ПО БАЗИ ОТ ДАННИ

**КУРСОВА РАБОТА**

***Тема: Предоставяне на социални услуги „Домашен социален патронаж“***

**Изготвили:**

**Стоян Стоянов, спец. БИС, III-ти курс, група: 52, фак. №: 101060**

**/…………………………………./**

**Георги Георгиев, спец. БИС, III-ти курс, група: 52, фак. №: 100703**

**/…………………………………./**

**Борислав Божилов, спец. БИС, III-ти курс, група: 52, фак. №: 99884**

**/…………………………………./**

**Проверили:**

**Х. доц. д-р Мария Кашева**

**/…………………………………./**

**Гл. ас. д-р Иван Куюмджиев**

**/…………………………………./**

**Варна, 2016 г.**

й

Съдържание

[1. Описание на предметната област и бизнес правила 3](#_Toc452486079)

[1.1 Описание на предметната област 3](#_Toc452486080)

[1.2 Бизнес правила 4](#_Toc452486081)

[2. Начален E-R модел 5](#_Toc452486082)

[3. E-R модел (с добавени атрибути) 6](#_Toc452486083)

[4. Преобразуване на връзките и усъвършенстване на модела. Релационен модел на базата от данни 9](#_Toc452486084)

[5. SQL скриптове за създаване на базите от данни, дефиниране на таблиците и интегритета 14](#_Toc452486085)

[5.1 SQL скриптове за създаване на базите от данни 14](#_Toc452486086)

[5.2 SQL скриптове за дефиниране на таблиците 15](#_Toc452486087)

[5.3 SQL скриптове за добавяне на информация 17](#_Toc452486088)

[5.4 SQL скриптове за манипулиране на бази от данни 24](#_Toc452486089)

[6. Импортиране на данни от SQL база от данни в MS Excel и анализ с PowerPivot 31](#_Toc452486090)

[6.1 Импортиране на данни от SQL база от данни в MS Excel 31](#_Toc452486091)

[6.2 Анализ с PowerPivot 34](#_Toc452486092)

# 1. Описание на предметната област и бизнес правила

## 1.1 Описание на предметната област

Домашен Социален Патронаж (ДСП) представлява комплекс от социални услуги, предоставяни в общността – по домовете на възрастните хора и хора с увреждания. Площта на сградата на ДСП гр. Шабла е 386 квадратни метра.

Основната мисия на ДСП е да реализира обществения ангажимент към възрастните хора и хората с увреждания, като ги подпомага в тяхната естествена семейна среда чрез доставяне на храна – ежедневно се предлага топла храна, съобразена с изискванията за рационално хранене на хората от третата възраст. По тази причина, организацията изпълнява различни заявки за обслужване.

Всички записани клиенти притежават ЕГН, име, фамилия, град/село, адрес и години.

За всеки служител се съхранява информация за ЕГН, неговото име, фамилия, град, адрес, електронна поща, години и телефонен номер.

Всеки служител от своя страна е заел определена длъжност. Длъжността се характеризира с номер, име на длъжност и описание на длъжността..

За ястията са включени код на ястие, група ястие, наименование, цена на ястие и произведено количество от ястието.

За продукт имаме следните атрибути – код на продукт, име на продукт, грамаж и количество в склада и цена на единица продукт.

За доставчици – БУЛСТАТ, име на доставчик, град на доставчик, адрес на доставчик, телефонен номер, електронна поща.

За доставки – код на доставка, код на доставчик, дата на доставка, цена и дали е издадена фактура за доставката.

Създадените заявки за обслужване включват данни за номер на заявка за обслужване, ЕГН на служителя, ЕГН на клиент, код на поръчано ястие, дата на транспорт, дестинация, изминати километри и допълнително описание.

Трябва да се създаде база от данни, която да съдържа информация за следните същности:

* Клиенти
* Служители
* Ястия
* Продукти
* Доставчици
* Заявки за обслужване
* Длъжности
* Доставки

## 1.2 Бизнес правила

* Един клиент е възможно да направи повече от една заявка за обслужване, а една заявка за обслужване се отнася само до един определен клиент.
* Един служител може да обработва няколко заявки, докато една заявка се обработва само от един служител.
* Една заявка за обслужване може да включва повече от едно приготвено ястие, а ястията могат да бъдат включени в повече от една заявка.
* За приготвянето на едно ястие се използват много продукти, а един продукт може да се ползва за приготвяне на повече от едно ястие.
* За една длъжност може да работят повече от един човек (например няколко готвача), но един човек работи по точно определена длъжност.
* Един доставчик може да изпълни повече от една доставка. Една доставка се изпълнява от един доставчик.
* В една доставка се съдържат много продукти. Един продукт може да бъде доставен от множество доставки.

# 2. Начален E-R модел

**Легенда:**

* **Същност**
* **Връзка „Едно към много“ (One to Many)**
* **Връзка „Много към много“ (Many to Many)**

**1 – една същност**

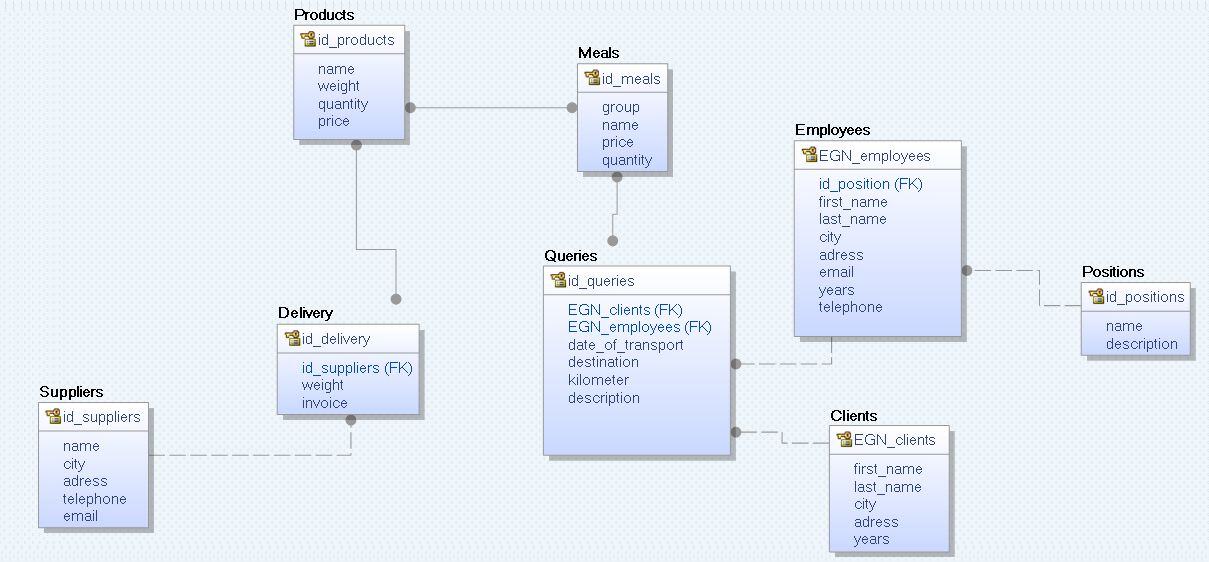
**M – много същности**

2.1 Кардиналност на връзките

**За свързването на отделните същности са използвани съответните връзки:**

* Връзката, която ще имаме е 1:M (едно към много) между „Клиенти“ и „Заявки за обслужване“.
* Между таблиците „Служители“ и „Заявки за обслужване“ имаме съответно връзка 1:M (едно към много).
* Връзка М:M (много към много) съществува при таблиците ,,Ястия“ и „Заявки за обслужване“.
* Между „Продукти“ и „Ястия“ съществува връзка M:M (много към много).
* Също така, връзка M:M (много към много) се характеризира и за същностите „Доставки“ и „Продукти“.
* Между таблиците „Доставчици“ и „Доставки“ имаме съответно връзка 1:M (едно към много).
* Между същностите „Длъжности“ и „Служители“ съответната връзка е 1:M (едно към много).

