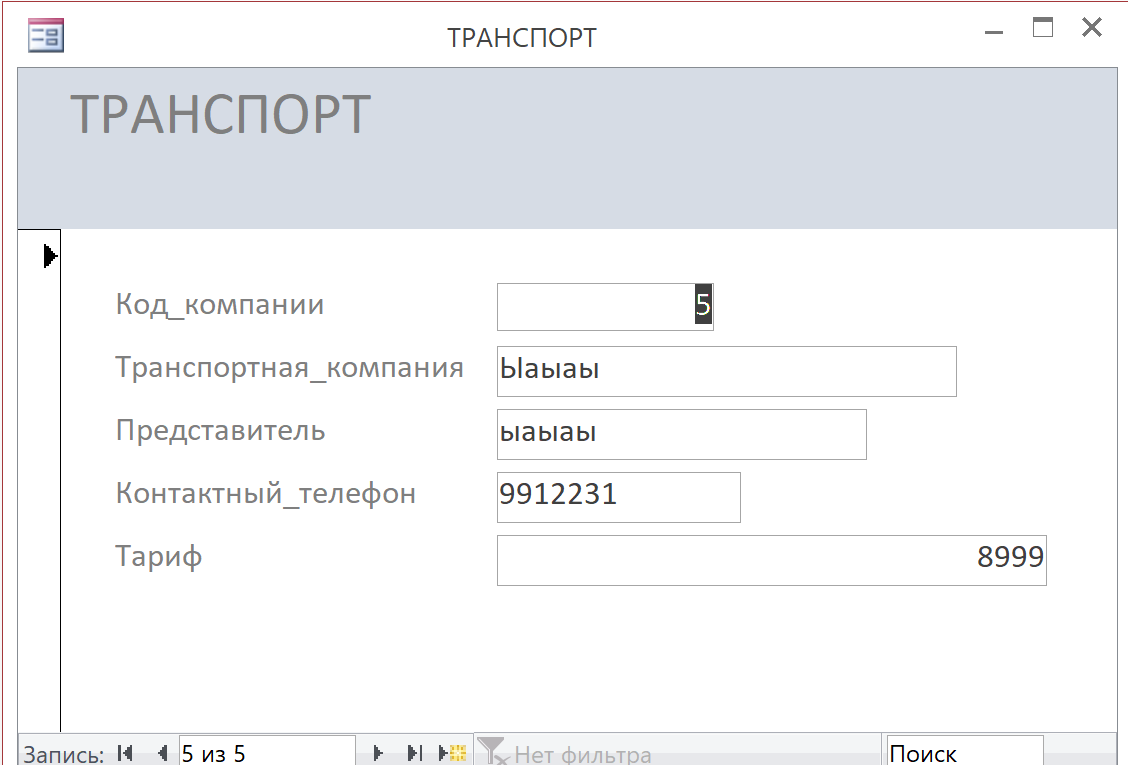
1. Конфеты:

