

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. ОБЩИЕ СВЕДЕНИЯ О СУБД MS SQL SERVER. ОБЩИЕ СВЕДЕНИЯ О СУБД MONGO DB.	6
1.1. Системные и аппаратные требования MS SQL SERVER.....	6
1.2. Системные и аппаратные требования MONGODB	7
2. ПРОЕКТИРОВАНИЕ АНАЛИТИЧЕСКИХ ЗАПРОСОВ К OLTP БАЗЕ ДАННЫХ.....	8
2.1. Описание учебной базы данных AdventureWorks. Развертывание учебной базы данных на сервере MS SQL Server 2016.....	8
2.2. Создание и выполнение аналитических запросов к учебной базе данных AdventureWorks.....	11
2.3. Контрольные задания по теме	16
3. СТРУКТУРА ХРАНИЛИЩА ДАННЫХ (НА ОСНОВЕ ХРАНИЛИЩА ADVENTUREWORKS DW).	17
3.1. Описание учебной базы данных AdventureWorks2019DW. Развертывание учебной базы данных на сервере MS SQL Server 2019	17
3.2. Контрольные задания по теме	23
4. ПРОЕКТИРОВАНИЕ ХРАНИЛИЩА ДАННЫХ MS SQL SERVER	24
4.1. Проектирование физической модели хранилища данных типа «Звезда».....	24
4.2. Тестирование физической модели хранилища данных типа «Звезда».....	32
4.3. Контрольные задания по теме	35
5. ИМПОРТ И ЭКСПОРТ ДАННЫХ СРЕДСТВАМИ MS SQL SERVER	36
5.1. Экспорт и импорт данных с помощью Мастера экспорта и импорта данных MS SQL Server	36
5.2. Контрольные задания по теме	44
6. ПРОЕКТИРОВАНИЕ ПОТОКОВ УПРАВЛЕНИЯ SSIS ДЛЯ MS SQL SERVER (ETL-ПРОЦЕДУРЫ).....	45
6.1. Базовое описание инструментария SSIS для проектирования ETL- процедур.....	45
6.2. Реализация примера ETL-процедур типа «Поток управления-Поток данных»	48
6.3. Контрольные задания по теме	63

7. ИМПОРТ И ОБРАБОТКА ДАННЫХ В NOSQL ХРАНИЛИЩАХ	
ДАННЫХ НА ПРИМЕРЕ MONGODB	64
7.1. Работа с СУБД MongoDB в режиме подключения SSH. Создание базовых элементов документного хранилища данных	64
7.2. Экспорт данных в хранилище MongoDB.	66
7.3. Запросы к данным в хранилище MongoDB.....	69
Список литературы	74
Сведения об авторе	75

ВВЕДЕНИЕ

Основная цель реализации проектов современных хранилищ данных – обеспечить надежное хранение и передачу в специальные аналитические программы больших массивов данных. В зависимости от текущих потребностей предприятия используются как традиционные, реляционные СУБД, так и специально разработанные для работы с большими данными noSQL СУБД. В виду большого количества работ и участников, проект хранилища данных можно сравнить по размеру и сложности с проектом разработки полноценной информационной системы. Не стоит забывать о необходимости постоянной модернизации и оптимизации хранилища уже после его релиза.

Данный практикум содержит 7 тем, связанных с разными этапами проектирования и эксплуатации реляционных и постреляционных хранилищ данных. Читателю последовательно предлагается практические материалы для отработки навыков развертывания и настройки программного обеспечения, изучения структуры готовых учебных образцов хранилищ данных и создания собственных рабочих моделей хранилищ данных. В качестве реляционной СУБД предлагается популярное программное обеспечение MS SQL Server, в качестве noSQL СУБД – MongoDB.

Каждая тема заканчивается несколькими заданиями для самостоятельной работы, позволяющими читателю закрепить полученные в процессе выполнения предложенных примеров практические навыки.

1. ОБЩИЕ СВЕДЕНИЯ О СУБД MS SQL SERVER. ОБЩИЕ СВЕДЕНИЯ О СУБД MONGO DB

1.1. Системные и аппаратные требования MS SQL SERVER

Для создания среды, в которой студентом будут выполняться поставленные в методических указаниях задания, потребуется установка и настройка ряда программных продуктов. Для изучения традиционных OLTP и OLAP хранилищ данных будет использовано программное обеспечение Microsoft SQL Server 2019 (далее - MS SQL Server) и программная оболочка менеджмента сервера баз данных Microsoft SQL Server Management Studio.

В таблице ниже приводятся системные и аппаратные требования для установки и эксплуатации ПО SQL Server (табл. 1):

Таблица 1. Требования к экземпляру MS SQL Server

Компонент	Требование
Жесткий диск	6 Гб
ОЗУ	1 Гб (рекомендуется 4 Гб)
Процессор	x64 1,4 ГГц (Рекомендуется 2,0 ГГц)
Тип процессора	AMD Opteron, AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T, Intel Pentium IV с поддержкой EM64T.
Монитор	Super VGA с разрешением 800x600 пикселей или более высоким
Интернет	При необходимости поддержки интернет-средств MS SQL Server

Помимо указанных выше требований, для функционирования СУБД требуется предварительно установить или проверить наличие на ПК следующего программного обеспечения (табл. 2):

Таблица 2. Требования к системному программному обеспечению экземпляра MS SQL Server

Тип	Требование
Операционная система	Windows 10 TH1 1507 или более поздней версии Windows Server 2016 или более поздней версии
.NET Framework	Версия зависит от операционной системы
Сетевое программное обеспечение	Поддерживаемые операционные системы для SQL Server содержат встроенное сетевое программное обеспечение. Именованные экземпляры и экземпляры по умолчанию

	изолированной установки поддерживают следующие сетевые протоколы: Shared memory, Named Pipes и TCP/IP.
--	--

Более подробно функционал MS SQL Server и графический интерфейс среды управления будет рассмотрен в ходе описания соответствующих практикумов.

1.2. Системные и аппаратные требования MONGODB

Для изучения noSQL хранилищ данных будет использовано программное обеспечение MongoDB актуальной на момент прочтения издания версии (далее - MongoDB) и программная оболочка менеджмента сервера баз данных MongoDB Compass.

В табл. 3 приведены системные и аппаратные требования к экземпляру СУБД mongoDB:

Таблица 3. Требования к экземпляру MongoDB

Компонент	Требование
Жесткий диск	Минимально - 10 Гб
Процессор и память	x64 как минимум два ядра и 2 Гб памяти для развертывания основного сервиса. Еще столько же для резервной копии
Тип процессора	AMD Opteron, AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T
Монитор	Super VGA с разрешением 800x600 пикселей или более высоким

В табл. 4 приведено краткое описание требований к системному программному обеспечению:

Таблица 4. Требования к системному программному обеспечению экземпляра MS SQL Server

Тип	Требование
Операционная система	Windows 10 TH1 1507 или более поздней версии Windows Server 2016 или более поздней версии Сервер на базе ОС Linux, предпочтительно дистрибутив Ubuntu

Более подробно функционал MongoDB и графический интерфейс среды управления будет рассмотрен в ходе описания соответствующих практикумов.

2. ПРОЕКТИРОВАНИЕ АНАЛИТИЧЕСКИХ ЗАПРОСОВ К OLTP БАЗЕ ДАННЫХ

2.1. Описание учебной базы данных AdventureWorks. Развертывание учебной базы данных на сервере MS SQL Server 2016

Учебная база данных AdventureWorks создана и распространяется компанией Microsoft для проведения практического обучения навыкам работы с программным обеспечением баз данных MS SQL Server. Это сравнительно большая по объему база данных производственного предприятия, включающая в себя более 700 различных объектов и значимые тестовые массивы данных. Для работы с этой учебной базой данных требуется предварительная установка и настройка локального экземпляра базы данных MS SQL Server, среды управления MS SQL Server Management Studio, а также непосредственно самой тестовой базы данных. Процесс установки и настройки программного обеспечения базы данных подробно описан в видеоролике по следующей ссылке: <https://www.youtube.com/watch?v=h6BGVRy68UY>

После установки следует убедиться в том, что экземпляр сервера запущен, после чего следует запустить среду управления MS SQL Server Management Studio и, следуя приведенной ниже инструкции, установить учебную базу данных.

Перейдя по следующей ссылке: <https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms> из таблицы с образцами следует скачать файл AdventureWorks2016.bak. Это архив учебной базы данных, которые будет установлен на предварительно развернутом локальном сервисе.

Далее, необходимо запустить MS SQL Server Management Studio и, указав в окне приглашения (рис. 1) адрес локального сервера (в случае, если сервис базы данных запущен, как правило, этот адрес проставляется автоматически), соединиться с ним.

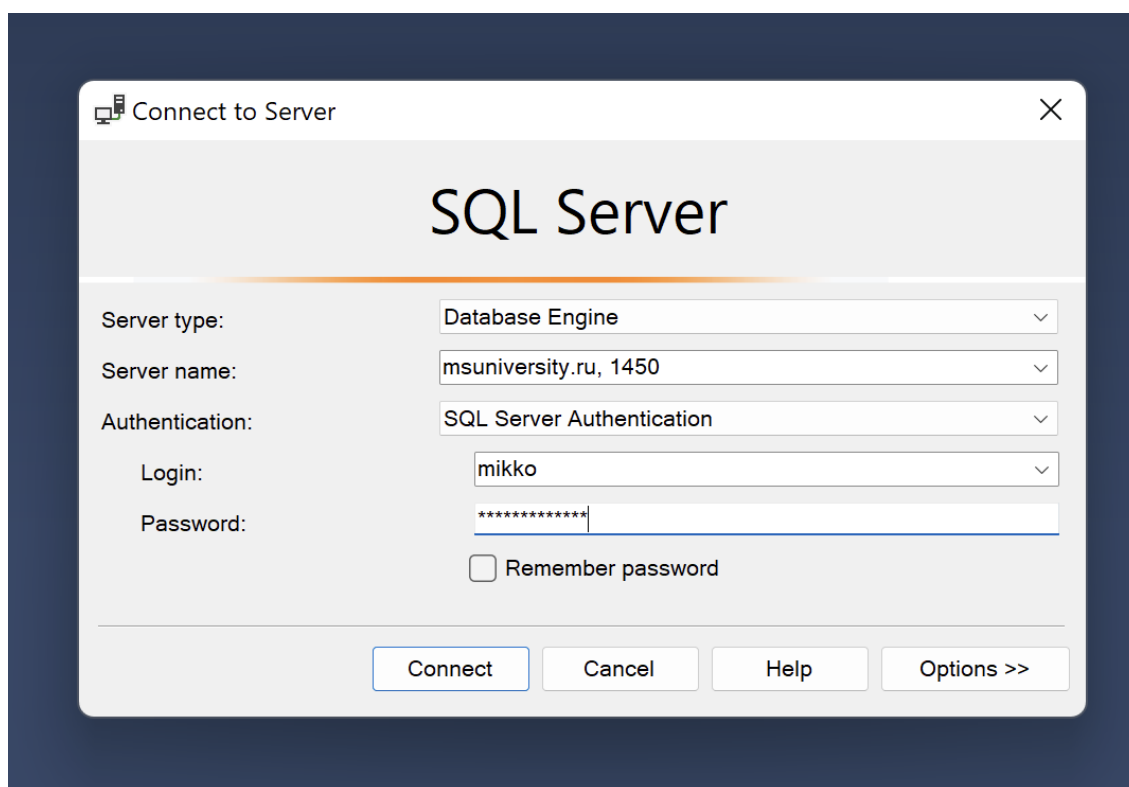


Рисунок 1. Подключение к экземпляру MS SQL Server

После успешного подключения, в Обозревателе объектов (слева), следует нажать правой клавишей мыши на папке Базы данных (Databases) и выбрать пункт «Восстановить базу данных», рис. 2.

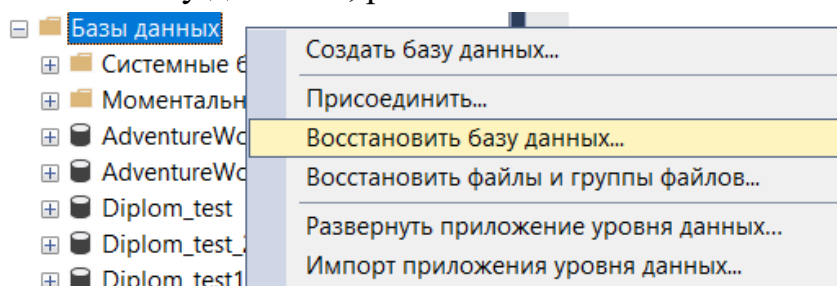


Рисунок 2. Запуск мастера восстановления базы данных

В появившемся окне мастера восстановления (рис. 3), в качестве Источника следует выбрать Устройство (1), после чего нажать на кнопку справа (2). В появившейся форме, нажав на кнопку Добавить (3) указать путь к физическому файлу учебной базы данных, скачанному на предыдущем этапе.