# 3. E-R модел (с добавени атрибути)

**Същности:**

* Suppliers (Доставчици) – Изпълняват доставянето на продуктите.
* Products (Продукти) – Продуктите, които се използват за приготвяне на ястия.
* Meals (Ястия) – Списък на ястията и групите ястия.
* Queries (Заявки) – Съдържат информация за клиент, служител, ястията и транспортирането на готовите ястия до клиентите.
* Clients (Клиенти) – Потребителите на ДСП. Информация за техните имена, град, адрес и години.
* Employees (Служители) – Хората работещи в звеното.
* Positions (Длъжности) – Заеманата длъжност от един работник.
* Delivery (Доставки) – Информация за доставките на продукти към ДСП.

**Идентифициране на същностите:**

* Suppliers (Доставчици) – id\_suppliers;
* Products (Продукти) – id\_products;
* Meals (Ястия) – id\_meals;
* Queries (Заявки) – id\_queries;
* Clients (Клиенти) – EGN\_clients;
* Employees (Служители) – EGN\_employees;
* Positions (Длъжности) – id\_positions;
* Delivery (Доставки) – id\_delivery;

**Атрибути, характеризиращи същностите:**

* **Suppliers (Доставчици):**
  + id\_suppliers [PK]
  + name – Име на доставчик
  + city – Град на доставчик
  + adress – Адрес на доставчика
  + telephone – Телефон за връзка
  + email – Електронна поща
* **Products (Продукти):**
  + id\_products [PK]
  + name – Име на продукт
  + products\_weight – Грамаж на дадения продукт в склада
  + quantity – Количество (в брой)
  + price – Цена за единица продукт
* **Meals (Ястия):**
  + id\_meals [PK]
  + group\_meals – Група ястия
  + name – Име на ястие
  + price – Цената на едно ястие
  + quantity – Количество на произведените ястия
* **Queries (Заявка):**
  + id\_queries [PK]
  + EGN\_employee [FK]
  + EGN\_clients [FK]
  + date\_of\_transport – Дата на транспортиране на заявката
  + destination – Място на транспортиране
  + kilometer – Изминати километри за един разнос
  + query\_description – Описание/ допълнителна информация
* **Clients (Клиенти):**
  + EGN\_clients [PK]
  + first\_name – Име на клиент
  + last\_name – Фамилия на клиент
  + city – Град на клиент
  + adress – Адрес на клиент
  + years – Години на клиент
* **Employees (Служители):**
  + EGN\_employees [PK]
  + id\_position [FK]
  + first\_name – Име на служител
  + last\_name – Фамилия на служител
  + city – Град на служител
  + adress – Адрес на служител
  + email – Електронна поща на служител
  + years – Години на служител
  + telephone – Телефон на служител
* **Positions (Длъжности):**
  + id\_positions [PK]
  + name – Наименование на длъжност
  + position\_description – Описание на длъжността
* **Delivery (Доставки):**
  + id\_delivery [PK]
  + id\_suppliers [FK]
  + date\_of\_delivery – Дата на доставката
  + price – Цена на една цяла доставка
  + invoice – Отговаря на това дали има издадена фактура за доставката

# 4. Преобразуване на връзките и усъвършенстване на модела. Релационен модел на базата от данни

4.1 Преобразуване на връзките и усъвършенстване на модела

Всички връзки в базата данни притежават свойството Enforce Referential  
Integrity, което осигурява цялостност, интегрираност и съвместимост на данните.  
По този начин потребителят не може да въведе данни, в свързани релации, които не си съответстват по тип и стойности. Така се осигурява съвместимостта между  
първични ключове и външни ключове.

При трансформиране на концептуалния E–R модел в релационен модел

могат да бъдат очертани следните стъпки и правила:

­ базата от данни, която е представена като E–R диаграма съдържаща

същности, може да бъде представена като съвкупност от таблици,

представляващи релационен модел на базата от данни;

­ атрибутите на типа същност се преобразуват директно в колони на

таблицата;

­ Чрез изрази се изразява директно подсхемата;

­ идентификаторът (първичният ключ) на типа същност става първичен ключ на таблицата;

­ установените правила за интегритет на данни се свързват с колоните на

таблиците;

­ за връзки от типа 1:1 или 1:М не се създават нови таблици. Те се представят чрез механизма на външните ключове. Ако връзката е от типа 1:1, първичния ключ на единия тип същност се включва като колона (външен ключ) в таблицата, съответстваща на другия тип същност. Ако връзката е от типа 1:М, първичния ключ на типа същност, към която връзката е 1, се добавя като колона (външен ключ) в таблицата, съответстваща на типа същност, към която връзката е М;

­ ако в E–R модела са останали връзки от типа М:М, то е необходимо да се

създаде свързваща релация, в която да се включват първичните ключове на релациите, съответстващи на двата типа същности.

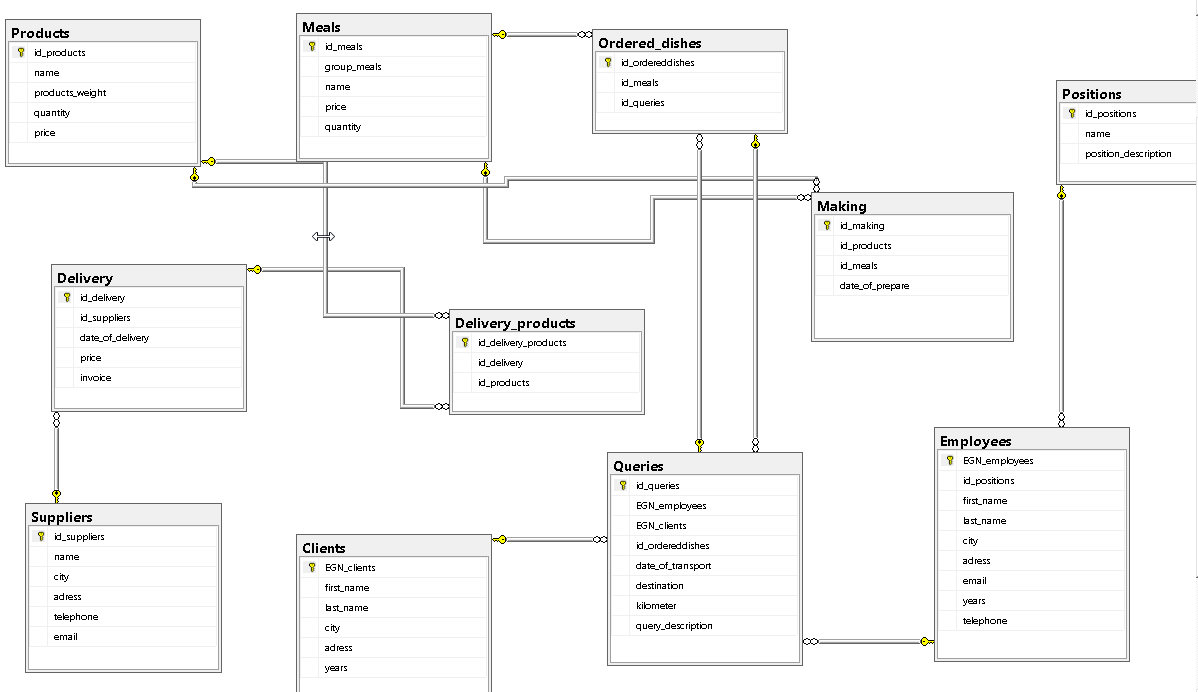
Във връзка с установените правила, за правилното функциониране на базата от данни е необходимо да бъдат разрешени връзките от тим M:M (много към много). В нашия случай това са връзките между двойките таблици:

* Delivery (Доставки) – Products (Продукти)
* Products (Продукти) – Meals (Ястия)
* Meals (Ястия) – Queries (Заявки)

Трябва да добавим три свързващи същности: Delivery\_Products (Доставки\_Продукти), Making (Приготвяне) и Ordered\_dishes (Поръчани ястия). Връзките М:М биват преобразувани в следния вид:

* „Products “ – „ Delivery\_products“ 1:M
* „Delivery\_products“ – „Delivery“ M:1
* „Products“ – „Making“ 1:M
* „Making“ – „Meals“ M:1
* „Meals“ – „Ordered\_dishes“ 1:M
* „Ordered\_dishes“ – „Queries“ M:1

4.2 Релационен модел на базата от данни



**Същности:**

* Suppliers (Доставчици) – Изпълняват доставянето на продуктите.
* Products (Продукти) – Продуктите, които се използват за приготвяне на ястия.
* Meals (Ястия) – Списък на ястията и групите ястия.
* Queries (Заявки) – Съдържат информация за клиент, служител, ястията и транспортирането на готовите ястия до клиентите.
* Clients (Клиенти) – Потребителите на ДСП. Информация за техните имена, град, адрес и години.
* Employees (Служители) – Хората работещи в звеното.
* Positions (Длъжности) – Заеманата длъжност от един работник.
* Delivery (Доставки) – Доставки към звеното.
* Delivery\_Products (Доставки\_Продукти) – Асоциираща таблица; Използва се за връзка между таблиците Delivery (Доставки) и Products (Продукти).
* Making (Приготвяне) – Асоциираща таблица; Използва се за връзка между таблиците Products (Продукти) и Meals (Ястия).
* Ordered\_dishes (Поръчани ястия) - Асоциираща таблица; Използва се за връзка между таблиците Meals (Ястия) и Queries (Заявки).

**Идентифициране на същностите:**

* Suppliers (Доставчици) – id\_suppliers;
* Products (Продукти) – id\_products;
* Meals (Ястия) – id\_meals;
* Queries (Заявки) – id\_queries;
* Clients (Клиенти) – EGN\_clients;
* Employees (Служители) – EGN\_employees;
* Positions (Длъжности) – id\_positions;
* Delivery\_products (Доставки\_Продукти) – id\_delivery\_products;
* Delivery (Доставка) – id\_delivery
* Making (Приготвяне) – id\_making
* Ordered\_dishes (Поръчани ястия) – id\_ordereddishes

**Атрибути, характеризиращи същностите:**

* **Suppliers (Доставчици):**
  + id\_suppliers [PK]
  + name – Име на доставчик
  + city – Град на доставчик
  + adress – Адрес на доставчика
  + telephone – Телефон за връзка
  + email – Електронна поща
* **Products (Продукти):**
  + id\_products [PK]
  + name – Име на продукт
  + products\_weight – Грамаж на даден продукт в склада
  + quantity – Налично количество в склада
  + price – Цена на единица продукт
* **Meals (Ястия):**
  + id\_meals [PK]
  + group\_meals – Група ястия
  + name – Име на ястие
  + price – Цената на едно ястие
  + quantity – Количество на произведените ястия
* **Queries (Заявка):**
  + id\_queries [PK]
  + EGN\_employee [FK]
  + EGN\_client [FK]
  + id\_ordereddishes [FK]
  + date\_of\_transport – Дата на транспортиране на заявката
  + destination – Място на транспортиране
  + kilometer – Изминати километри за един разнос
  + query\_description – Описание / допълнителна информация
* **Clients (Клиенти):**
  + EGN\_clients [PK]
  + first\_name – Име на клиент
  + last\_name – Фамилия на клиент
  + city – Град на клиент
  + years – Години на клиент
  + adress – Адрес на клиент
* **Employees (Служители):**
  + EGN\_employees [PK]
  + id\_position [FK]
  + first\_name – Име на служител
  + last\_name – Фамилия на служител
  + city – Град на служител
  + adress – Адрес на служител
  + email – Електронна поща на служител
  + years – Години на служител
  + telephone – Телефон на служител
* **Positions (Длъжности):**
  + id\_positions [PK]
  + name – Наименование на длъжност
  + position\_description – Описание на длъжността
* **Delivery (Доставка)**
  + id\_delivery [PK]
  + id\_suppliers [FK] – Външен ключ от таблицата Suppliers (Доставчици)
  + date\_of\_delivery – Дата на доставка
  + price – Цена на една цяла доставка
  + invoice – Определя дали е получена фактура за доставката
* **Delivery\_products (Доставки)**
  + id\_delivery\_products [PK]
  + id\_delivery [FK]
  + id\_products [FK]
* **Making (Приготвяне)**
  + id\_making [PK]
  + id\_products [FK]
  + id\_meals [FK]
  + date\_of\_prepare – Дата на приготвяне на ястие
* **Ordered\_dishes (Поръчани ястия)**
  + id\_ordereddishes [PK]
  + id\_meals [FK]
  + id\_queries [FK]

# 5. SQL скриптове за създаване на базите от данни, дефиниране на таблиците и интегритета

## 5.1 SQL скриптове за създаване на базите от данни

За създаване на базата от данни, използваме следния скрипт:

CREATE DATABASE db\_dsp\_2016 -- създаване на нова база

Резултат:

Command(s) completed successfully.

За използване на новосъздадената база:

GO

USE db\_dsp\_2016-- използване на новата база

Резултат:

Command(s) completed successfully.

## 5.2 SQL скриптове за дефиниране на таблиците

Създаване на таблицата Suppliers (Доставчици):

CREATE TABLE Suppliers(

id\_suppliers char(10) primary key, -- първичен ключ, БУЛСТАТ, 10 символа

name nvarchar(50), -- име на доставчик, 50 символа

city nvarchar(50), -- име на града, 50 символа

adress nvarchar(50), -- адрес на доставчик, 50 символа

telephone nvarchar(20), -- телефонен номер, 20 символа

email nvarchar(50) -- електронна поща, 50 символа

)

Създаване на таблицата Products (Продукти):

CREATE TABLE Products(

id\_products int primary key identity, -- първичен ключ, нараства автоматично с единица

name nvarchar(20), -- име на продукт

products\_weight decimal(6,3), -- тегло на един продукт съхранен в склада; (Пример: Колко е общото тегло на краставиците в склада ?)

quantity int, -- количество от дадения продукт, съхранявано в склада

price money -- цена на единица продукт; (Пример: Колко струва един домат ?)