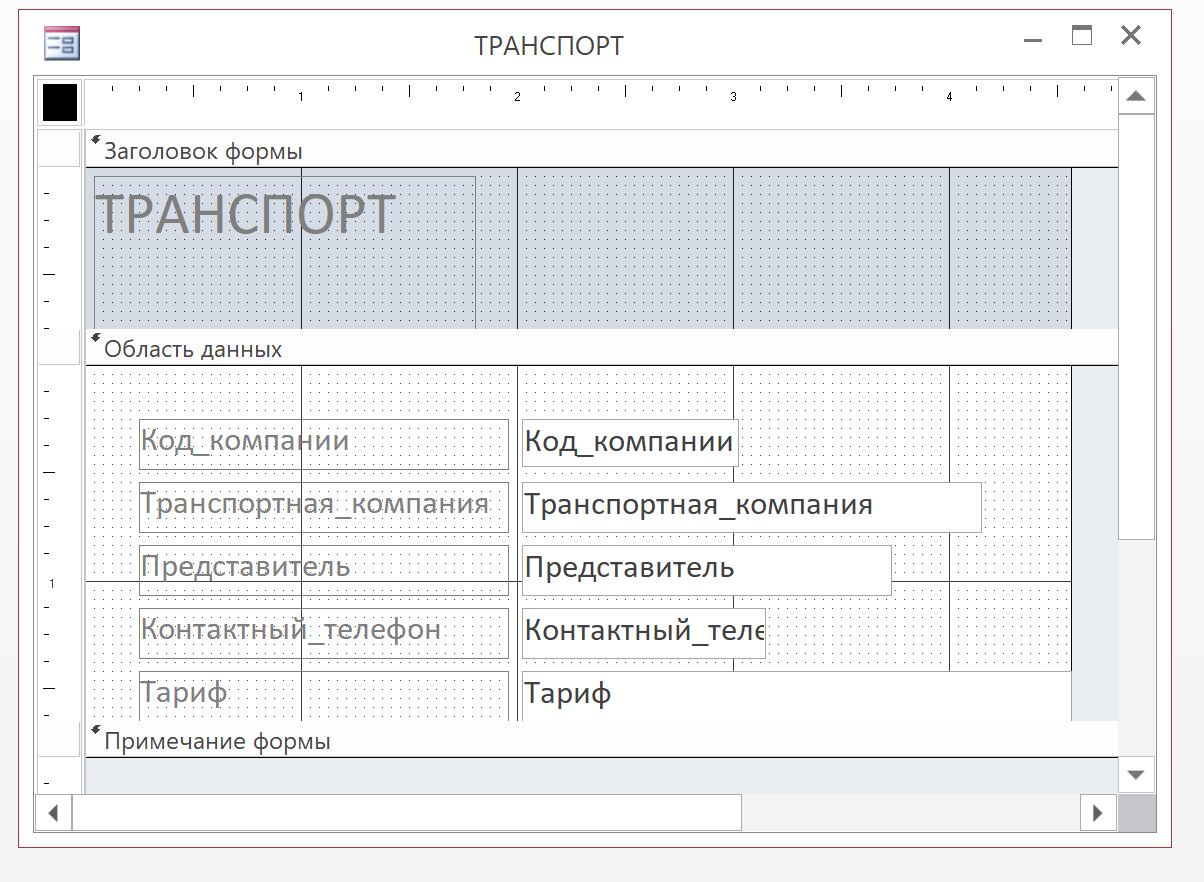




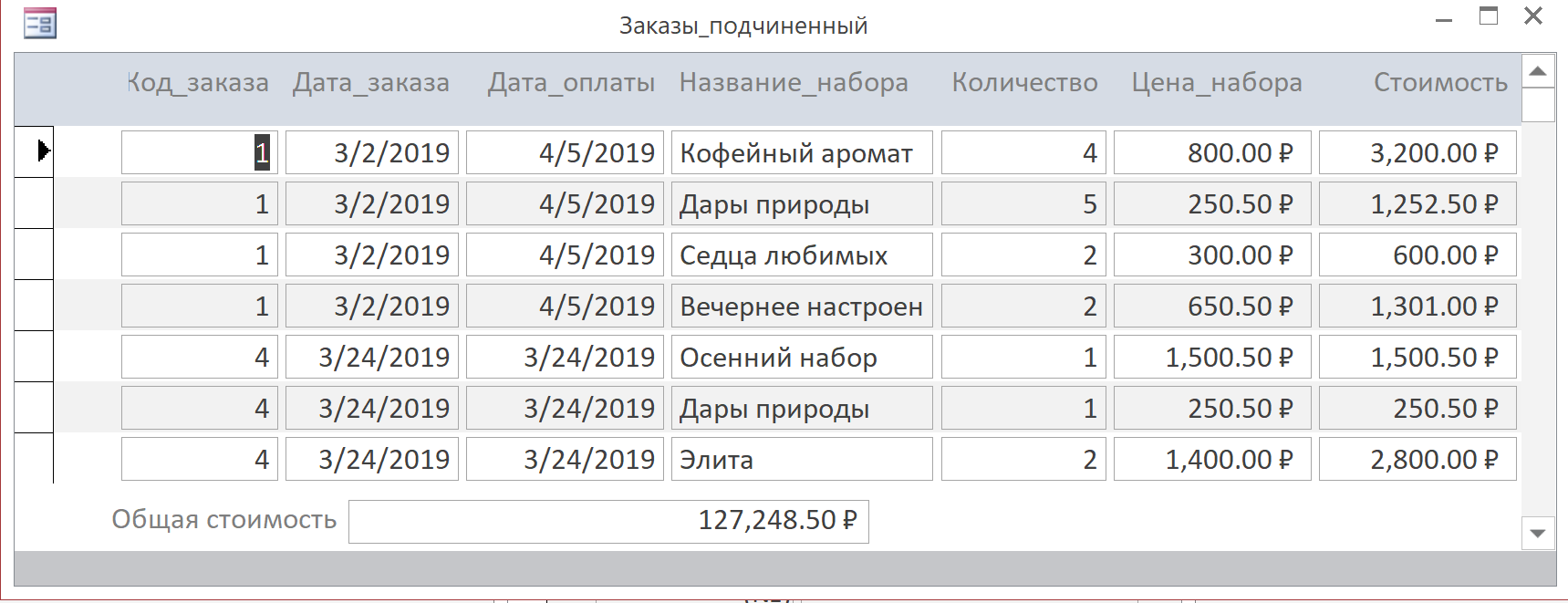