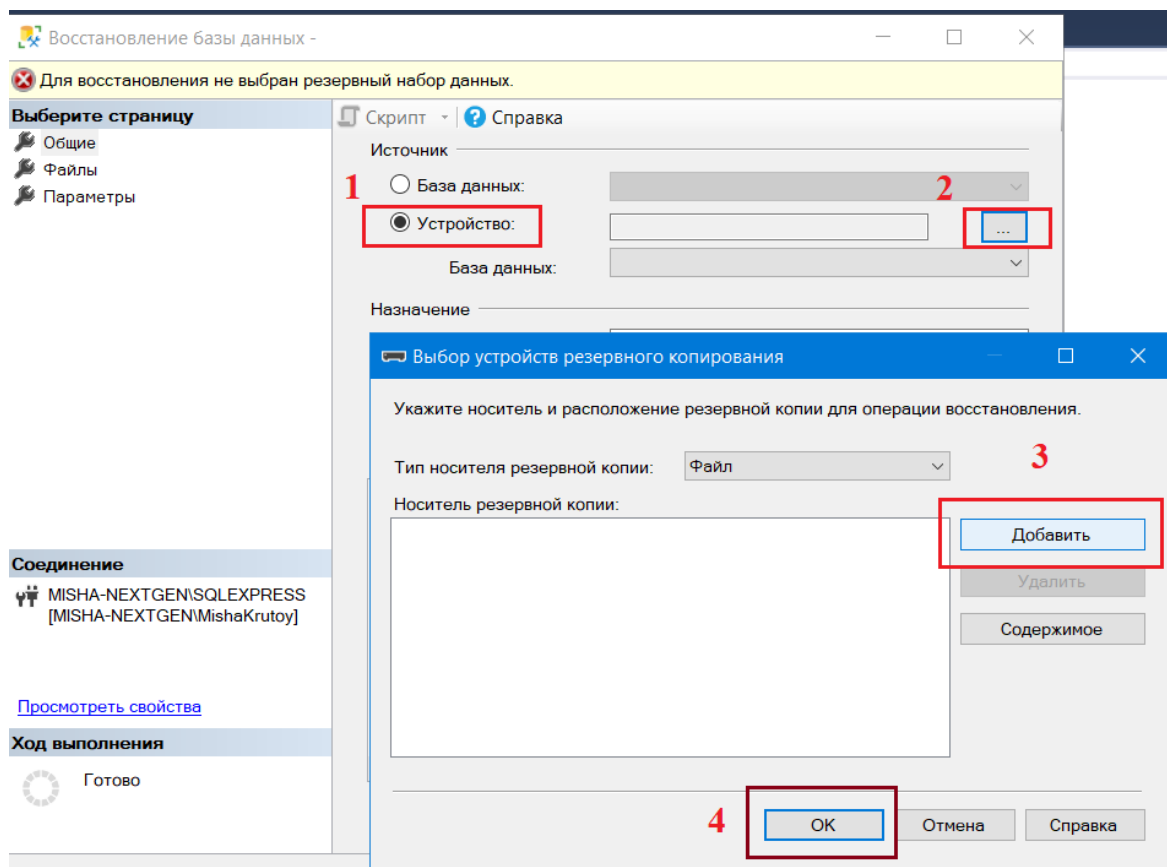


Рисунок 3. Выбор файла для восстановления БД

Далее следует визуально проверить в соответствующем окне путь и название базы данных (рис. 4) и запустить процесс восстановления кнопкой Ок.

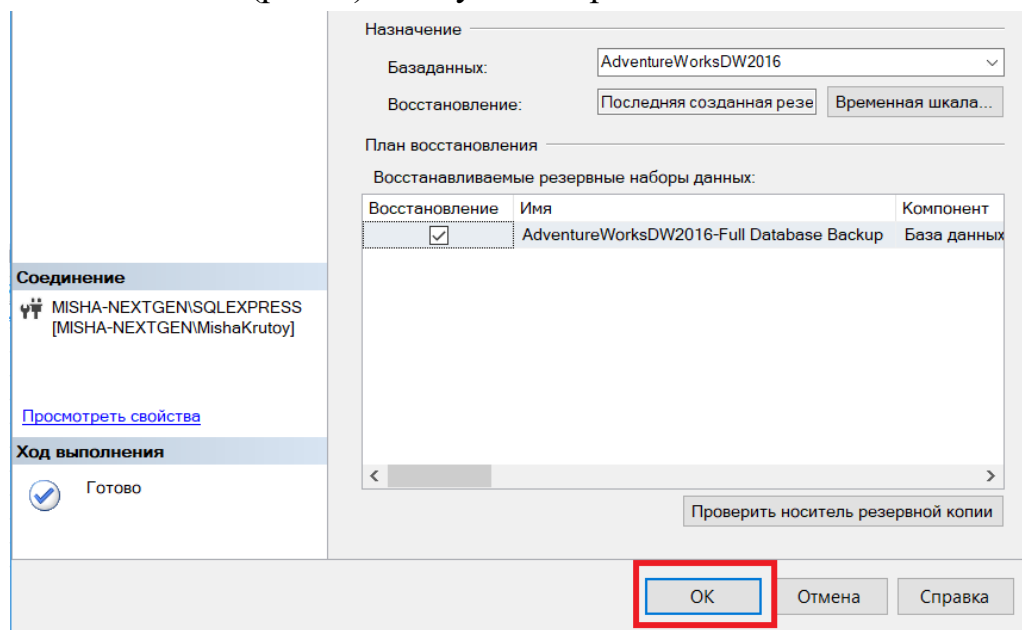


Рисунок 4. Подтверждение расположения восстановления

В случае успешного выполнения описанных выше процедур, учебная база данных AdventureWorks2016 появится в раскрытом списке Базы данных окна Обзор объектов (рис. 5)

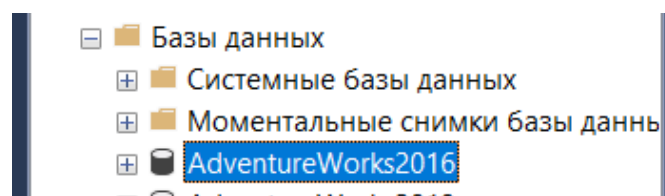


Рисунок 5. Проверка наличия БД в обозревателе объектов

После успешной установки тестовой базы данных, следует ознакомиться с ее физической моделью в документе, представленном по ссылке: https://msuniversity.ru/uploads/msu_file/file/30311/adventureworks.zip на предмет представленных в базе данных таблиц и связей между ними. Эта информация потребуется для выполнения второй части практического задания.

2.2. Создание и выполнение аналитических запросов к учебной базе данных AdventureWorks

Структура хранилища данных организована таким образом, чтобы оптимизировать скорость и качество выполнения аналитических запросов. Особенность аналитических запросов к данным заключается в том, что основной массив данных подлежащих выгрузке и обработке – это числовые данные. Символьные данные, как правило, играют роль меток, обозначающих оси графиков или столбцы табличной отчетности. Также, как правило, в ходе аналитических исследований, специалисты агрегируют массивы данных из нескольких (подчас десятков) разных таблиц. В этих обстоятельствах важную роль играет практика создания и исполнения скриптов аналитических запросов в среде реляционной модели хранения.

Далее, на нескольких запросах к данным разберем базовые принципы агрегации данных нескольких таблиц, а также результатную структуру аналитических отчетов. Для начала работы потребуется экземпляр запущенный сервер баз данных MS SQL Server с тестовой базой данных AdventureWorks, а также запущенная среда MS SQL Server Management Studio, подключенная к экземпляру сервера (см. п. 2.1).

Для написания скрипта запроса в языке SQL используется инструмент Новый запрос (New Query). Следует в Обозревателе объектов левой клавишей мыши выбрать базу данных, к которой будет составлен запрос, после чего нажать кнопку Новый запрос (New Query) в инструментарии Management Studio (рис. 6).

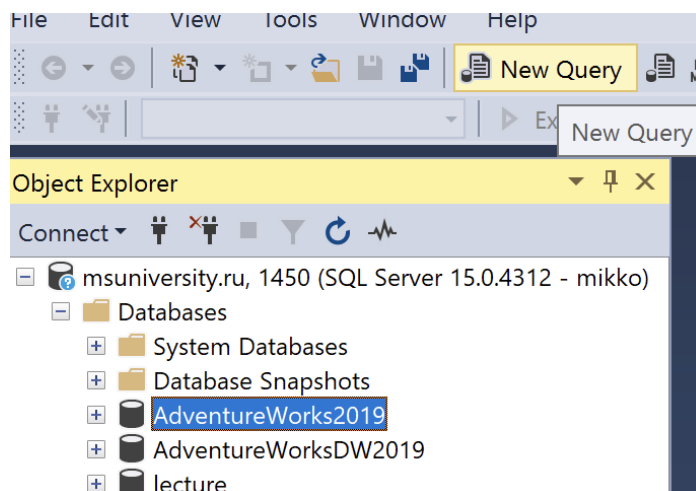


Рисунок 6. Создание нового запроса

Запрос SQL пишется в открывшемся для этого специальном окне, после чего запускается на исполнение клавишей F5 или кнопкой Выполнить (Execute, рис. 7).

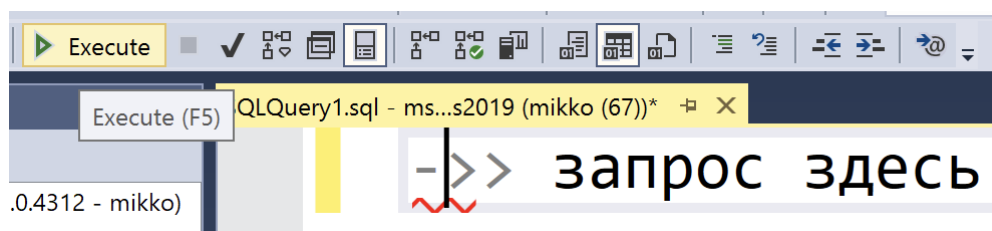


Рисунок 7. Запуск запроса на исполнение

Поставим первую задачу для проведения аналитического исследования. Компания AdventureWorks осуществляет дистрибьюцию своей продукции (велосипеды, запчасти к ним и экипировка) по всему миру. Список доступных к исследованию регионов и стран мира, где осуществляются продажи приведен в таблице SalesTerritory в домене Sales. Данная таблица состоит из 9 столбцов и, помимо названий территорий, приводит данные о суммах продаж по регионам в текущем (YTD) и прошлом (LastYear) году (рис. 8).

SQLQuery1.sql - ms...s2019 (mikko (67))*										
SELECT * FROM Sales.SalesTerritory;										
Results Messages										
TerritoryID	Name	CountryRegionCode	Group	SalesYTD	SalesLastYear	CostYTD	CostLastYear	rowguid	ModifiedDate	
1	Northwest	US	North America	7887186,7882	3298694,4938	0,00	0,00	43689A10-E30B-497F-B0DE-11DE20267FF7	2008-04-30 00:00:00.000	
2	Northeast	US	North America	2402176,8476	3607148,9371	0,00	0,00	00FB7309-96CC-49E2-8363-0A1BA72486F2	2008-04-30 00:00:00.000	
3	Central	US	North America	3072175,118	3205014,0767	0,00	0,00	DF6E7FD8-1A8D-468C-B103-ED8ADB8452C1	2008-04-30 00:00:00.000	
4	Southwest	US	North America	10510853,8739	5366575,7098	0,00	0,00	DC3E9EAD-7950-4431-9428-99DBCB33865	2008-04-30 00:00:00.000	
5	Southeast	US	North America	2538667,2515	3925071,4318	0,00	0,00	6DC4165A-5E4C-42D2-809D-4344E0AC75E7	2008-04-30 00:00:00.000	
6	Canada	CA	North America	6771829,1376	5693988,86	0,00	0,00	06B4AF8A-1639-476E-9266-110461D66B00	2008-04-30 00:00:00.000	
7	France	FR	Europe	4772398,3078	2396539,7601	0,00	0,00	BF806804-9B4C-4B07-9D19-706F2E689552	2008-04-30 00:00:00.000	
8	Germany	DE	Europe	3805202,3478	1307949,7917	0,00	0,00	6D2450DB-8159-414F-A917-E73EE91C38A9	2008-04-30 00:00:00.000	
9	Australia	AU	Pacific	5977814,9154	2278548,9776	0,00	0,00	602E612E-0FE9-41D9-B894-27E489747885	2008-04-30 00:00:00.000	
10	United Kingdom	GB	Europe	5012905,3656	1635823,3967	0,00	0,00	05FC7E1F-2DEA-414E-9ECD-09D150516FB5	2008-04-30 00:00:00.000	

Рисунок 8. Данные таблицы SalesTerritory

Также, в базе данных представлена таблица `SalesOrderHeader`, в которой содержатся детализированные данные о выполненных представительствами `AdventureWorks` заказах. Таблица очень «широкая» (содержит большое количество столбцов) и с ее содержимым предлагается ознакомиться самостоятельно. Учитывая наличие связи между таблицами `SalesTerritory` и `SalesOrderHeader` в одном домене, исследователь данных решил узнать итоговые показатели продаж (`TotalDue`) по всем регионам, в которых представлена компания.