)

Създаване на таблицата Delivery (Доставки):

CREATE TABLE Delivery(

id\_delivery int primary key identity, -- първичен ключ, номер на доставка, нараства автоматично с единица

id\_suppliers char(10) references Suppliers(id\_suppliers), /\*Задаваме id\_suppliers от таблица Suppliers за външен ключ\*/

date\_of\_delivery date, -- дата на доставка

price money, -- цена на цялата доставка

invoice nvarchar(5) – отговаря на това дали има получена фактура

)

Създаване на таблицата Delivery\_products (Доставки\_Продукти):

CREATE TABLE Delivery\_products(

id\_delivery\_products int primary key identity, -- първичен ключ, нараства автоматично с единица

id\_delivery int references Delivery(id\_delivery), /\*Задаваме id\_delivery от таблица Delivery за външен ключ\*/

id\_products int references Products(id\_products), /\*Задаваме id\_products от таблица Products за външен ключ\*/

)

Създаване на таблицата Meals (Ястия):

CREATE TABLE Meals(

id\_meals int primary key identity, -- Първичен ключ, номер на ястие от списъка с предлагани ястия

group\_meals nvarchar(20), -- Група ястие, до 20 символа

name nvarchar(30), -- Име ястие, до 30 символа

price money, -- цена на 1 бр. от дадено ястие

quantity int -- общ брой от произведеното ястие за деня

)

Създаване на таблицата Making (Приготвяне):

CREATE TABLE Making(

id\_making int primary key identity, -- пореден номер на приготвено ястие

id\_products int references Products(id\_products), /\*Задаваме id\_products от таблица

Products\_meals за външен ключ и е връзка към таблица Products\*/

id\_meals int references Meals(id\_meals), /\*Задаваме id\_meals от таблица

Products\_meals за външен ключ и е връзка към таблица Meals\*/

date\_of\_prepare date -- Дата на приготвяне на ястие

)

Създаване на таблицата Clients (Клиенти):

CREATE TABLE Clients(

EGN\_clients char(10) primary key, -- Първичен ключ

first\_name nvarchar(20), -- Име на клиент, до 20 символа

last\_name nvarchar(20), -- Фамилия на клиент, до 20 символа

city nvarchar(50), -- Град на клиент, до 50 символа

adress nvarchar(50), -- Адрес на клиент, до 50 символа

years tinyint -- Години на клиент

)

Създаване на таблицата Positions (Позиции):

CREATE TABLE Positions(

id\_positions int primary key identity, -- Пореден номер на длъжност, първичен ключ

name nvarchar(30), -- Име на длъжност

position\_description nvarchar(50) -- Описание на длъжност

)

Създаване на таблицата Employees (Служители):

CREATE TABLE Employees(

EGN\_employees char(10) primary key, -- Първичен ключ, показва ЕГН на служител

id\_positions int references Positions(id\_positions), /\*Задаваме id\_positions от таблица

Positions за външен ключ\*/

first\_name nvarchar(20), -- Име на служител, до 50 символа

last\_name nvarchar(20), -- Фамилия на служител, до 50 символа

city nvarchar(50), -- Град на служителя, до 50 символа

adress nvarchar(50), -- Адрес на служителя, до 50 символа

email nvarchar(50), -- Електронна поща на служител

years tinyint, -- Години на служител

telephone nvarchar(20) -- Телефон на служител

)

Създаване на таблицата Ordered\_dishes (Поръчани ястия):

CREATE TABLE Ordered\_dishes(

id\_ordereddishes int primary key identity, -- Уникален номер на поръчаните ястия

id\_meals int references Meals(id\_meals), /\*Задаваме id\_meals от таблица

Meals за външен ключ\*/

id\_queries int references Queries(id\_queries) /\*Задаваме id\_queries от таблица

Queries за външен ключ\*/

)

Създаване на таблицата Queries (Заявки):

CREATE TABLE Queries(

id\_queries int primary key identity, -- Първичен ключ, пореден номер на заявка

EGN\_employees char(10) references Employees(EGN\_employees), -- ЕГН на служител, взето от таблицата Employees

EGN\_clients char(10) references Clients(EGN\_clients), -- ЕГН на клиент, взето от таблицата Clients

id\_ordereddishes int,

date\_of\_transport date, -- Дата на изготвяне на заявка

destination nvarchar(30),

kilometer int,

query\_description nvarchar(50) -- Описание на заявка

)

Добавяне на ограничение FOREIGN KEY за таблицата Queries

ALTER TABLE Queries

ADD CONSTRAINT Ordered\_dishes\_FK FOREIGN KEY (id\_ordereddishes)

REFERENCES Ordered\_dishes (id\_ordereddishes);

## 5.3 SQL скриптове за добавяне на информация

-- INSERT INTO - Вмъкване на данни в таблица

-- VALUES - Стойности за полетата

Добавяне на данни в таблицата **Suppliers (Доставчици):**

INSERT INTO Suppliers (id\_suppliers, name, city, adress, telephone, email)

VALUES('5363420901', 'Лотус', 'Варна', 'Шипка 2', '0879203920', 'lotus@abv.bg'),

('5363423411', 'Еделвайс', 'Каварна', 'Нов живот 3', '0889234520', 'edelvais@abv.bg'),

('4346342341', 'Роял', 'Варна', 'Ангел Кънчев 10', '0887920901', 'royal1@gmail.com'),

('4326708921', 'Гергана - Деница', 'Каварна', 'Черни връх', '0898710794', 'gerganadenica@gmail.com'),

('4492038640', 'ЗлатенКлас', 'Балчик', 'Черно море 34', '0889019891', 'zlatenklas@yahoo.com'),

('4024900194', 'Био', 'Каварна', 'Братя Шкорпил 33', '0872019047', 'bio2004@abv.bg'),

('3792032610', 'Николай2003', 'Девня', '', '', 'niki2003@mail.bg'),

('3235832900', 'Царевец', 'Варна', 'Слънчев бряг 21', '0887204910', ''),

('3238992960', 'Асония', 'Балчик', 'Гео Милев 22', '0872008693', 'asonia@abv.bg'),

('8964907852', 'SeaFoods', 'Варна', 'Иван Вазов 29', '0892034820', 'seafoods@mail.bg'),

('8098643905', 'Yummy', 'Варна', 'Преслав 12', '0884039048', 'yummyfoods@gmail.com'),

('5930289028', 'HealthyFoods', 'Варна', 'Македония 73' , '0894890451', 'healthyfoods@gmail.com')

INSERT INTO Suppliers(id\_suppliers, name)

VALUES('1029408902', 'Paradise'),

('1945049056', 'Сладкарница Лондон'),

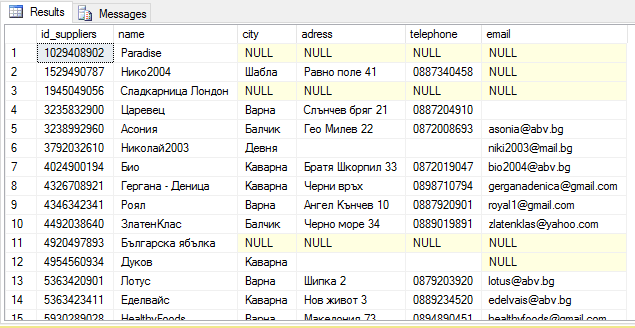
('4920497893', 'Българска ябълка')