Комментарий: Ради эксперимента, я сделал интерфейс адаптивным.

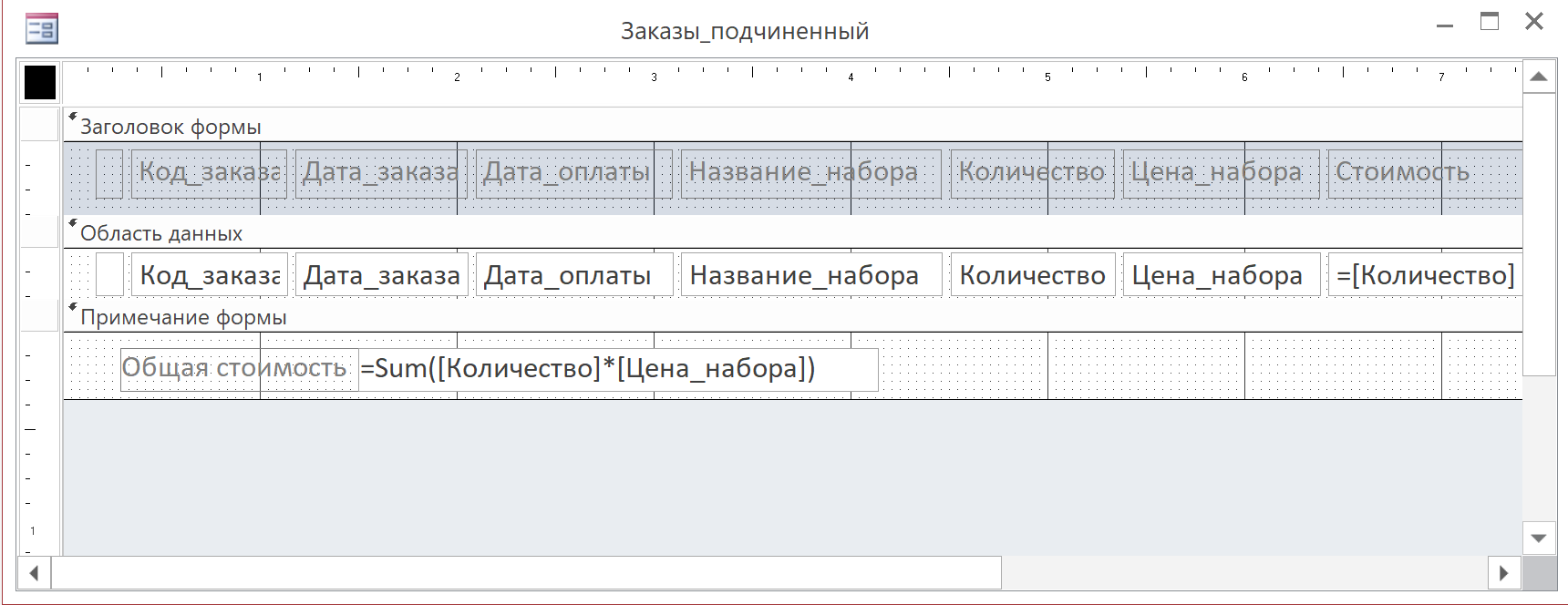
Транспорт:



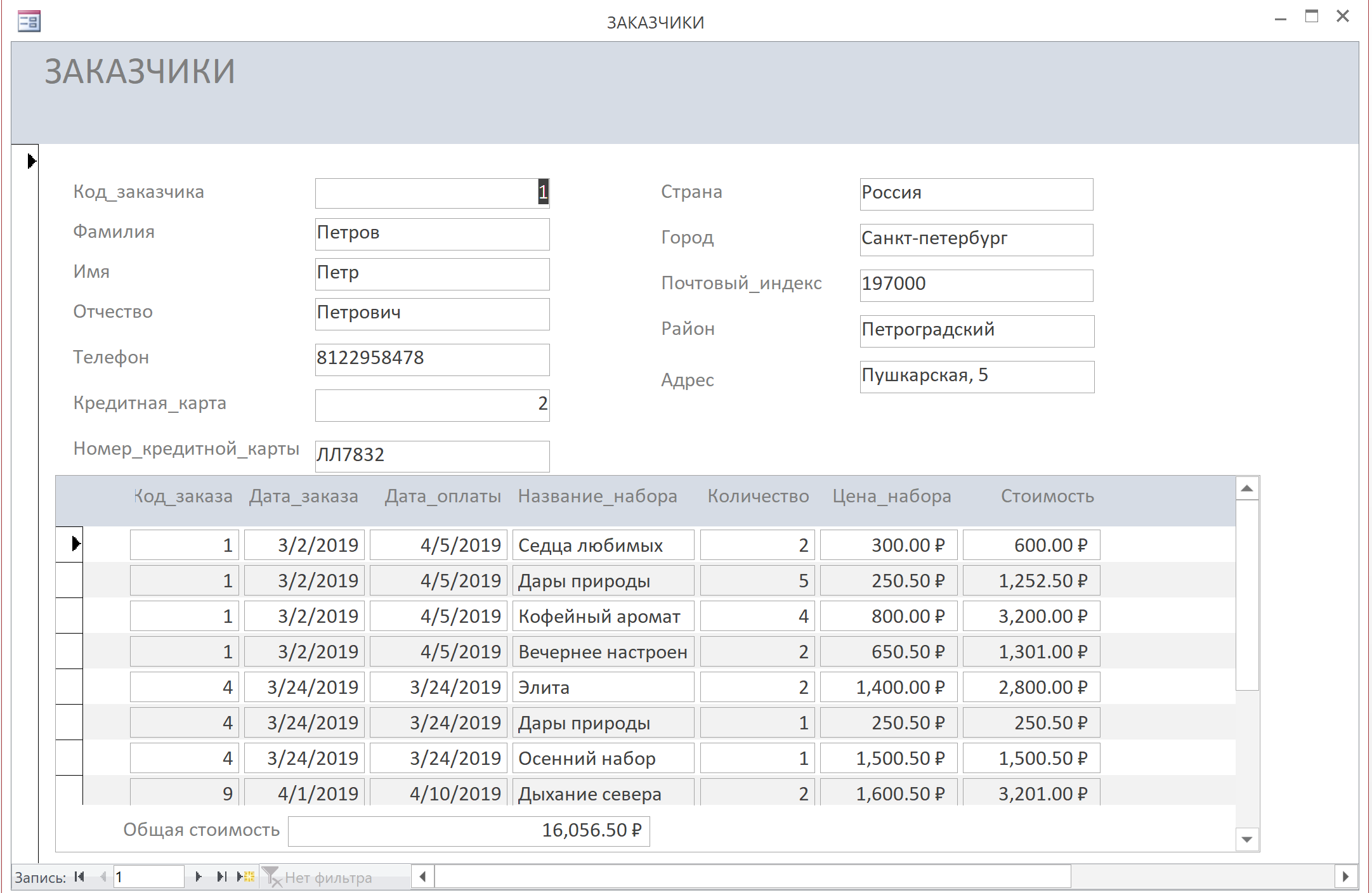