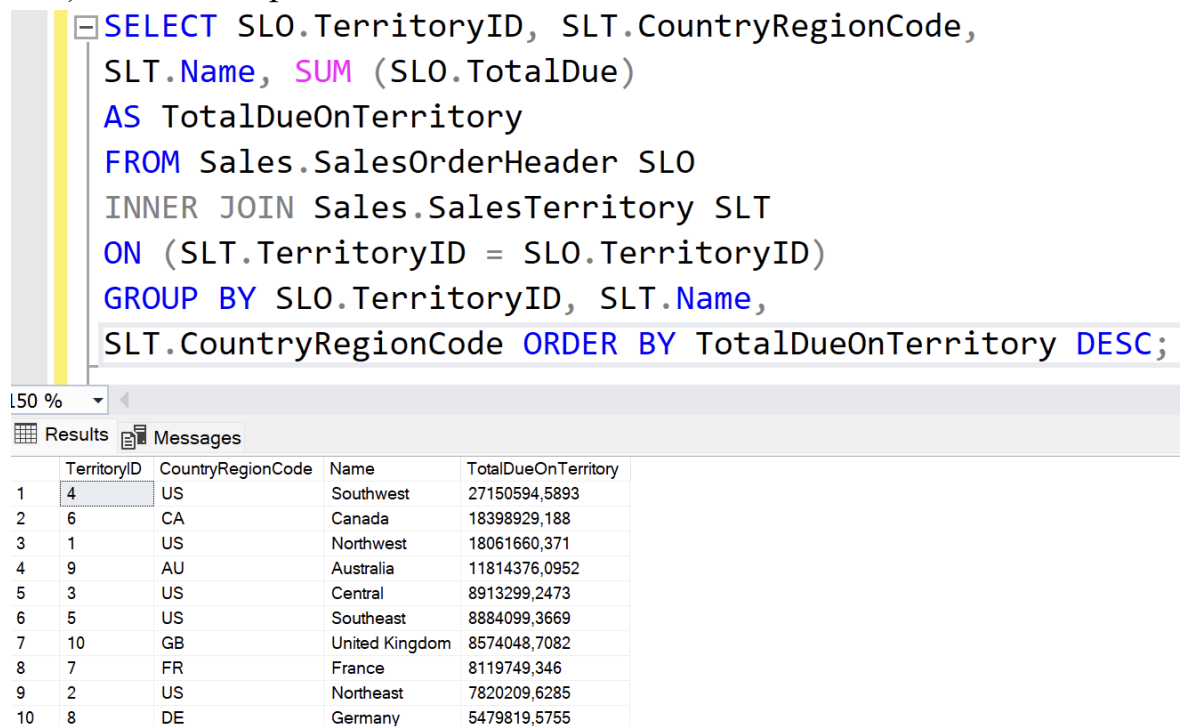
Для получения результата будет использована инструкция группы `SELECT` с внутренним соединением `INNER JOIN`, агрегатной функцией `SUM` и группировкой `GROUP BY` (листинг 1). Несмотря на свою относительную простоту, приведенный запрос включает в себя весь ключевой инструментарий, нужный аналитику для работы.

Листинг 1 – Расчет итоговых показателей продаж `AdventureWorks` по регионам

```
SELECT SLO.TerritoryID, SLT.CountryRegionCode, SLT.Name,
SUM (SLO.TotalDue)
AS TotalDueOnTerritory
FROM Sales.SalesOrderHeader SLO
INNER JOIN Sales.SalesTerritory SLT
ON (SLT.TerritoryID = SLO.TerritoryID)
GROUP BY SLO.TerritoryID, SLT.Name, SLT.CountryRegionCode
ORDER BY TotalDueOnTerritory DESC;
```

Для таблиц запроса, при их объявлении, создаются акронимы (`SLO` = `SalesOrderHeader` и `SLT` = `SalesTerritory`) для удобства работы с кодом. В объявлении столбцов для вывода на экран, а также операций группировки и агрегации будут использованы комбинации коротких акронимов и названий столбцов (через точку), а не длинные имена таблиц. В результате запроса также будет создан столбец `TotalDueOnTerritory`, который будет содержать данные суммирования данных о заказах в рамках одного региона. Внутреннее соединение понадобится для того, чтобы не выводить в результате данные, не закрепленные за конкретным регионом, а также регионы, по которым не были собраны данные. Более подробно код будет разобран на соответствующем

практическом занятии. Результат выполнения запроса (табличный отчет для аналитика) показан на рис. 9.



```

SELECT SLO.TerritoryID, SLT.CountryRegionCode,
SLT.Name, SUM (SLO.TotalDue)
AS TotalDueOnTerritory
FROM Sales.SalesOrderHeader SLO
INNER JOIN Sales.SalesTerritory SLT
ON (SLT.TerritoryID = SLO.TerritoryID)
GROUP BY SLO.TerritoryID, SLT.Name,
SLT.CountryRegionCode ORDER BY TotalDueOnTerritory DESC;

```

	TerritoryID	CountryRegionCode	Name	TotalDueOnTerritory
1	4	US	Southwest	27150594,5893
2	6	CA	Canada	18398929,188
3	1	US	Northwest	18061660,371
4	9	AU	Australia	11814376,0952
5	3	US	Central	8913299,2473
6	5	US	Southeast	8884099,3669
7	10	GB	United Kingdom	8574048,7082
8	7	FR	France	8119749,346
9	2	US	Northeast	7820209,6285
10	8	DE	Germany	5479819,5755

Рисунок 9. Результат выполнения аналитического запроса

Существенно усложним задачу, поставленную перед аналитиком. Теперь, используя данные использованных выше таблиц, исследователь должен получить помесечные (за ноябрь 2011 года и за декабрь 2011 года) данные о финансовых результатах (TotalDue) по регионам, а затем вычислить динамику изменения полученных финансовых показателей в процентах. Скрипт запроса показан в листинге 2.

Листинг 2 – Помесечные итоговые показатели с динамикой

```

WITH NovemberSum AS
(SELECT      SLT.CountryRegionCode,      SLT.Name,      SUM
(SLO.TotalDue) as sumONterr FROM Sales.SalesOrderHeader
SLO
JOIN Sales.SalesTerritory SLT ON (SLT.TerritoryID =
SLO.TerritoryID)
WHERE MONTH (SLO.ShipDate) = 11 AND YEAR (SLO.ShipDate)
= 2011
GROUP BY SLT.Name, SLT.CountryRegionCode),
DecemberSum AS

```

```

(SELECT      SLT.CountryRegionCode,      SLT.Name,      SUM
(SLO.TotalDue) as sumONterr FROM Sales.SalesOrderHeader
SLO
JOIN Sales.SalesTerritory SLT ON (SLT.TerritoryID =
SLO.TerritoryID)
WHERE MONTH (SLO.ShipDate) = 12 AND YEAR (SLO.ShipDate)
= 2011
GROUP BY SLT.Name, SLT.CountryRegionCode)
SELECT ds.CountryRegionCode, ds.Name,
ns.sumONterr as 'Сумма за ноябрь', ds.sumONterr as 'Сумма
за декабрь',
(((ds.sumONterr - ns.sumONterr)/ ns.sumONterr)*100) as
'Увеличение дохода, %'
FROM NovemberSum as ns
LEFT JOIN DecemberSum as ds ON ds.Name = ns.Name;

```

Студенту предлагается перенести код листинга 2 в среду исполнения Management Studio с построчным комментарием, после чего запустить запрос на выполнение. Ожидаемый результат выполнения запроса показан на рис. 10.

Results Messages					
	CountryRegionCode	Name	Сумма за ноябрь	Сумма за декабрь	Увеличение дохода, %
1	AU	Australia	278379,968	253840,415	-8,81
2	CA	Canada	340068,9286	240170,9507	-29,37
3	US	Central	222720,4468	55307,5114	-75,16
4	FR	France	40860,2668	44644,8808	9,26
5	DE	Germany	48894,499	24874,9388	-49,12
6	US	Northeast	179292,0018	65400,4476	-63,52
7	US	Northwest	508212,3726	261818,1082	-48,48
8	US	Southeast	283831,7016	215605,3235	-24,03
9	US	Southwest	721712,4885	269864,509	-62,60
10	GB	United Kingdom	56254,5966	82963,0124	47,47

Рисунок 10. Результат выполнения аналитического отчета с динамикой по месяцам

С целью закрепления полученных знаний студенту предлагается самостоятельно выполнить ряд контрольных заданий с последующей проверкой преподавателем.

2.3. Контрольные задания по теме

1. Внимательно изучите физическую модель базы данных AdventureWorks (можно использовать файл с физическим словарем базы данных, можно скачать по следующей ссылке:

https://msuniversity.ru/uploads/msu_file/file/30311/adventureworks.zip)

2. Разработайте, напишите скрипт и выполните до трех аналитических запросов, включающих в себя объединение таблиц, группировку, агрегатные и оконные функции.

3. Проверенные скрипты сохраните в отчет.

3. СТРУКТУРА ХРАНИЛИЩА ДАННЫХ (НА ОСНОВЕ ХРАНИЛИЩА ADVENTUREWORKS DW)

3.1. Описание учебной базы данных AdventureWorks2019DW.

Развертывание учебной базы данных на сервере MS SQL Server 2019

В отличие от традиционных реляционных баз данных, хранилища данных лучше подходят для решения аналитических задач. Типовая структура хранилища данных спроектирована таким образом, чтобы сделать максимально удобной и быстрой работу с большим количеством числовых значений разной степени детализации, которые являются приоритетом для проведения аналитических исследований. В этой части методических указаний будет рассмотрена типовая структура хранилища данных на примере учебного хранилища данных MS SQL Server.

Поскольку типовые паттерны построения хранилищ данных (паттерн «звезда», паттерн «снежинка») достаточно подробно рассмотрены в лекционных занятиях, далее сфокусируемся на практическом аспекте изучения хранилища данных. Для дальнейшей работы в подготовленный в рамках предыдущих практических занятий экземпляр MS SQL Server необходимо установить учебное хранилище данных AdventureWorksDW.

Учебное хранилище данных AdventureWorksDW хранит операционные результаты деятельности одноименной компании, занимающейся производством и дистрибьюцией велосипедов и запасных частей к ним. Схемы этого хранилища реализованы как в рамках паттерна «звезда», так и в паттерне «снежинка». Хранилище имеет большое количество таблиц фактов и измерений, что позволяет изучить особенности практического применения хранилищ данных.

Для установки базы данных на рабочий экземпляр, следует повторить шаги по установке учебной базы данных из п. 2.1. этих указаний, предварительно скачав по ссылке: <https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms> файл AdventureWorks2016DW.bak. Далее следует повторить процедуру развертывания базы данных на локальном экземпляре сервера баз данных. После окончания восстановления следует убедиться, что база данных появилась в Обозревателе объектов MS SQL Server Management Studio (рис. 11).

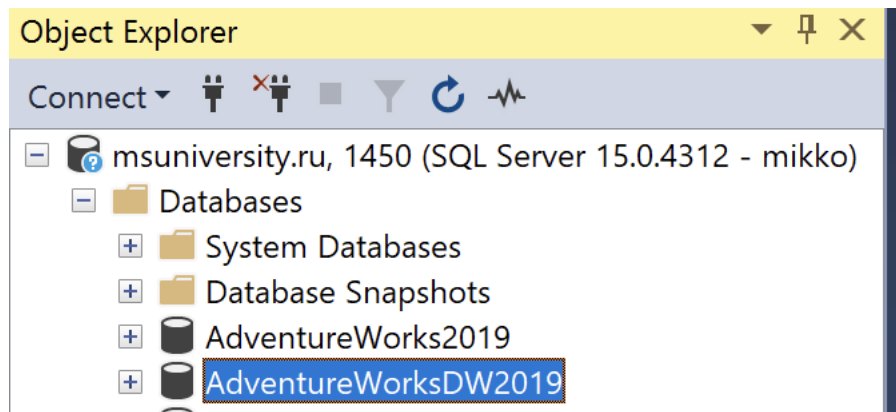


Рисунок 11. Проверка наличия хранилища данных в обозревателе объектов

Далее, приступим к изучению структуры учебного хранилища данных. Построим из фрагмента хранилища данных схему, содержащую одну таблицу фактов, связанную с группой измерений. Для построения диаграммы сперва требуется выдать текущему пользователю права на использование инструмента построения диаграммы. Права выдаются исполнением в базе данных master SQL инструкции `ALTER AUTHORIZATION ON DATABASE :: <Название базы данных> TO <Имя пользователя>`. Скрипт и результат инструкции с параметрами «по умолчанию» (база данных AdventureWorksDW2016, пользователь SA) показан на рис. 12.

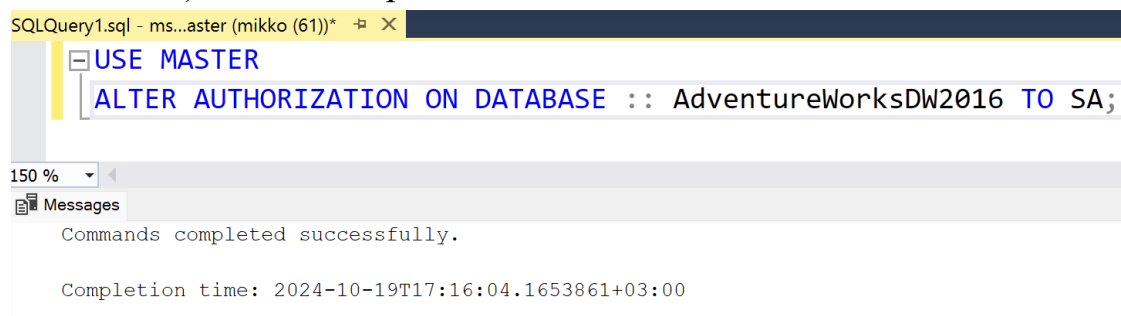


Рисунок 12. Выдача прав пользователю для создания диаграмм базы данных

Для создания диаграммы, в Обозревателе объектов следует открыть папку с целевой базой данных AdventureWorksDW2016, нажать правой клавишей на папку База данных (Database) и выбрать в меню Новая диаграмма (New Database Diagram). Далее следует согласиться с установкой компонентов для построения диаграммы и, если необходимо, повторить действия по ее созданию.

В меню добавления элементов в диаграмму, зажав клавишу клавиатуры Ctrl, следует подсветить следующие элементы хранилища данных: таблицу фактов по продажам через интернет-магазин FactInternetSales, измерение покупателей DimCustomer, измерение дат DimDate, измерение адресов DimGeography, измерение продуктов DimProduct, измерение категорий

продуктов DimProductCategory и измерение подкатегорий продуктов DimProductSubcategory, рис. 13.