INSERT INTO Suppliers(id\_suppliers, name, city, adress, telephone)

VALUES('1529490787', 'Нико2004', 'Шабла', 'Равно поле 41', '0887340458'),

('4954560934', 'Дуков', 'Каварна', '','')

Резултат:



Добавяне на данни в таблицата **Products (Продукти)**:

INSERT INTO Products(name,products\_weight, quantity, price)

VALUES('Кисело мляко', 0.400, 30, 0.85),

('Моркови', 1, 25, 0.55),

('Захар', 1, 25, 1.50),

('Леща', 1, 40, 2.50),

('Мандарини', 1, 20, 2.20),

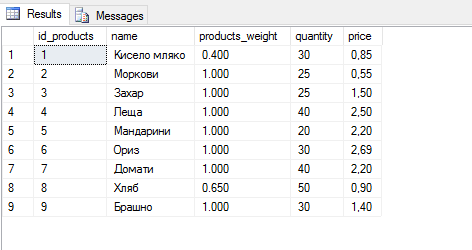
('Ориз', 1, 30, 2.69),

('Домати', 1, 40, 2.20),

('Хляб', 0.650, 50, 0.90),

('Брашно', 1, 30, 1.40)

Резултат:



Добавяне на данни в таблицата **Delivery (Доставки)**:

INSERT INTO Delivery(date\_of\_supply, price)

VALUES('12-04-2016', 300)

INSERT INTO Delivery(id\_suppliers,date\_of\_delivery, price, invoice)

VALUES ('5930289028','12-04-2016', 35, 'да'),

('3792032610','2016-04-15', 200, 'да'),

('4024900194','2016-04-18', 60, 'да'),

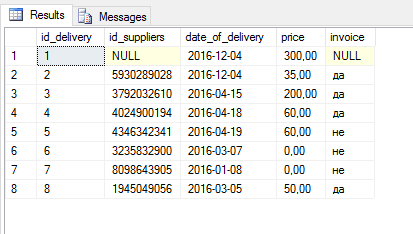
('4346342341','2016-04-19', 60, 'не'),

('3235832900','2016-03-07', '', 'не'),

('8098643905','2016-01-08', '', 'не'),

('1945049056','2016-03-05', 50, 'да')

Резултат:



Добавяне на данни в таблицата **Delivery\_products (Доставки\_Продукти)**:

INSERT INTO Delivery\_products(id\_delivery, id\_products)

VALUES(1, 2),

(3,1),

(2,1),

(2,4),

(4,5),

(3,1),

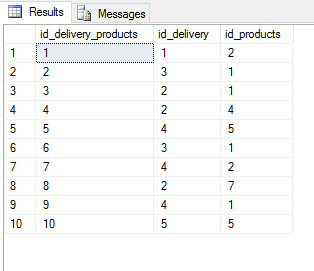
(4,2),

(2,7),

(4,1),

(5,5)

Резултат:



Добавяне на данни в таблицата **Meals (Ястия)**:

INSERT INTO Meals(group\_meals, name, price, quantity)

VALUES('Салата', 'Шопска', 3, 70 ),

('Супа', 'Пилешка', 2, 50 ),

('Супа', 'Леща', 2, 50),

('Супа', 'Месо', 3, 45),

('Супа', 'Топчета', 2.20, 40),

('Супа', 'Таратор', 1.50, 30),

('Десерт', 'Ашуре', 2, 30),

('Десерт', 'Бисквитена торта', 2.20, 39),

('Основно ястие', 'Руло Стефани', 2.60, 40),

('Основно ястие', 'Пиле с ориз', 2.20, 40),

('Основно ястие', 'Боб яхния', 2.40, 50),

('Основно ястие', 'Зеле с домати', 2.80, 40),

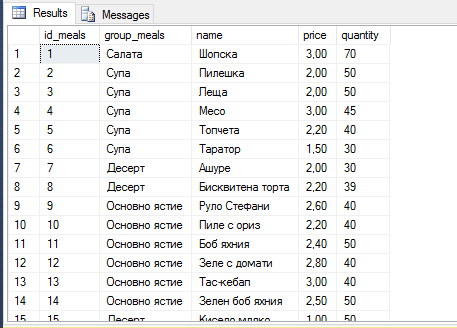
('Основно ястие', 'Тас-кебап', 3, 40),

('Основно ястие', 'Зелен боб яхния', 2.50, 50),

('Десерт', 'Кисело мляко', 1, 50),

('Десерт', 'Каварма', 0.60, 62)

Резултат:



Добавяне на данни в таблицата **Clients (Клиенти)**:

INSERT INTO Clients(EGN\_clients, first\_name, last\_name, city, adress, years)

VALUES('3402017983', 'Иван', 'Стоянов', 'Шабла', 'Марица 29', 82),

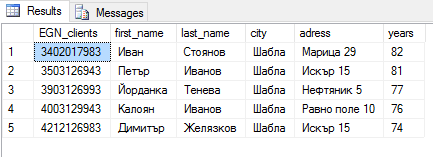
('3503126943', 'Петър', 'Иванов', 'Шабла', 'Искър 15', 81),

('4003129943', 'Калоян', 'Иванов', 'Шабла', 'Равно поле 10', 76),

('4212126983', 'Димитър', 'Желязков', 'Шабла', 'Искър 15', 74),

('3903126993', 'Йорданка', 'Тенева', 'Шабла', 'Нефтяник 5', 77)

Резултат:



Добавяне на данни в таблицата **Positions (Длъжности)**:

INSERT INTO Positions(name, position\_description)

VALUES ('Главен готвач', 'Отговаря за приготвянето на храната'),

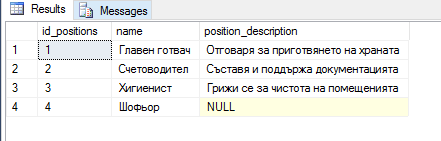
('Счетоводител', 'Съставя и поддържа документацията'),

('Хигиенист', 'Грижи се за чистота на помещенията')

INSERT INTO Positions(name)

VALUES ('Шофьор')

Резултат:



Добавяне на данни в таблицата **Employees (Служители)**:

INSERT INTO Employees(EGN\_employees, id\_positions, first\_name, last\_name, city, adress, email, years, telephone)

VALUES ('8203291293', 1, 'Надежда', 'Николова' , 'Шабла', 'Равно поле 45', 'nadejda82@abv.bg', 34, '0889203020'),

('7303190093', 2, 'Искра', 'Пенчева' , 'Шабла', 'Хан Кубрат 3', 'iskra.pencheva@mail.bg', 43, '0883204019'),

('8012039013', 3, 'Силвия', 'Христова' , 'Шабла', 'Хаджи Димитър 12', 'silviq.hristova@mail.bg', 35, '0887404281'),

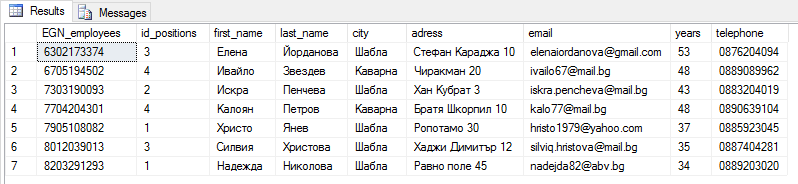
('6302173374', 3, 'Елена', 'Йорданова' , 'Шабла', 'Стефан Караджа 10', 'elenaiordanova@gmail.com', 53, '0876204094'),