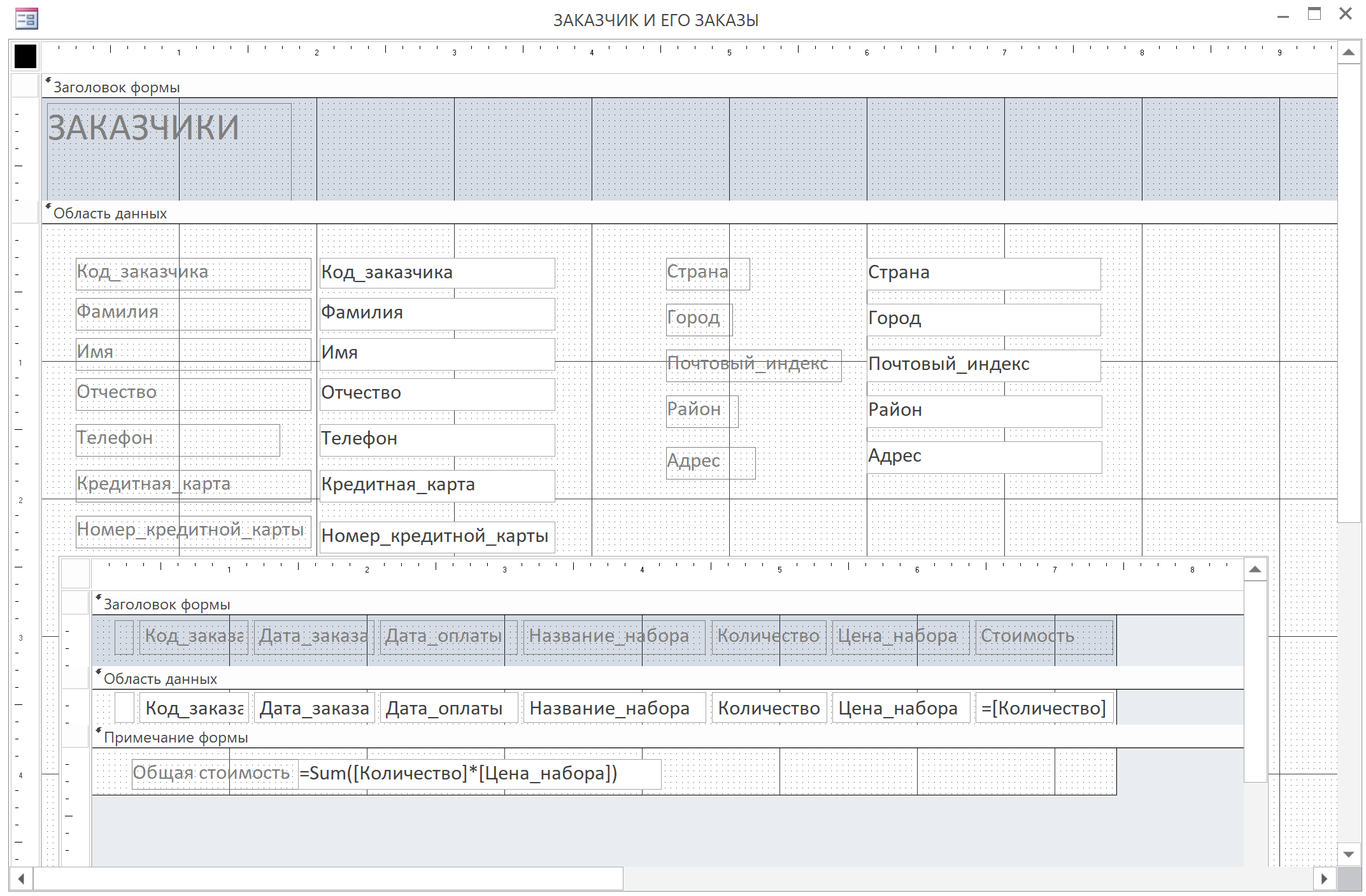
4) Заказы\_подчиненная