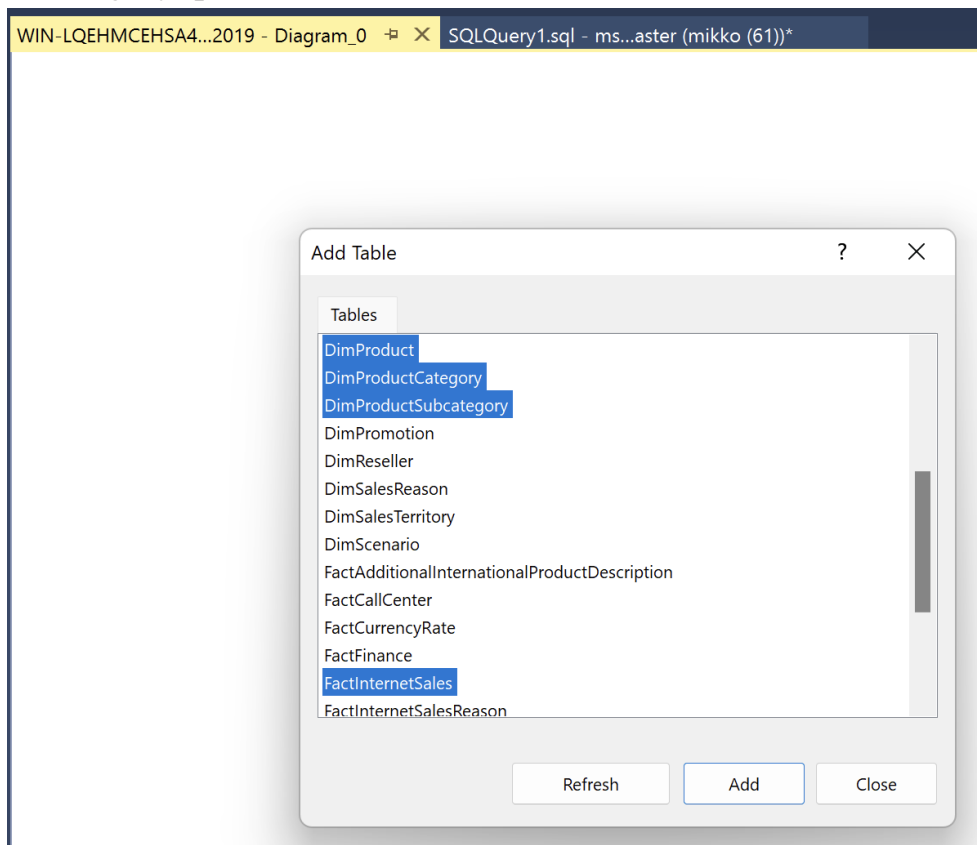


Рисунок 13. Настройка содержимого диаграммы с фрагментом хранилища данных

После упорядочивания выбранных элементов на форме с диаграммой должна получиться схема, похожая на схему на рис. 14.

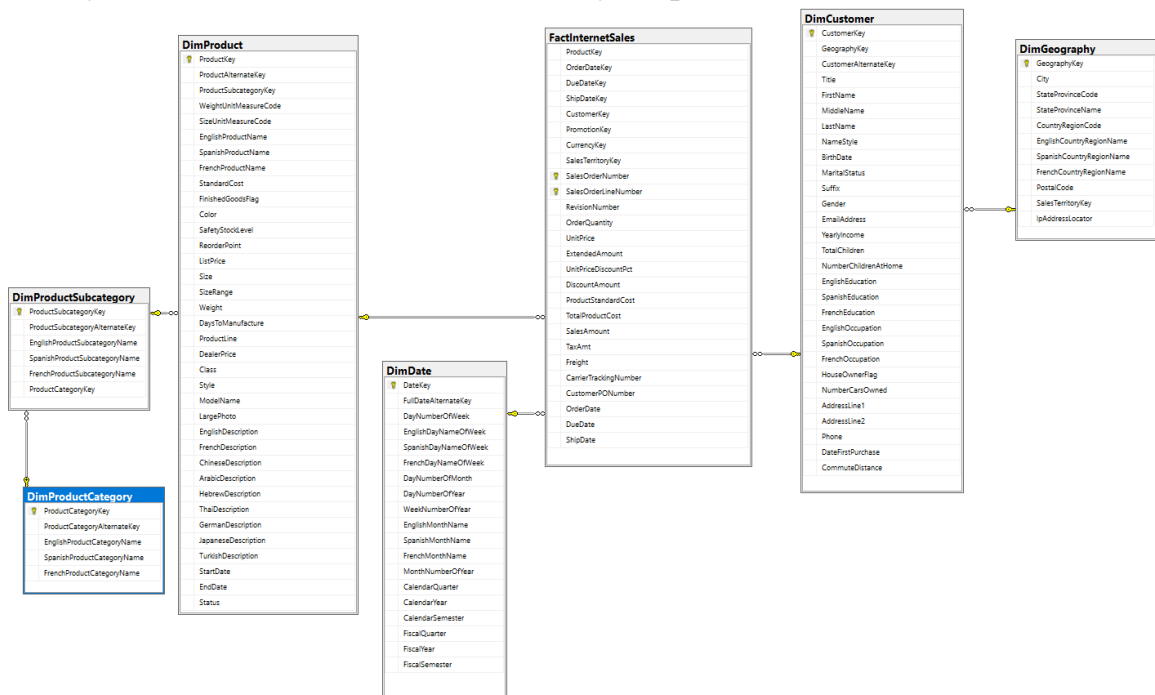


Рисунок 14. Фрагмент хранилища данных

Далее, разберем получившийся фрагмент подробнее. Таблица фактов FactInternetSales является центром фрагмента, представленного на диаграмме. В ней хранится множество числовых значений, отражающих операционные результаты деятельности компании в интернет-пространстве. Окружающие таблицу фактов множественные измерения содержат с себе преимущественно символьные данные, которые будут использоваться в аналитических исследованиях как меры и метки для числовой значимой информации из таблицы фактов.

Обратите внимание на то, что измерение продуктов DimProduct на диаграмме последовательно нормализуется в два других измерения: измерение категорий продуктов DimProductCategory и измерение подкатегорий продуктов DimProductSubcategory, что характерно для паттерна хранилища данных типа «Снежинка» (нормализованные измерения). Сохраните созданную диаграмму с именем Practice_01_01_InternetSales и перейдем к изучению свойств и элементов таблиц измерений.

Главная задача любой таблицы измерений – предоставлять контекстную информацию для мер (числовой значимой информации), которые содержатся в таблицах фактов. Этот факт обуславливает типы столбцов, которые могут встречаться в таблице измерений. Дадим краткие определения всем типам столбцов:

1. Атрибут. Столбцы с дискретными значениями, применяемые при формировании таблиц и сводных графиков (например – диапазоны возрастов людей для проведения исследований).
2. Ключ. Столбец, содержащий только уникальные, неповторяющиеся значения. Однозначно идентифицирующий каждый элемент измерения.
3. Свойство элемента. Столбец, данные которого не нужны для формирования аналитического отчета. Эти данные используются только как пояснительные метки (юридический адрес, адрес электронной почты клиента и т.д.). Столбцы со свойствами элементов могут быть представлены в одном измерении на разных языках, для удобства работы пользователей.
4. Столбец со сведениями о жизненном пути данных. Содержат происхождение и хронологию преобразования данных, информация для администрирования и аудита хранилища данных, закрытая от обычных пользователей.
5. Иерархии. Группа столбцов с разной грануляцией данных. Например, позволяет определить масштаб аналитического исследования по

показателю «Дата» (Год – Месяц – Квартал - Год).

Подготовим диаграмму для изучения типов столбцов в измерении. Повторите действия, указанные на стр. 19-20 для создания новой диаграммы, и добавьте в нее следующие измерения учебного хранилища данных: DimProduct (измерение Продукт), DimProductCategory (измерение Категория продукта), DimProductSubcategory (измерение Подкатегория продукта). Также добавьте не связанное с представленной схемой измерение DimSalesReason (измерение Причина продажи). Итогом проделанной работы должна стать схема, представленная на рис. 15.

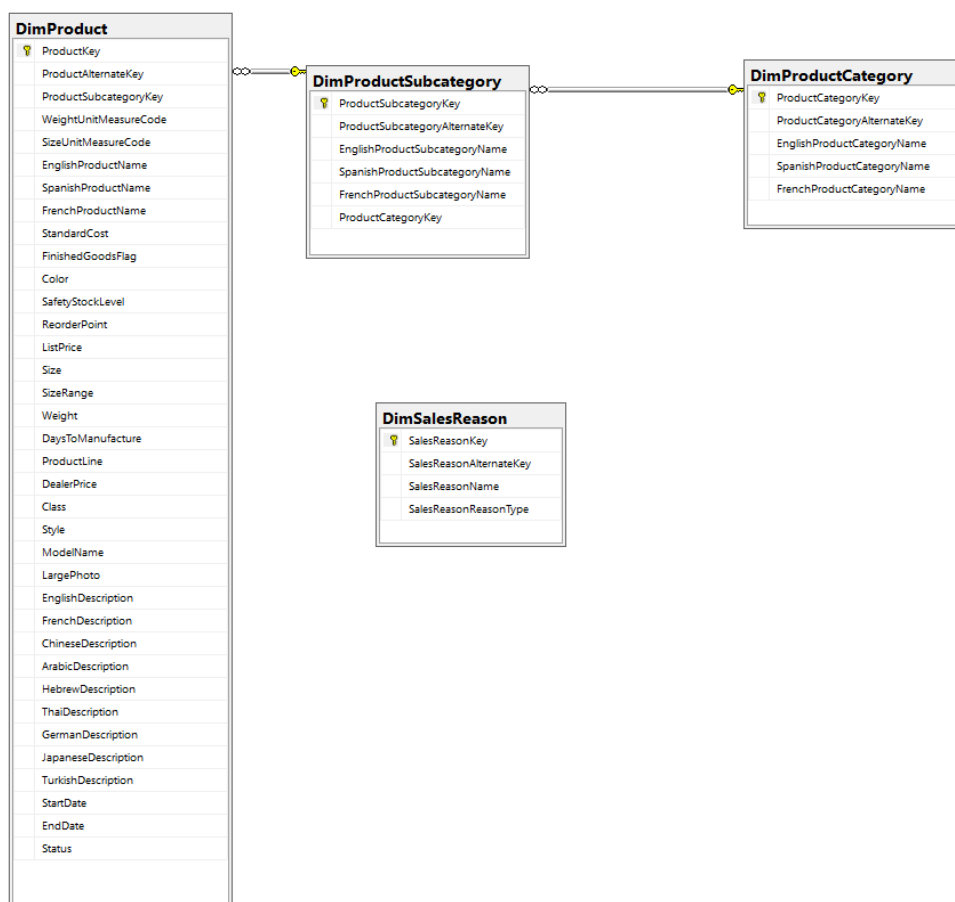


Рисунок 15. Исследование столбцов измерений хранилища данных
Диаграмму следует сохранить с названием Practice_01_02_Dimensions.

Перейдем к исследованию таблиц фактов хранилища данных. Ключевые данные, которыми наполняются таблицы такого типа – это количественные показатели, получаемые в ходе операционной деятельности предприятия или его бизнес-процессов. Эти данные являются основой для аналитических исследований. Также, как и в случае с измерениями, столбцы таблиц фактов могут иметь разные названия и смысл применения.

1. Непосредственно столбец со значимыми количественными показателями называется мерой.

2. Ключи всех связанных с таблицей фактов измерений, перешедшие в столбцы таблицы фактов называются внешними ключами.
3. Столбцы с происхождением и хронологией, по аналогии с аналогичными столбцами в измерениях, называются сведения о жизненном пути данных. Столбцы-меры, в зависимости от применяемых к ним агрегатным функциям (SUM, AVG, MIN, и т.д.) в ходе аналитических исследований, могут быть разных типов.
 1. Аддитивные меры могут быть объединены с помощью агрегатной функции SUM (сумма всех значений) по всем измерениям этой таблицы фактов. Например – сумма продаж.
 2. Неаддитивные меры не могут быть агрегированы по любому измерению своей таблицы фактов. Они или вообще не подвергаются статистическому анализу, или к ним применяется агрегатная функция AVG (среднее арифметическое значение). Например, - цена товара.
 3. К полуаддитивным мерам возможно применение агрегатной функции SUM по всем измерениям, кроме типового измерения времени. Например, баланс банковского счета клиента.

Подготовим диаграмму для изучения типов столбцов в таблицах фактов. Повторите действия, указанные на стр. 19-20 для создания новой диаграммы, и добавьте в нее следующие элементы (два измерения и их таблица фактов) учебного хранилища данных: DimProduct (измерение Продукт), DimDate (измерение Даты), FactProductInventory (таблица фактов с информацией о продуктах на хранении). Итогом проделанной работы должна стать схема, представленная на рис. 16.

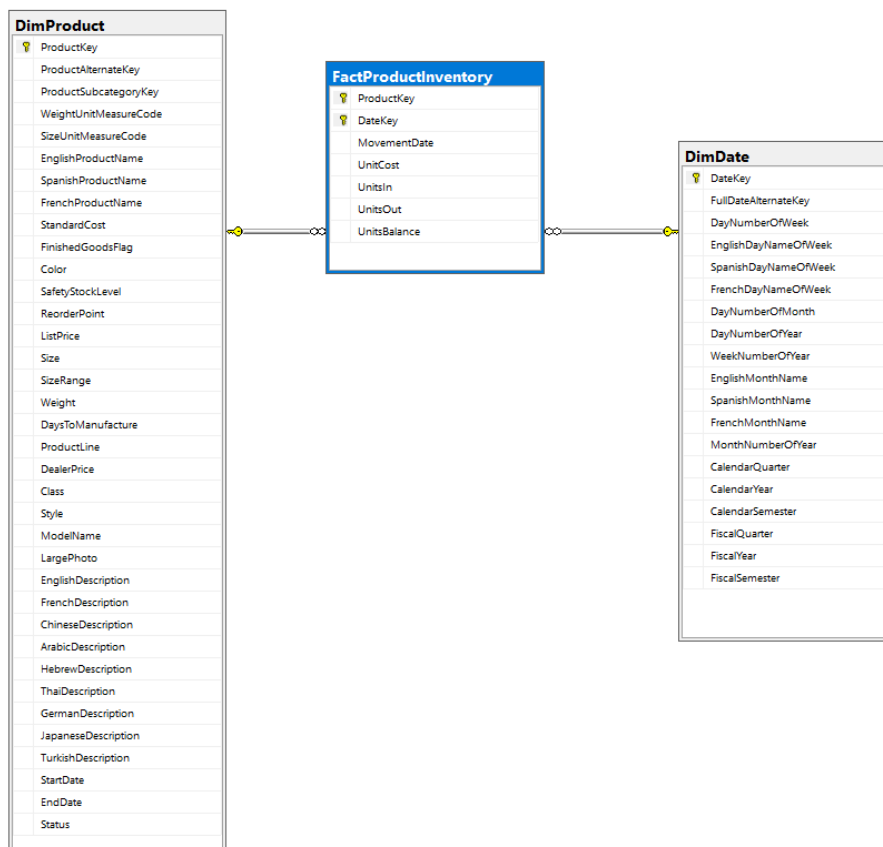


Рисунок 16. Исследование столбцов измерений хранилища данных

Диаграмму следует сохранить с названием Practice_01_03_ProductInventory.