('7905108082', 1, 'Христо', 'Янев', 'Шабла', 'Ропотамо 30', 'hristo1979@yahoo.com', 37, '0885923045'),

('6705194502', 4, 'Ивайло', 'Звездев', 'Каварна', 'Чиракман 20', 'ivailo67@mail.bg', 48, '0889089962'),

('7704204301', 4, 'Калоян', 'Петров', 'Каварна', 'Братя Шкорпил 10', 'kalo77@mail.bg', 48, '0890639104')

Резултат:



Добавяне на данни в таблицата **Making (Приготвяне)**:

INSERT INTO Making(id\_products, id\_meals, date\_of\_prepare)

VALUES (1, 3, '2016-03-12'),

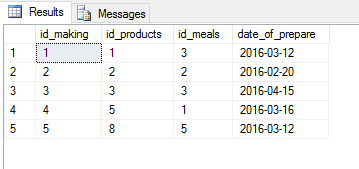
(2, 2, '2016-02-20'),

(3, 3, '2016-04-15'),

(5, 1, '2016-03-16'),

(8, 5, '2016-03-12')

Резултат:



Добавяне на данни в таблицата **Orders\_dishes (Поръчани ястия)**:

INSERT INTO Ordered\_dishes(id\_meals)

VALUES (1),

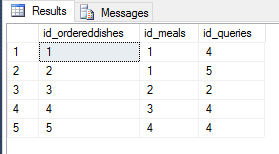
(1),

(2),

(3),

(4)

Резултат:



Добавяне на данни в таблицата **Queries (Заявки)**:

INSERT INTO Queries(EGN\_employees, EGN\_clients, id\_ordereddishes, date\_of\_transport, destination, kilometer, query\_description)

VALUES ('8203291293', '3503126943', 2, '2016-04-30', 'Шабла', 3, ''),

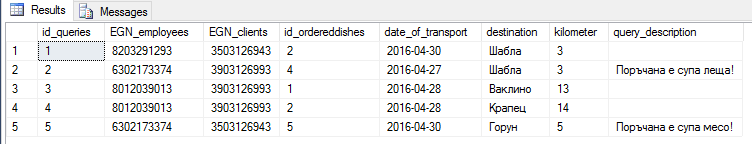
('6302173374', '3903126993', 4, '2016-04-27', 'Шабла', 3, 'Поръчана е супа леща!'),

('8012039013', '3903126993', 1, '2016-04-28', 'Ваклино', 13, ''),

('8012039013', '3903126993', 2, '2016-04-28', 'Крапец', 14, ''),

('6302173374', '3503126943', 5, '2016-04-30', 'Горун', 5, 'Поръчана е супа месо!')

Резултат:



## 5.4 SQL скриптове за манипулиране на бази от данни

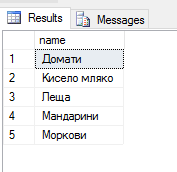
1. Покажете имената на продуктите за които има доставки:

SELECT DISTINCT name FROM Products -- Избираме уникалните имена от таблицата 'Продукти'

JOIN Delivery\_products -- Свързваме таблицата 'Продукти' и 'Доставки\_Продукти'

ON Products.id\_products = Delivery\_products.id\_products -- Свързваме таблиците посредством първичен/външен ключ

Резултат:

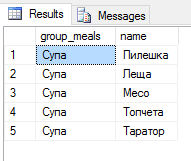


2. Изведете всички ястия от група „Супа“:

SELECT group\_meals, name FROM Meals -- Избираме групата ястие и името на ястието от таблицата 'Ястия'

WHERE group\_meals = 'супа' -- Където групата на ястието е 'супа'

Резултат:

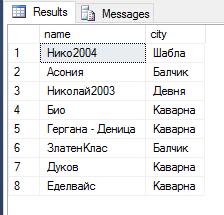


3. Изведете имената и градовете на всички доставчици които не са от Варна:

SELECT name, city FROM Suppliers -- Избираме име и град от таблицата 'Доставчици'

WHERE city != 'Варна' -- За всички доставчици от град Варна

Резултат:



4. Добавете ново ястие с име „Мусака“ от група „Основно ястие“:

INSERT INTO Meals(group\_meals, name, price, quantity) -- Добави група ястие, име на ястие, цена на ястие и приготвен брой ястия в таблицата 'Ястия'

VALUES ('Основно ястие', 'Мусака', 2.20, 80) -- Със съответните стойности

Резултат:



5. Изведете номер на доставката и име на доставчик за доставки които са извършени през миналия месец:

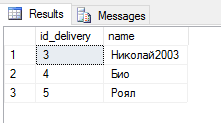
SELECT id\_delivery, Suppliers.name FROM Delivery -- Показваме номер на доставка и името на доставчик

JOIN Suppliers -- Свързваме таблиците 'Доставки' и 'Доставчици'

ON Delivery.id\_suppliers = Suppliers.id\_suppliers -- Използваме номер на доставчик като идентификатор за връзка

WHERE datediff(month,date\_of\_delivery,getdate())=1 -- Където датата на доставка е от предния месец

Резултат:



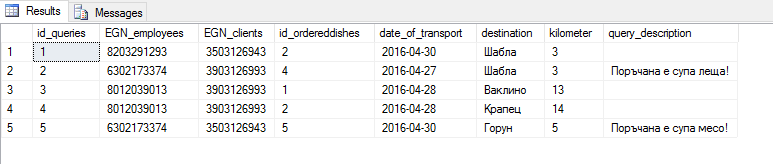
6. Поради допусната грешка, променете датата на транспортиране на храната, като при тези при които годината на доставяне на храна е 30 Април 2016 година, да се промени към дата 29 Април 2016 година.

UPDATE Queries -- Обновяваме таблицата 'Заявки за обслужване'

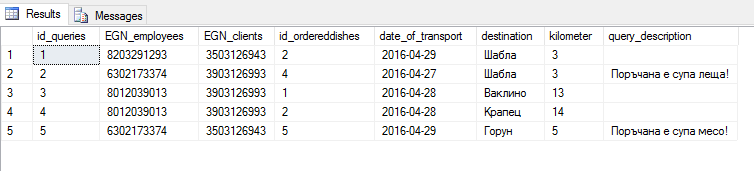
SET date\_of\_transport = '2016-04-29' -- Задаваме датата на транспорт да бъде 29 Април 2016 г.

WHERE date\_of\_transport = '2016-04-30' -- Където датата на транспорт е 30 Април 2016 г.

Резултат преди изпълнение на заявката:



Резултат след изпълнение на заявката:

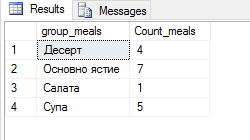


7. Покажете броя на ястията, които се предлагат по определена група ястия:

SELECT DISTINCT group\_meals, COUNT (id\_meals) as Count\_meals FROM Meals -- Показваме групата на ястията и брой различни ястия от съответната група от таблицата 'Ястия'

GROUP BY group\_meals -- Групираме по група ястия

Резултат:

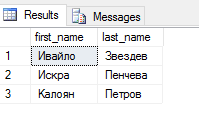


8. Изведете името и фамилията на служителите, които са между 40 и 50 години.