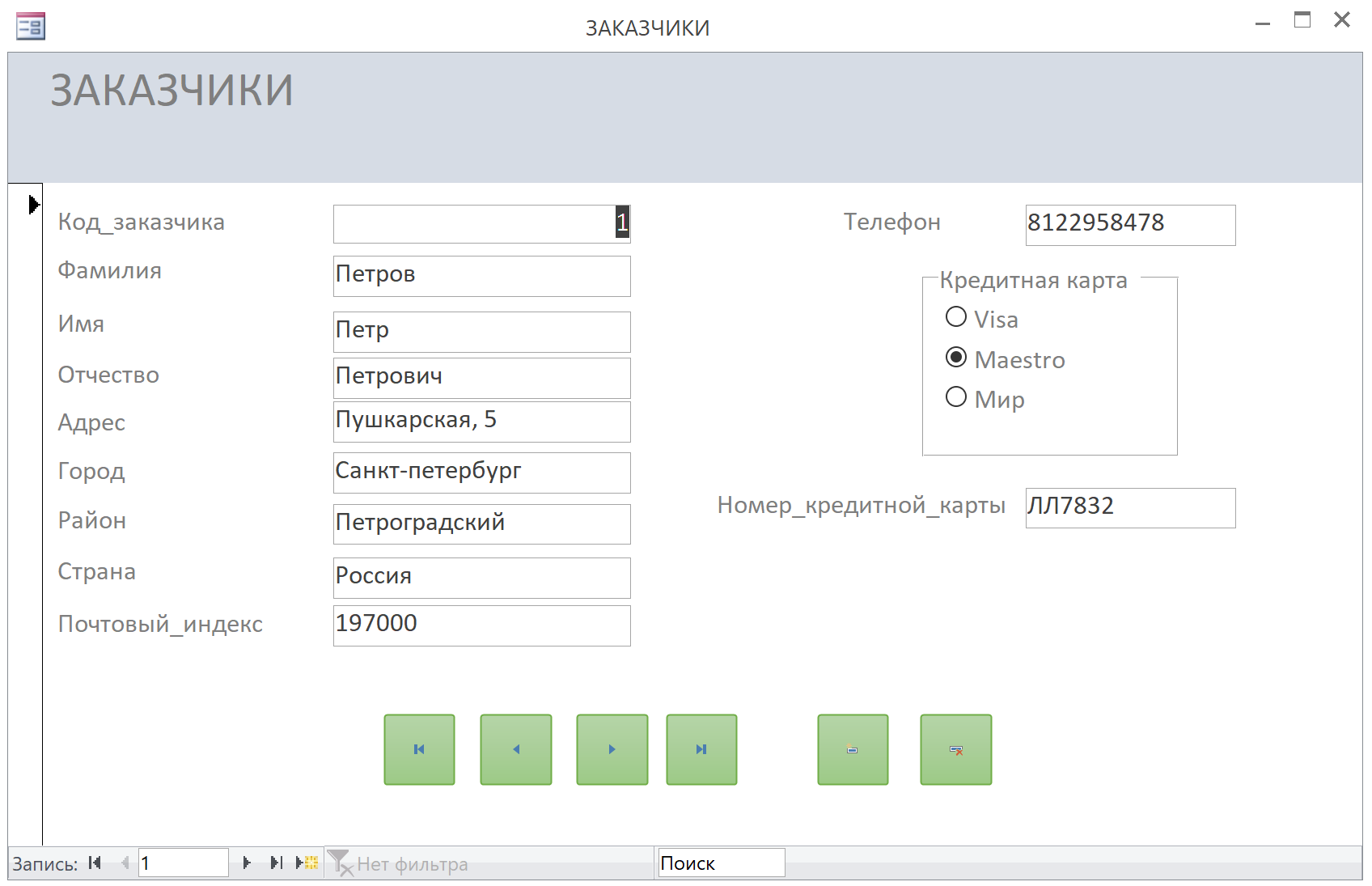


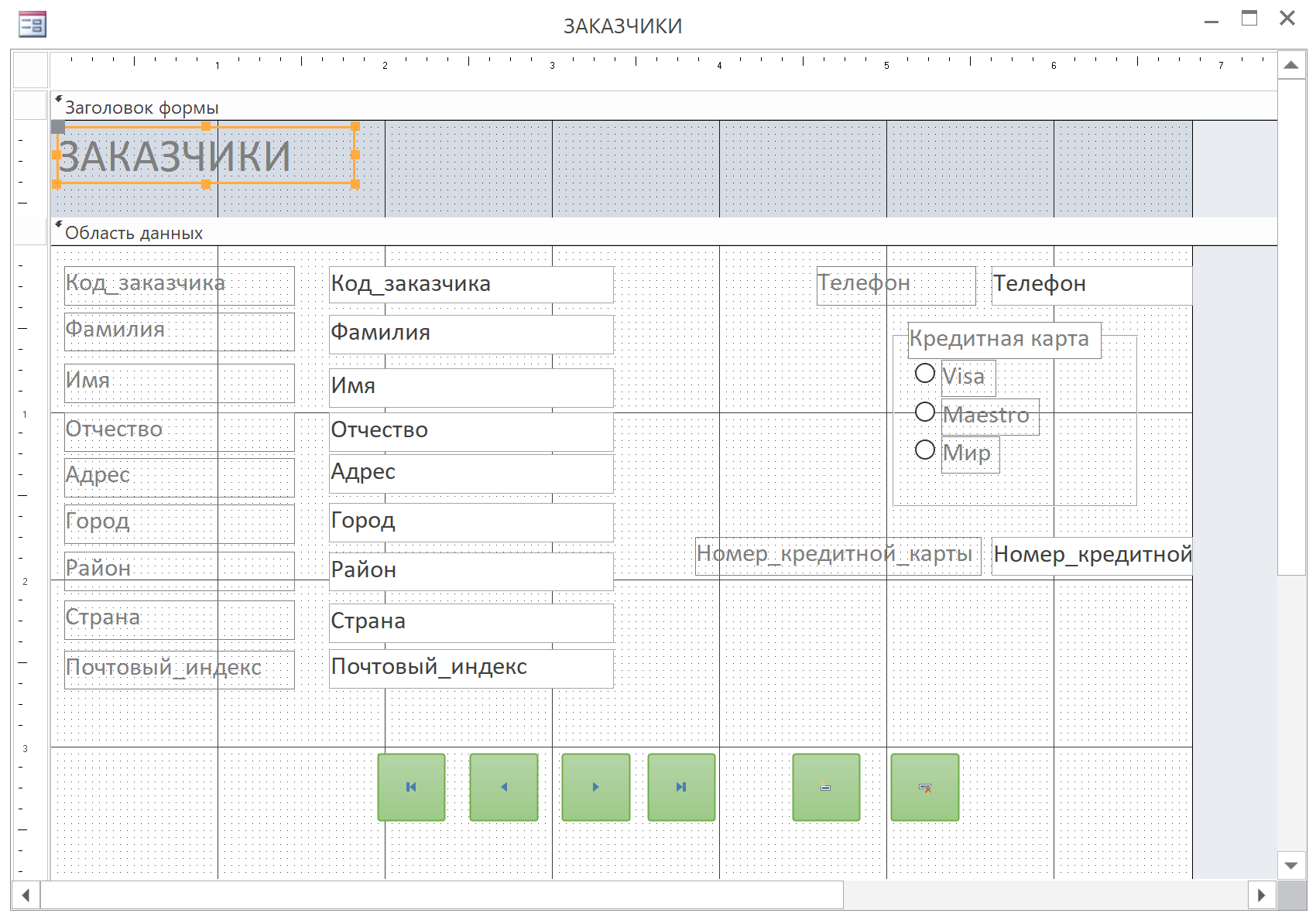
5) Заказчики и его заказы:





6) Заказчики





7) Начинки