3.2. Контрольные задания по теме

1. Определите для всех столбцов таблиц измерений на рис. 15 их тип относительно классификации: атрибут, ключ, свойство элемента, сведения о жизненном пути данных, иерархия. Полученные результаты оформите в отчет, вместе со скриншотом полученной в ходе практикума диаграммы Practice_01_02_Dimensions.

2. Определите для всех столбцов таблицы фактов на рис. 16 их тип относительно классификации: ключ, ключ измерения, а также меры: аддитивная, неаддитивная, полуаддитивная. Полученные результаты оформите в отчет, вместе со скриншотом полученной в ходе практикума диаграммы Practice_01_03_ProductInventory.

4. ПРОЕКТИРОВАНИЕ ХРАНИЛИЩА ДАННЫХ MS SQL SERVER

4.1. Проектирование физической модели хранилища данных типа «Звезда»

Проект создания хранилища данных, по составу работ в целом соответствует проекту создания базы данных. После постановки задачи происходит создание логической модели хранилища в нотации, аналогичной рассмотренным в главе 3 этой книги. После создания логической модели, программисты пишут скрипт, позволяющий развернуть смоделированное хранилище на сервере. Ниже рассмотрим последовательность действий по созданию скрипта физической модели хранилища.

Предположим, перед проектной группой стоит задача создания и наполнения первичными данными фрагмента хранилища данных, состоящего из трех измерений и одной таблицы фактов. Для удобства, в качестве основы модели возьмем таблицы и данные для их заполнения из учебного хранилища данных AdventureWorksDW.

Для начала создадим два ключевых файла будущего хранилища данных: непосредственно файл хранилища и файл журнала транзакций этого хранилища. Используя контекст системной базы данных MS SQL Server с названием MASTER (для подключения, в скрипте нужно выполнить команду SQL USE master) создадим новую базу данных с произвольным именем (в данном примере, TK463DW) Обратите внимание на то, что ваш путь к физическим файлам хранилища может отличаться от приведенного в листинге. В качестве стартовых параметров хранилища данных определим следующие:

- один файл данных и один журнал транзакций на любом диске;
- начальный размер файла данных 300 Мб, автоувеличение 10 Мб;
- начальный размер файла журнала 50 Мб, автоувеличение 10%.

С целью использования в этом хранилище данных (в измерениях) автоинкрементации, заранее создадим объект с названием Последовательность (SeqCustomerDWKey), где установим следующие параметры автоинкремента: начальное значение – 1, приращение – 1.

Вариант скрипта для решения поставленной задачи показан в листинге 3.

Листинг 3 – Создание базы данных и объекта-последовательности
SeqCustomerDWKey

```
USE master;

IF DB_ID('TK463DW') IS NOT NULL

DROP DATABASE TK463DW;

GO

CREATE DATABASE TK463DW

ON PRIMARY

(NAME = N'TK463DW', FILENAME = N'E:\5
курс_2сем\Проектирование ХД\Пр_3\TK463DW.mdf',

SIZE = 307200KB, FILEGROWTH = 10240KB)

LOG ON

(NAME = N'TK463DW_log', FILENAME = N'E:\5
курс_2сем\Проектирование ХД\Пр_3\TK463DW_log.ldf',

SIZE = 51200KB, FILEGROWTH = 10%);

GO

ALTER DATABASE TK463DW SET RECOVERY SIMPLE WITH NO_WAIT;

GO

USE TK463DW;

GO

CREATE SEQUENCE SeqCustomerDWKey AS INT

    START WITH 1

    INCREMENT BY 1;
```

Последовательно создадим элементы хранилища данных. Поскольку в качестве паттерна хранилища определена схема «Звезда», для нормального функционирования нашему фрагменту понадобится центральная таблица с фактами, и окружающие ее ненормализованные измерения. Первое измерение Customers будет содержать информации о покупателях продукции исследуемой компании. На рис. 17 показан логический словарь создаваемого измерения. Попробуйте сами, следуя приведенным указаниям воспроизвести скрипт в языке SQL для создания измерения по заданным параметрам.

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
CustomerDwKey	INT	NOT NULL	Суррогатный ключ; значения присваиваются с применением последовательности
CustomerKey	INT	NOT NULL	
FullName	NVARCHAR(150)	NULL	Сцепление имени и фамилии из DimCustomer
EmailAddress	NVARCHAR(50)	NULL	
BirthDate	DATE	NULL	
MaritalStatus	NCHAR(1)	NULL	
Gender	NCHAR(1)	NULL	
Education	NVARCHAR(40)	NULL	EnglishEducation из DimCustomer
Occupation	NVARCHAR(100)	NULL	EnglishOccupation из DimCustomer
City	NVARCHAR(30)	NULL	City из DimGeography
StateProvince	NVARCHAR(50)	NULL	StateProvinceName из DimGeography
CountryRegion	NVARCHAR(50)	NULL	EnglishCountryRegionName из DimGeography
Age	Inherited	Inherited	Вычисляемый столбец. Вычислите разницу между BirthDate и текущей датой и дискретизируйте результат, отнеся его к одной из трех групп: <ul style="list-style-type: none"> • если разница меньше либо равна 40, пометьте "Younger" (молодой); • если разница больше 50, пометьте "Older" (пожилой); • в противном случае пометьте "Middle Age" (среднего возраста)
CurrentFlag	BIT	NOT NULL	По умолчанию 1

Рисунок 17. Фрагмент логического словаря хранилища данных с измерением Customers

Для самоконтроля и исправления ошибок можно использовать листинг 4 со скриптом создания измерения Customers.

Листинг 4. Скрипт создания элемента «Измерение Customers»

```
CREATE TABLE Customers
(
CustomerDWKey INT NOT NULL,
CustomerKey INT NOT NULL,
FullName NCHAR(150) NULL,
EmailAdress NCHAR(50) NULL,
BirthDate DATE NULL,
MaritalStatus NCHAR(1) NULL,
Gender NCHAR(1) NULL,
Education NCHAR(40) NULL,
Occupation NCHAR(100) NULL,
City NCHAR(30) NULL,
StateProvince NCHAR(50) NULL,
CountryRegion NCHAR(50) NULL,
Age AS
    CASE
    WHEN BirthDate IS NULL THEN NULL
    WHEN DATEDIFF(yy, BirthDate, CURRENT_TIMESTAMP) > 50
    THEN 'Older'
    WHEN DATEDIFF(yy, BirthDate, CURRENT_TIMESTAMP) > 40
    THEN 'MiddleAge'
    ELSE 'Younger'
    END,
CurrentFlag BIT NOT NULL DEFAULT 1,
CONSTRAINT PK_Customers PRIMARY KEY (CustomerDWKey)
);
GO
```

Следующим элементом физической модели хранилища данных станет измерение Products, которое будет содержать информацию о предлагаемых клиентам продуктах компании. Фрагмент логического словаря с метаданными проектируемого измерения показан на рис. 18.

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
ProductKey	INT	NOT NULL	
ProductName	NVARCHAR(50)	NULL	EnglishProductName из DimProduct
Color	NVARCHAR(15)	NULL	
Size	NVARCHAR(50)	NULL	
SubcategoryName	NVARCHAR(50)	NULL	EnglishProductSubcategoryName из DimProductSubcategory
CategoryName	NVARCHAR(50)	NULL	EnglishProductCategoryName из DimProductCategory

Рисунок 18. Фрагмент логического словаря хранилища данных с измерением Products

В целях самоконтроля и исправления ошибок можно использовать листинг 5 со скриптом создания измерения Products.

Листинг 5. Скрипт создания измерения Products

```
CREATE TABLE Products
(
    ProductKey INT NOT NULL,
    ProductName NCHAR(50) NULL,
    Color NCHAR(15) NULL,
    Size NCHAR(50) NULL,
    SubcategoryName NCHAR(50) NULL,
    CategoryName NCHAR(50) NULL,
    CONSTRAINT PK_Products PRIMARY KEY (ProductKey)
);
GO
```

Создадим для удобства грануляции дат в аналитических исследованиях измерение Dates. Фрагмент логического словаря с метаданными проектируемого измерения показан на рис. 19. Воспроизведите в скрипте, написанном в языке SQL заданные ограничения и параметры.

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
DateKey	INT	NOT NULL	
FullDate	DATE	NOT NULL	FullDateAlternateKey из DimDate
MonthNumberName	NVARCHAR(15)	NULL	Сцепите MonthNumberOfYear (с ведущими нулями для номеров месяцев меньших 10) и EnglishMonthName из DimDate
CalendarQuarter	TINYINT	NULL	
CalendarYear	SMALLINT	NULL	

Рисунок 19. Фрагмент логического словаря хранилища данных с измерением Dates

В целях коррекции ошибок скрипта можно использовать листинг 6 со скриптом создания измерения Dates.

Листинг 5. Скрипт создания измерения Dates

```
CREATE TABLE Dates
(
    DateKey INT NOT NULL,
    FullDate DATE NOT NULL,
    MonthNumberName NCHAR(15) NULL,
    CalendarQuarter TINYINT NULL,
    CalendarYear SMALLINT NULL,
    CONSTRAINT PK_Dates PRIMARY KEY (DateKey)
);
GO
```

Наконец создадим главный элемент проектируемого фрагмента хранилища данных, - таблицу фактов InternetSales, которая будет содержать информацию о количественных значениях продаж продуктов компании покупателям. Обратите внимание на фрагмент логического словаря данных для этой таблицы, приведенный на рис. 20. Половину столбцов этой таблицы представляют собственный ключ фактов и ключи измерений, собранные со всех, окружающих таблицу фактов измерений, созданных на предыдущих этапах проектирования. Руководствуясь параметрами и метаданной, приведенной в логическом словаре, напишите скрипт в языке SQL для создания требуемой таблицы фактов.

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
InternetSalesKey	INT	NOT NULL	IDENTITY(1,1)
CustomerDwKey	INT	NOT NULL	Используя бизнес-ключ CustomerKey из измерения Customers, найдите подходящее значение суррогатного ключа CustomerDwKey из измерения Customers
ProductKey	INT	NOT NULL	
DateKey	INT	NOT NULL	OrderDateKey из FactInternetSales
OrderQuantity	SMALLINT	NOT NULL	По умолчанию 0
SalesAmount	MONEY	NOT NULL	По умолчанию 0
UnitPrice	MONEY	NOT NULL	По умолчанию 0
DiscountAmount	FLOAT	NOT NULL	По умолчанию 0

Рисунок 20. Фрагмент логического словаря хранилища данных с таблицей фактов InternetSales

Как и в предыдущих случаях, важно попытаться написать скрипт языка SQL самому, а в целях самоконтроля в случае возникновения ошибок при выполнении скрипта, использовать для проверки листинг 6.

Листинг 6. Скрипт создания таблицы фактов InternetSales

```
CREATE TABLE InternetSales
(
    InternetSalesKey INT NOT NULL IDENTITY(1,1),
    CustomerDWKey INT NOT NULL,
    ProductKey INT NOT NULL,
    DateKey INT NOT NULL,
    OrderQuantity SMALLINT NOT NULL DEFAULT 0,
    SalesAmount MONEY NOT NULL DEFAULT 0,
    UnitPrice MONEY NOT NULL DEFAULT 0,
    DiscountAmount FLOAT NOT NULL DEFAULT 0,
CONSTRAINT          PK_InternetSales          PRIMARY          KEY
(InternetSalesKey)
);
GO

ALTER          TABLE          InternetSales          ADD          CONSTRAINT
FK_InternetSales_Customers FOREIGN KEY (CustomerDWKey)
REFERENCES Customers (CustomerDWKey);
```

```

ALTER TABLE InternetSales ADD CONSTRAINT
FK_InternetSales_Products FOREIGN KEY (ProductKey)
REFERENCES Products (ProductKey);

ALTER TABLE InternetSales ADD CONSTRAINT
FK_InternetSales_Dates FOREIGN KEY (DateKey)
REFERENCES Dates (DateKey);
GO

ALTER AUTHORIZATION ON DATABASE :: TK463DW TO [sa];

```

Полученные в ходе проектирования результаты продемонстрируйте с помощью диаграммы MS SQL Server, сохранив ее под именем InternetSalesDW (рис. 21).