SELECT first\_name, last\_name FROM Employees -- Показване на името и фамилията на служителите от таблицата 'Служители'

WHERE years BETWEEN 40 AND 50 -- За които тяхната възраст е между 40 и 50 години.

Резултат:



9. Покажете името и фамилията на служителя, както и неговата длъжност, ако използва електронна поща в Mail.bg и заеманата длъжност да е хигиенист.

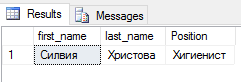
SELECT DISTINCT first\_name, last\_name, Positions.name AS Position FROM Employees -- Покажете името, фамилия и работната позиция на служителя от таблицата 'Служители'

JOIN Positions -- Свързваме таблиците 'Длъжности' и 'Служители'

ON Positions.id\_positions = Employees.id\_positions -- Чрез номера на служителя

WHERE email LIKE '%@mail.bg' AND Positions.name = 'Хигиенист' -- Където в колоната email се съдържа стойност @mail.bg и името на длъжността е Хигиенист

Резултат:



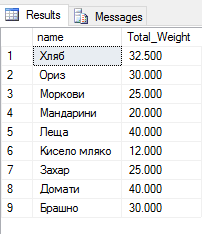
10. Покажете имената на продуктите и пресметнете общото тегло (products\_weight \* quantity) на всеки продукт от склада. Изчисленото тегло също да се показва в таблицата. Подредете записите в обратна посока на азбучния ред (от Я до А).

SELECT name, SUM (products\_weight \* quantity) as Total\_Weight FROM Products -- Показване на името на продукт, общото количество от продукт, съхранено в склада от таблицата 'Продукти'

GROUP BY id\_products, name -- Групираме по номер на продукт и име на продукт

ORDER BY name DESC -- Подреждаме записите по име обратно по азбучен ред

Резултат:



11. Изведете имената на ястията и техните групи, ако за тях не е въведена дата на приготвяне.

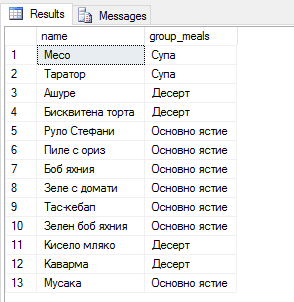
SELECT name, group\_meals FROM Meals -- Извеждане на име на ястие и неговата група от таблицата 'Ястия'

LEFT JOIN Making -- Вземат се всички редове от таблицата 'Приготвяне'

ON Meals.id\_meals = Making.id\_meals

WHERE Making.date\_of\_prepare IS NULL -- Където датата на приготвяне не е въведена

Резултат:

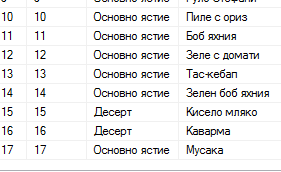


12. Изтрийте всички ястия, които съдържат в края на името си думата „яхния“

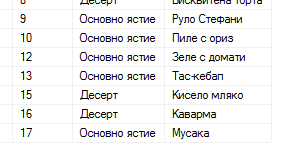
DELETE FROM Meals -- Изтриване на записи от таблицата 'Ястия'

WHERE name LIKE '%яхния' -- Където името на ястието завършва с думата 'яхния'

Резултат преди изпълнението на DELETE:



Резултат след изпълнението на DELETE:



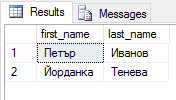
13. Покажете името и фамилията на клиентите, които имат заявка за обслужване.

SELECT DISTINCT first\_name, last\_name FROM Clients -- Показване на името и фамилията на клиентите

JOIN Queries -- Свързване на таблиците 'Клиенти' и 'Заявки за обслужване'

ON Clients.EGN\_clients = Queries.EGN\_clients -- Чрез ЕГН на клиента

Резултат:



14. Създайте нов изглед (View) и изведете данните след неговото изпълнение. Изгледа трябва да покаже имената и цените на ястията, които имат цени над средната цена за ястие.

GO

CREATE VIEW Meals\_above\_average\_price AS -- Създаване на изглед с име 'Meals\_above\_average\_price'

SELECT Meals.group\_meals, name, price -- Който да показва групата на ястието, името на ястието и неговата цена

FROM Meals -- От таблицата 'Ястия'

WHERE price > (SELECT AVG(price) FROM Meals) -- Където цената на ястието е над средната цена

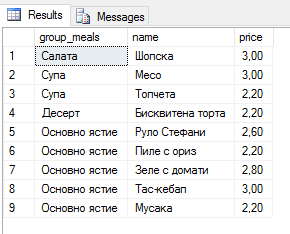
Резултат след създаване на VIEW:

Command(s) completed successfully.

Резултат от използване на създаденото VIEW:

GO

SELECT \* FROM Meals\_above\_average\_price -- Показване на резултата от създадения изглед

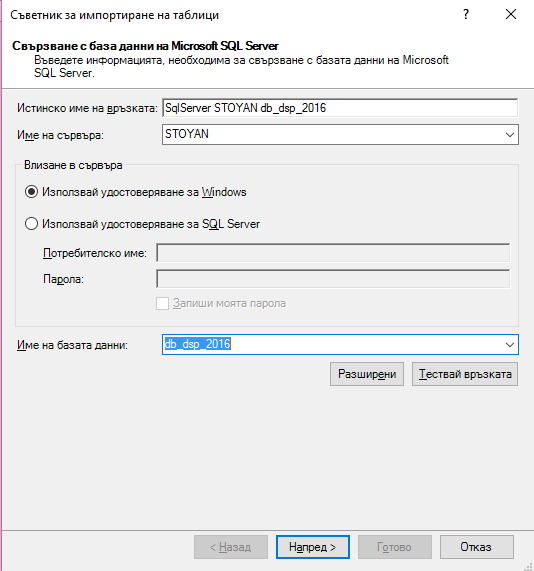


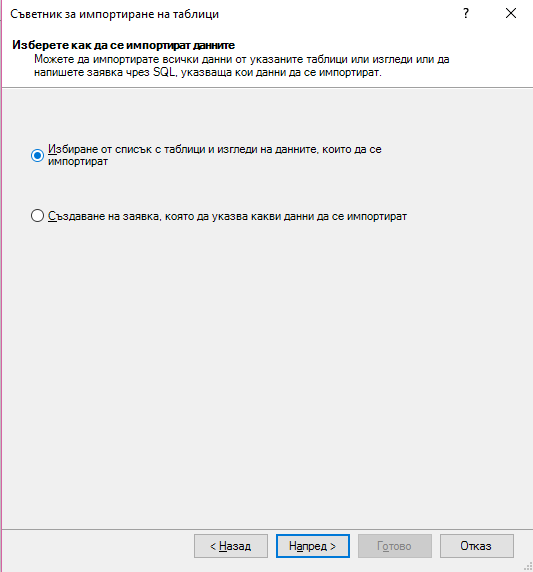
# 6. Импортиране на данни от SQL база от данни в MS Excel и анализ с PowerPivot