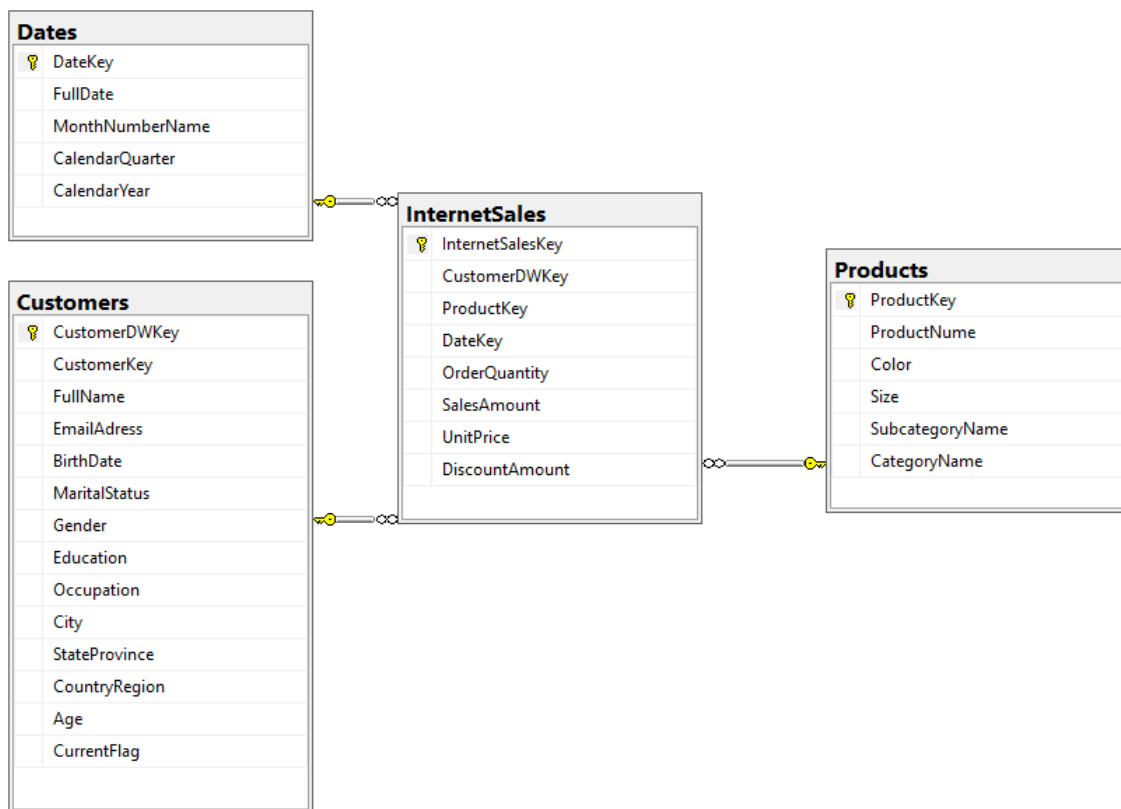


Рисунок 21. Диаграмма хранилища данных InternetSalesDW

Изучив полученную диаграмму, убедимся в том, что полученное хранилище действительно является образцом паттерна типа «Звезда».

4.2. Тестирование физической модели хранилища данных типа «Звезда»

После создания структуры хранилища данных следует заполнять его данными для проведения тестирования его работы с аналитическими запросами и дальнейшего ввода в эксплуатацию. Для заполнения тестовыми данными, используем данные из учебного хранилища, используемого ранее. Последовательно заполним созданные измерения хранилища данных Customers, Products, Dates данными из соответствующих таблиц из хранилища данных AdventureWorksDW2016: DimCustomer, DimProduct, DimDate. Для копирования и переноса данных будут использованы вложенные запросы типа 2 (nested query type II) с комбинацией инструкций INSERT INTO (вставка) и SELECT (выборка). Попробуйте написать и исполнить скрипт переноса данных самостоятельно. Для самоконтроля можно использовать скрипт из листинга 7.

Листинг 7. Скрипт переноса данных в измерения Customers, Products, Dates

```
INSERT INTO Customers (CustomerDWKey, CustomerKey,
FullName, EmailAdress, BirthDate, MaritalStatus, Gender,
Education, Occupation, City, StateProvince,
CountryRegion)
SELECT
NEXT VALUE FOR SeqCustomerDWKey AS CustomerDWKey,
C.CustomerKey, C.FirstName + ' ' + C.LastName AS FullName,
C.EmailAddress, C.BirthDate, C.MaritalStatus, C.Gender,
C.EnglishEducation, C.EnglishOccupation, G.City,
G.StateProvinceName, G.EnglishCountryRegionName
FROM AdventureWorksDW2016.dbo.DimCustomer AS C
INNER JOIN AdventureWorksDW2016.dbo.DimGeography AS G
ON C.GeographyKey = G.GeographyKey;
GO

-----

INSERT INTO Products (ProductKey, ProductName, Color,
Size, SubcategoryName, CategoryName)
SELECT P.ProductKey, P.EnglishProductName, P.Color,
P.Size, S.EnglishProductSubcategoryName,
C.EnglishProductCategoryName
```

```

FROM AdventureWorksDW2016.dbo.DimProduct AS P INNER JOIN
AdventureWorksDW2016.dbo.DimProductSubcategory AS S
ON P.ProductSubcategoryKey = S.ProductSubcategoryKey
INNER JOIN AdventureWorksDW2016.dbo.DimProductCategory
AS C
ON S.ProductCategoryKey = C.ProductCategoryKey;
GO

-----

INSERT INTO Dates (DateKey, FullDate, MonthNumberName,
CalendarQuarter, CalendarYear)
SELECT DateKey, FullDateAlternateKey, SUBSTRING (CONVERT
(CHAR (8), FullDateAlternateKey, 112), 5, 2) + ' ' +
EnglishMonthName, CalendarQuarter, CalendarYear
FROM AdventureWorksDW2016.dbo.DimDate;

GO

```

После заполнения данными измерений, повторите те же действия для таблицы фактов созданного фрагмента хранилища данных. Загрузите таблицу фактов InternetSales данными из соответствующих таблиц из хранилища данных AdventureWorksDW2016 (FactInternetSales). Попробуйте написать и исполнить скрипт переноса данных самостоятельно. Для самоконтроля можно использовать скрипт из листинга 8.

Листинг 7. Скрипт переноса данных в таблицу фактов InternetSales

```

INSERT INTO InternetSales (CustomerDWKey, ProductKey,
DateKey, OrderQuantity, SalesAmount, UnitPrice,
DiscountAmount)
SELECT
C.CustomerDWKey, FIS.ProductKey, FIS.OrderDateKey,
FIS.OrderQuantity, FIS.SalesAmount, FIS.UnitPrice,
FIS.DiscountAmount
FROM
AdventureWorksDW2016.dbo.FactInternetSales AS FIS
INNER JOIN Customers AS C
ON FIS.CustomerKey = C.CustomerKey;

```

Проверьте функционал построенного хранилища данных, составив и выполнив аналитический запрос к данным. Пример аналитического запроса приведен в листинге 8, а фрагмент результата его выполнения – на рисунке 22.

Листинг 8. Скрипт запроса к созданному хранилищу данных

```
SELECT C.CountryRegion, P.CategoryName, D.CalendarYear,
SUM (I.SalesAmount) AS Sales
FROM InternetSales AS I
INNER JOIN Customers AS C
ON I.CustomerDWKey = C.CustomerDWKey
INNER JOIN Products AS P
ON I.ProductKey = P.ProductKey
INNER JOIN Dates AS D
ON I.DateKey = D.DateKey
GROUP BY C.CountryRegion, P.CategoryName, D.CalendarYear
ORDER BY C.CountryRegion, P.CategoryName, D.CalendarYear;
```

	CountryRegion	CategoryName	CalendarYear	Sales
1	Australia	Accessories	2012	573,99
2	Australia	Accessories	2013	132763,21
3	Australia	Accessories	2014	5353,43
4	Australia	Bikes	2010	20909,78
5	Australia	Bikes	2011	2563732,2493
6	Australia	Bikes	2012	2127687,0151
7	Australia	Bikes	2013	4139720,96
8	Australia	Clothing	2012	146,45
9	Australia	Clothing	2013	66959,21
10	Australia	Clothing	2014	3154,29
11	Canada	Accessories	2012	56,97
12	Canada	Accessories	2013	96922,04
13	Canada	Accessories	2014	6398,84
14	Canada	Bikes	2010	3578,27
15	Canada	Bikes	2011	571571,7984
16	Canada	Bikes	2012	307497,5637
17	Canada	Bikes	2013	938654,76
18	Canada	Clothing	2012	49,99
19	Canada	Clothing	2013	50055,85
20	Canada	Clothing	2014	3058,78
21	France	Accessories	2012	442,12
22	France	Accessories	2013	60599,81
23	France	Accessories	2014	2364,85

Рисунок 22. Фрагмент результата выполнения аналитического запроса к хранилищу данных

Таким образом, в результате реализации проекта, была создана и протестирована рабочая физическая модель хранилища данных.

4.3. Контрольные задания по теме

1. Определите предметную область хранилища данных в соответствии с личными предпочтениями.
2. Выбрав архитектуру «Звезда» или «Снежинка» (или смешанную), спроектируйте модель хранилища данных, содержащую как минимум одну таблицу фактов и 3-4 измерения.
3. Напишите скрипты развертывания модели хранилища и заполнения его 15-20 экземплярами фактов (также заполните достаточным количеством экземпляров измерения).
4. В отчете приведите диаграмму полученного хранилища данных, а также скриншоты с результатами выполнения 2-3 аналитических запросов к нему.

5. ИМПОРТ И ЭКСПОРТ ДАННЫХ СРЕДСТВАМИ MS SQL SERVER

5.1. Экспорт и импорт данных с помощью Мастера экспорта и импорта данных MS SQL Server

Время от времени, компании сталкиваются с необходимостью принять или передать существенные по объему данные. Эти данные в конечной точке (сервер баз данных, или хранилище данных) должны быть упорядочены, приведены в требуемый вид и сохранены. Если задача по объему небольшая и рутинная, могут быть использованы встроенные в программное обеспечение сервера дополнительные программные средства или функции на уровне языка SQL. Если задача большая и комплексная, применяются более мощные средства ETL. В этой главе речь пойдет о решении сравнительно легких задач экспорта и импорта.

Мастер экспорта и импорта данных, с помощью которого можно решать простые и рутинные задачи экспорта и импорта данных входит в комплект поставки MS SQL Server, как отдельное программное обеспечение. Далее, разберем возможности и ключевые принципы работы с этим программным продуктом.

Для начала работы с мастером, следует загрузить программный продукт из комплекта поставки MS SQL Server (рисунок 23).

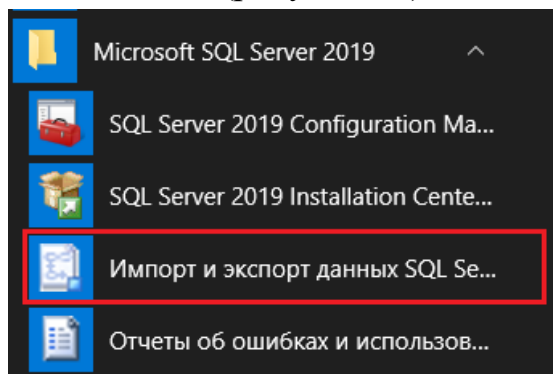


Рисунок 23. Загрузка мастера экспорта и импорта SQL Server

Для запуска мастера в работу, пользователю необходимо указать параметры входного (данные забираются) и целевого (данные размещаются) источника, а также сформировать массив данных, который будет передан в ходе выполнения процедуры. Далее будет разобран пример экспорта данных в рамках одного ядра базы данных MS SQL Server с небольшими изменениями схемы экспортируемых таблиц. Остальные возможности экспорта и импорта данных, предоставляемые

сервисом экспорта и импорта MS SQL Server (рис. 24) предлагается изучить студенту самостоятельно.

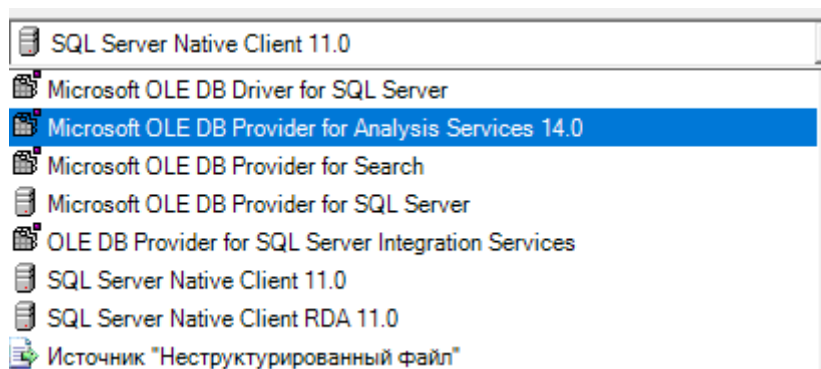


Рисунок 24. Фрагмент списка доступных входных источников данных в Мастере экспорта и импорта данных

В данном примере воспользуемся установленной ранее на экземпляр сервера учебной базой данных AdventureWorks2016 (см. п. 2.17 этой книги) и осуществим экспорт в новую базу данных схемы и содержимого двух таблиц, предварительно скорректировав их схему (название).

Работа мастера заключается в последовательном описании источника и назначения данных. При определении назначения данных, массив можно также можно изменить схему таблицы, после чего, выполнить запланированную процедуру.

Для начала работы, в окне Выбор источника данных укажем путь к ядру базы данных и к учебной базе данных AdventureWorks (вариант подключения для выделенного сервера показан на рис. 25).

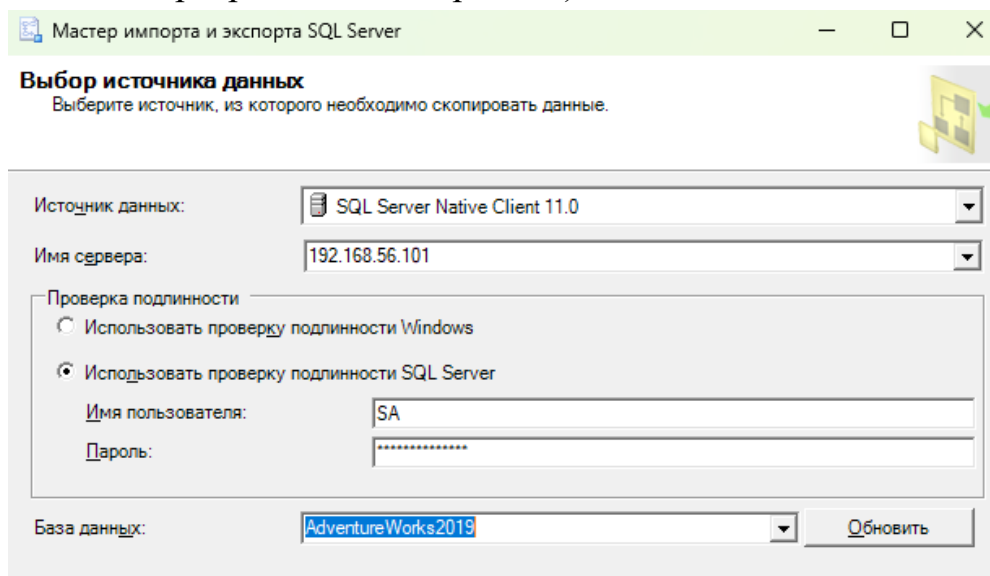


Рисунок 25. Настройка подключения к Источнику данных

Обратите внимание, что в вашем случае, параметры подключения могут быть иными.

После подключения к источнику (а если вы смогли в ниспадающем меню База данных найти и выбрать установленную на ядре учебную базу данных, то все в порядке), следует перейти к настройке Назначения данных. Нажав кнопку далее, перейдем к следующей форме Мастера с указанием параметров Назначения данных (места, куда данные будут экспортированы). В качестве целевого пункта укажем тот же самое ядро сервера базы данных, как и в Источниках данных ранее, но выбор базы данных при этом следует оставить пустым (рис. 26).

The screenshot shows the 'Master of Import and Export SQL Server' window, specifically the 'Select Destination' (Выбор назначения) step. The title bar reads 'Мастер импорта и экспорта SQL Server'. Below the title, the text 'Выбор назначения' is followed by the instruction 'Укажите, куда копировать данные.' (Specify where to copy the data). The main area contains several fields: 'Назначение:' (Destination) is set to 'SQL Server Native Client 11.0'; 'Имя сервера:' (Server name) is '192.168.56.101'. Under the 'Проверка подлинности' (Authentication) section, the 'Использовать проверку подлинности SQL Server' (Use SQL Server authentication) radio button is selected. Below this, the 'Имя пользователя:' (Username) is 'SA' and the 'Пароль:' (Password) is masked with dots. At the bottom, the 'База данных:' (Database) dropdown is set to '<по умолчанию>' (default). To the right of this dropdown are two buttons: 'Обновить' (Update) and 'Создать...' (Create...).

Рисунок 26. Настройка подключения к Назначению данных

Перед тем, как перейти далее, создадим целевую базу данных. Для этого на форме на рис. 26 следует нажать кнопку Создать. При создании базы данных следует указать разные параметры файла базы данных и файла журнала транзакций. В этом руководстве пропустим этот момент и оставим параметры «по умолчанию», лишь указав название новой базы данных (рис. 27).

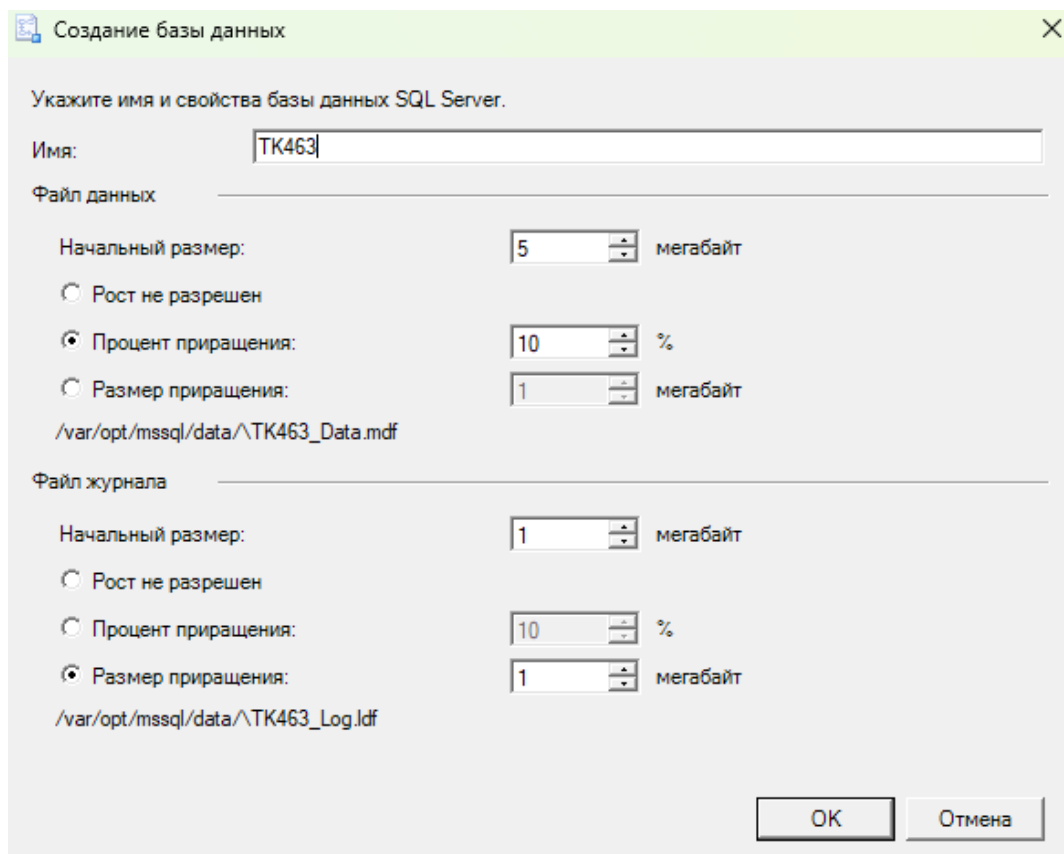


Рисунок 27. Создание целевой базы данных для процедуры экспорта данных

Нажав кнопку **Ок**, вернитесь к прошлому окну Мастера. После нажатия кнопки **Далее**, Мастер предложит выбрать вариант экспорта данных из двух существующих: экспорт данных с выбором таблиц и представлений или экспорт данных, предварительно выбранных пользовательским SQL запросом (рис. 28). Выберем вариант с экспортом данных из таблиц и представлений целиком.

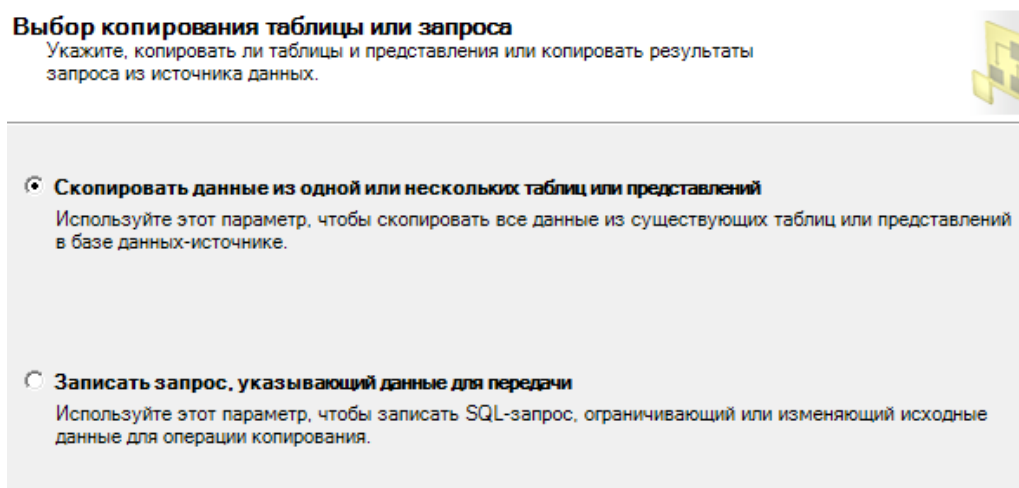


Рисунок 28. Выбор варианта копирования (экспорта) данных

Поскольку был выбран вариант с копирование таблиц целиком, после нажатия кнопки **Далее** будет выведен список всех таблиц и представлений базы данных, выбранной ранее в качестве источника (рис. 29). В соответствии с рисунком, выберите галочкой представления из нижней части списка, а также

для удобства, уберите в Назначении из названия обеих таблиц букву v (view, представление), как это сделано в примере на рисунке.

Выбор исходных таблиц и представлений

Выберите одну или несколько таблиц или представлений для копирования.

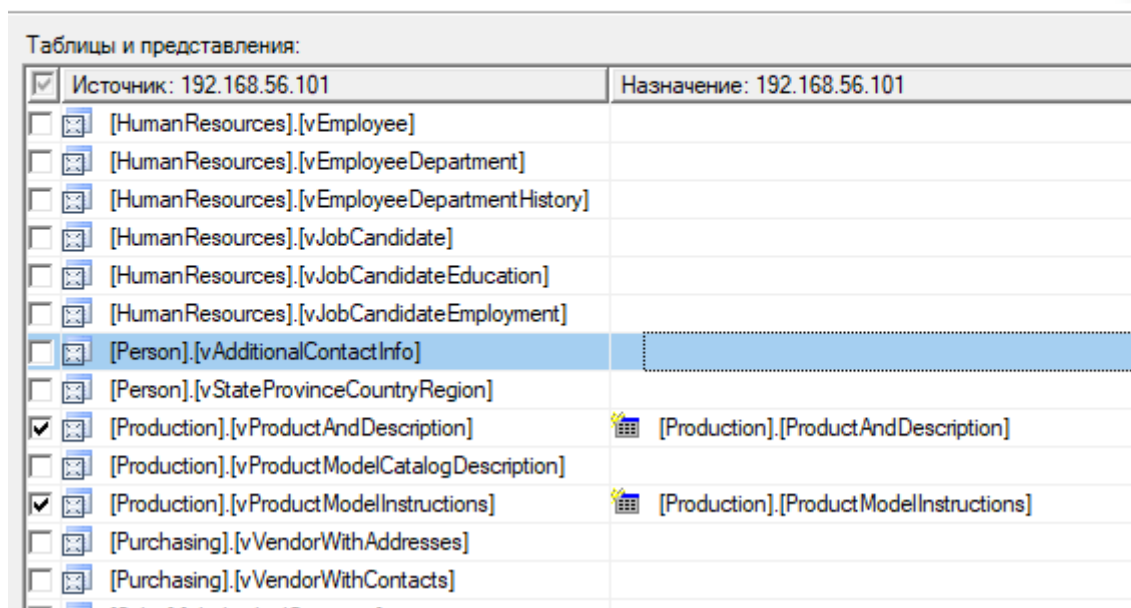


Рисунок 29. Выбор таблиц и представлений для копирования (экспорта) данных

Перед тем, как осуществить процедуру экспорта, ознакомимся с двумя дополнительными возможностями Мастера экспорта и импорта. Перед осуществлением экспорта, используя форму выбора исходных таблиц, можно посмотреть содержимое этих таблиц, выбрав таблицу из списка и нажав кнопку Просмотр (рис. 30), а также изменить параметры схемы и метаданные таблицы для ее экспорта, выбрав таблицу из списка и нажав кнопку Изменить (рис. 31). Ознакомьтесь с предлагаемыми инструментами самостоятельно.

Форма Просмотр позволяет со структурой экспортируемых данных выбранной таблицы или представления. В окне показан SQL-запрос с помощью, которого будет формироваться массив данных при экспорте, а также экземпляры, входящие в этот массив.

Форма Сопоставления столбцов, в свою очередь, позволяет внести изменения в SQL-запрос к таблице или представлению, а также внести необходимые изменения в схему таблицы или представления, подлежащие экспорту.