## 6.1 Импортиране на данни от SQL база от данни в MS Excel

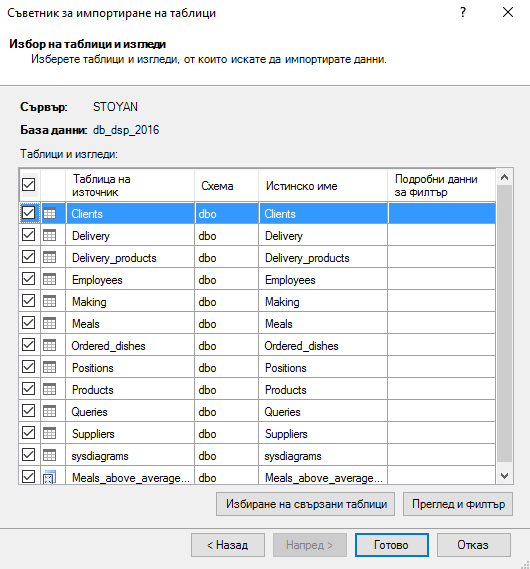
За извършването на този процес е използван продукта Microsoft Excel 2016 и инструмента PowerPivot for Excel. За внасянето на данните в MS Excel се използва следната процедура:

1. След като вече е активиран инструмента PowerPivot for Excel, появява се нов таб в лентата с инструменти на Ribbon интерфейса. След като изберем таб PowerPivot, избираме бутона Manage (Управление). След появата на прозореца се появява нов прозорец на PowerPivot. От там ще внесем данните от нашата база от данни като изберем опцията за получаване на външни данни.
2. Въвеждаме името на сървъра и името на базата от данни:

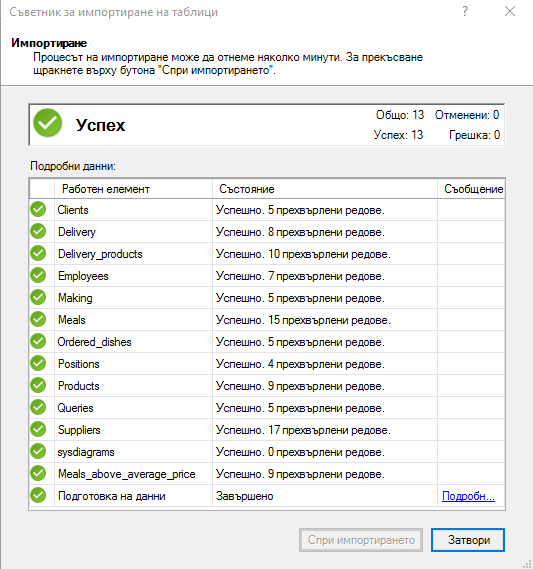
3. Избираме начина по който да се импортират данните. В случая ще изберем списъка от таблици и изгледи.



4. Посочваме всички възможни таблици и изгледи от нашата база от данни.



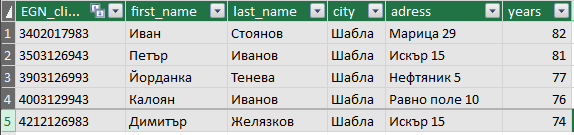
5. След успешно завършване на процеса по импортиране, получаваме прозорец със следния текст:



## 6.2 Анализ с PowerPivot

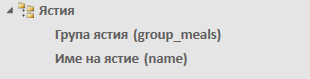
След като данните биват внесени в MS Excel чрез PowerPivot, ние можем да подготвим различни анализи и изчисления, които да представят данните в един по-обобщен план.

Ето как изглежда таблицата Clients (Клиенти) в Microsoft Excel:

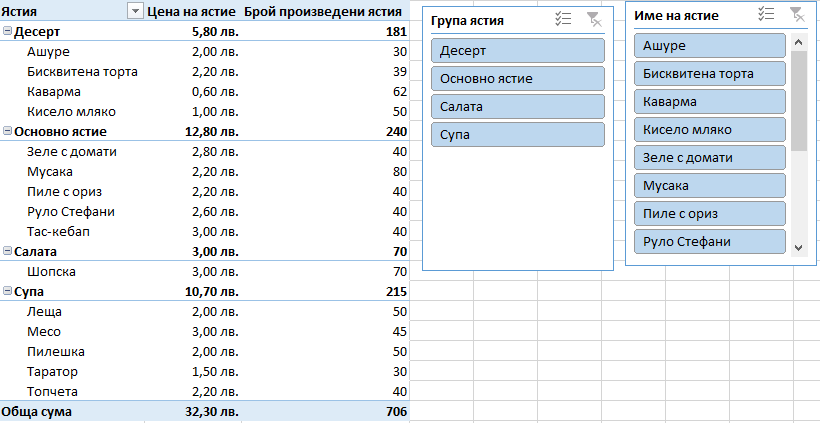


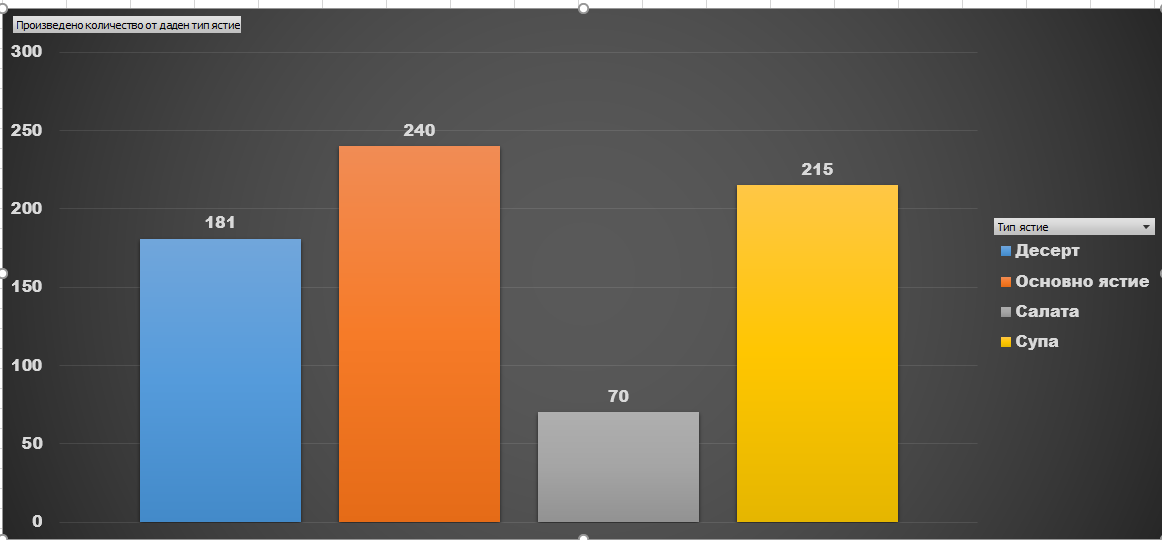
Чрез помощта на Pivot таблица и SQL Server ще покажем общата сума на произведените ястия, като за целта ще създадем изглед в базата от данни които ще бъде пренесен в PowerPivot.

В изглед на диаграма, чрез PowerPivot сме създали йерархия наречена „Ястия“, която да включва групите ястия и техните наименования.



Тази йерархия се използва за показване на групата и името на ястието. Като стойности се добавят цените и количеството на произведеното ястие. Чрез изчисление от Excel се пресмята общите цени по тип ястие, общите количества и обща сума за всяко ястие, по групи ястия и общо за всички ястия. Изображение от Pivot таблицата:



Microsoft Excel ни предоставя възможност да създадем и диаграми. Създадената диаграма показва общото количество за произведените ястия по типове. Можем да установим, че имаме най-много произведени ястия от тип „Основни ястия“ – цели 240.