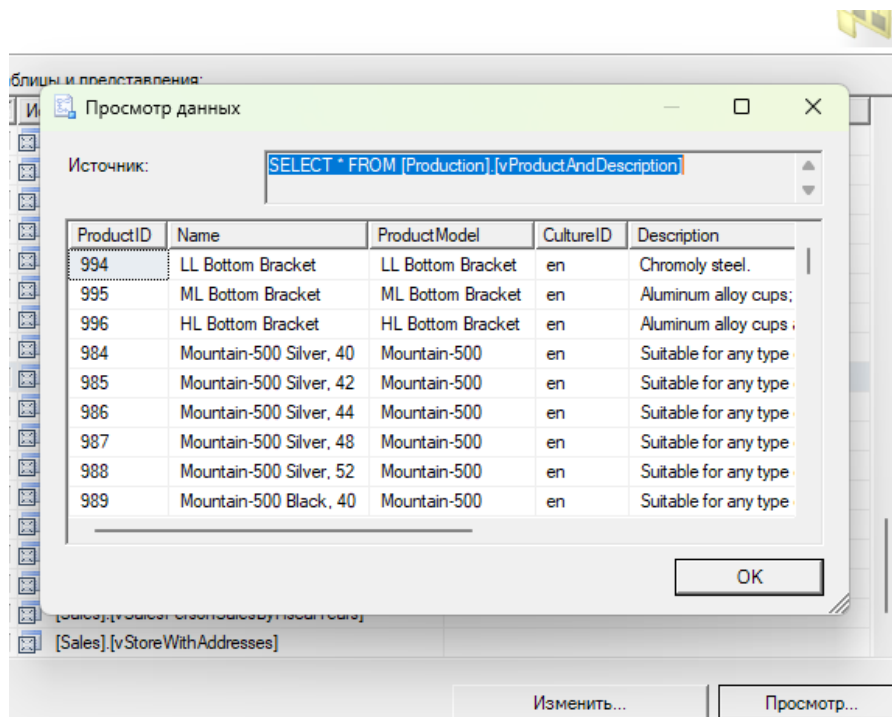


Рисунок 30. Форма «Просмотр данных» Мастера экспорта и импорта

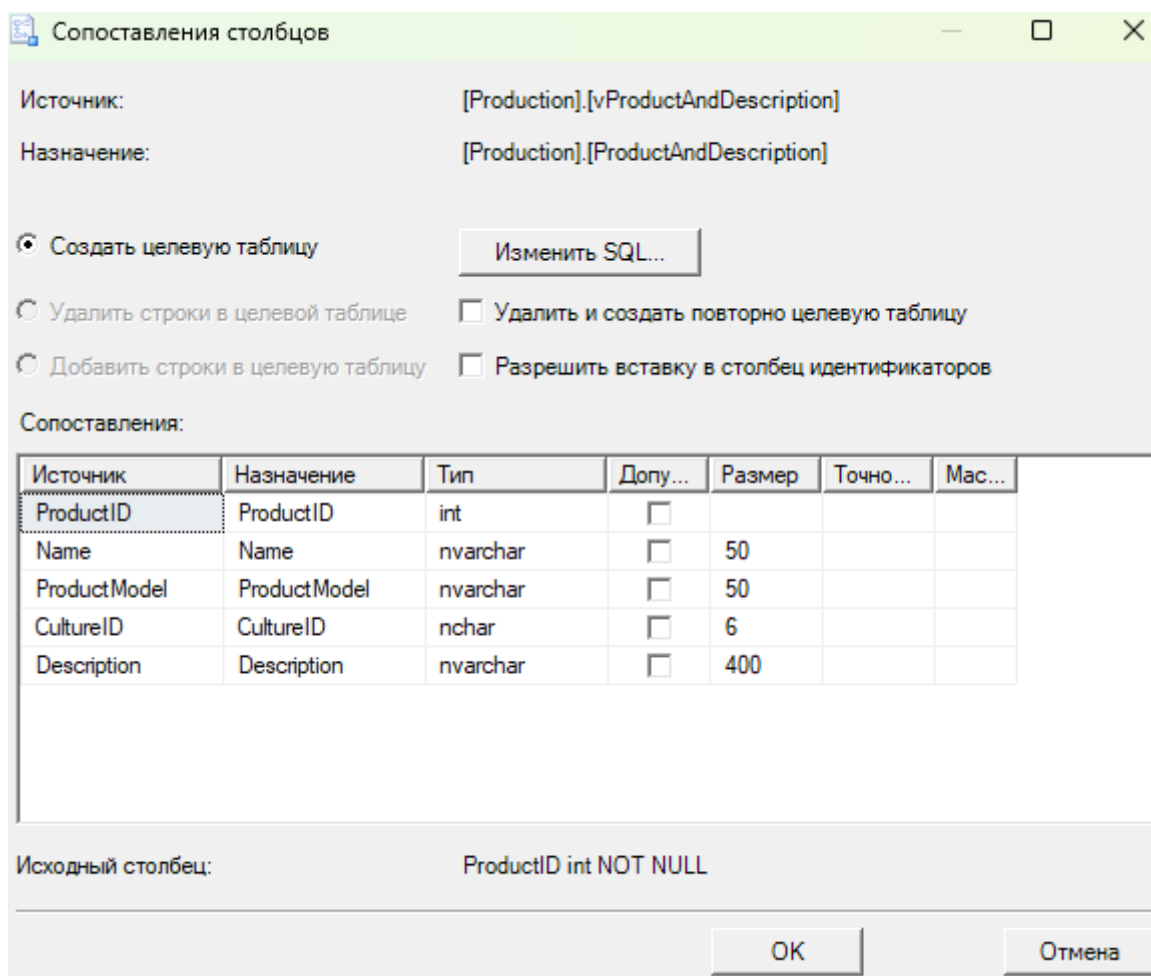


Рисунок 31. Форма Сопоставления столбцов Мастера экспорта и импорта

В рамках решения заданной задачи изменения в сопоставлении столбцов не требуется. Следует нажатием кнопки Далее перейти в следующей экранной форме мастера. Экранная форма «Сохранение и запуск пакета» (рис. 32) регламентирует дальнейшие действия с настроенной процедурой экспорта или импорта данных. Так, если планируется единоразовое ее исполнение, достаточно оставить галочку у пункта «Запустить немедленно» и перейти к исполнению процедуры. Если созданная процедура будет выполняться итеративно, то имеет смысл сохранить ее как ETL-пакет и дополнительно поставить галочку в опции «Сохранить пакет служб SSIS» в виде отдельного файла или же сразу на ядре сервера MS SQL Server. Подробнее про ETL пакеты и про SSIS будет рассказано в следующей главе книги.

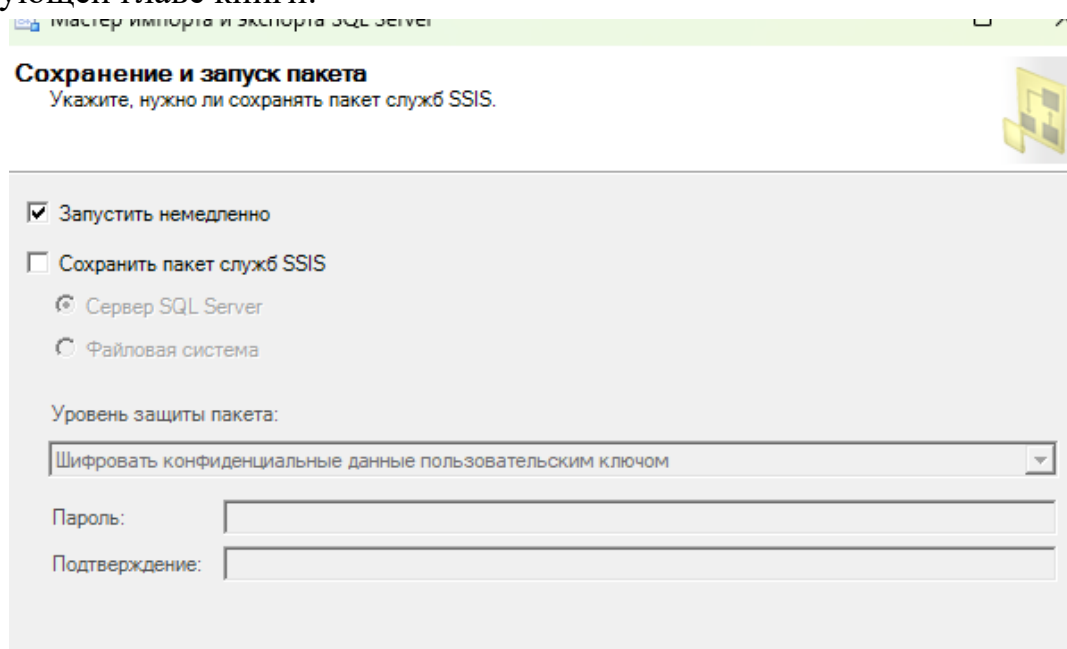


Рисунок 32. Управление пакетом служб SSIS

Поскольку процедура экспорта или импорта может быть комплексной (включать в себя значительное количество этапов), перед ее запуском будет целесообразно еще раз проверить структуру запланированной операции экспорта или импорта. После очередного нажатия кнопки Далее будет показан отчет «Завершение работы Мастера» с детальным планом будущей процедуры (рис. 33).

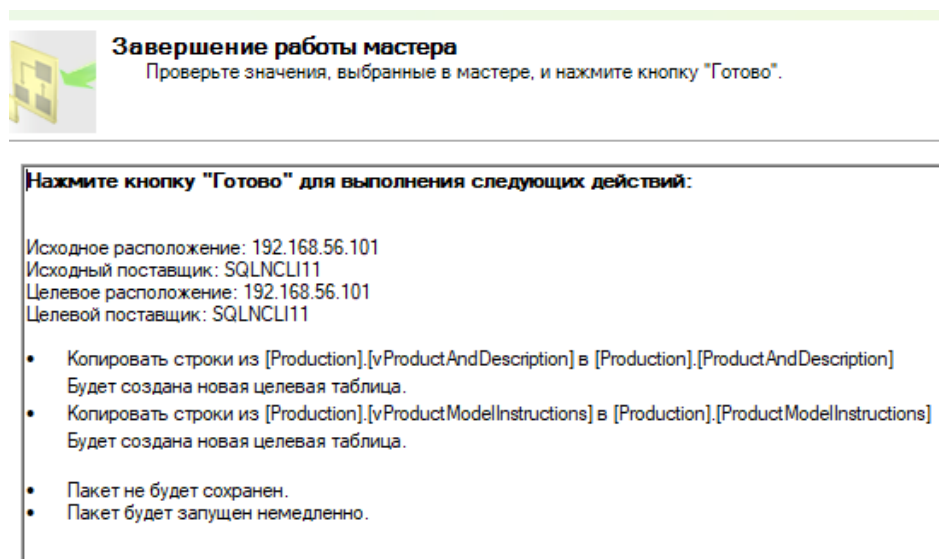


Рисунок 33. План выполнения процедуры экспорта и импорта данных

Для выполнения созданного пакета следует нажать на кнопку Готово. В режиме реального времени будет отслеживаться результативность каждого этапа пакета экспорта или импорта. В случае успешного прохождения всех этапов, результаты будут сохранены в базе данных (рис. 34).

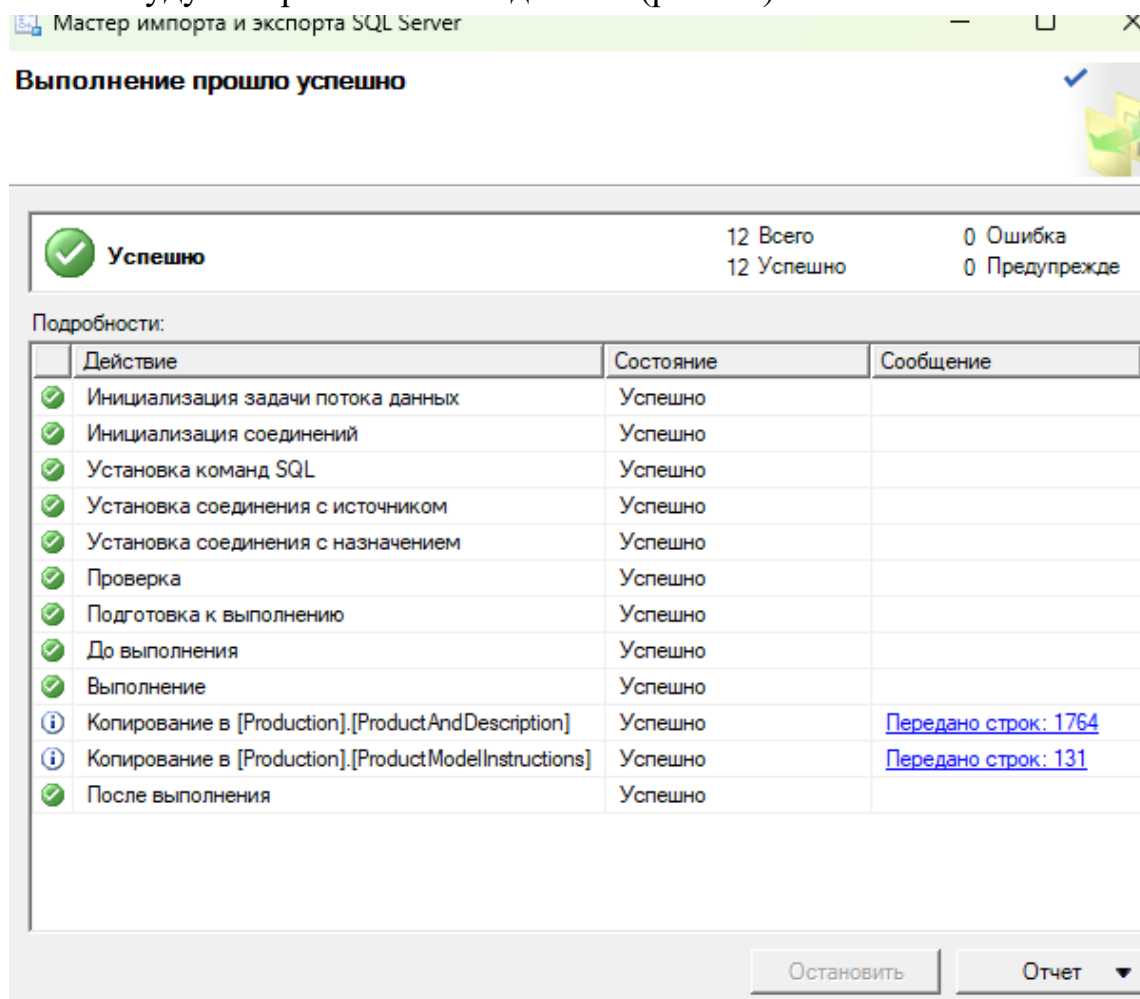


Рисунок 34. Мониторинг выполнения пакета экспорта и результаты

Таким образом решаются относительно несложные и зачастую неитеративные задачи экспорта и импорта данных. Для более комплексных задач, а также задач, которые будут регулярно повторяться, удобнее разработать полноценную ETL-процедуру с использованием специального программного обеспечения. Об этом будет рассказано в следующей главе этой книги.

5.2. Контрольные задания по теме

1. В свободных источниках интернета найдите подходящий массив «открытых данных» (например, открытые данные министерства культуры РФ <https://opendata.mkrf.ru>). Изучите описание и метаданные предлагаемых наборов данных, выберите подходящий (минимум очистки данных) и загрузите файл набора данных в формате *.csv. Создайте в вашем экземпляре сервера новую целевую базу данных с названием OpenData. Попробуйте, используя средства импорта и экспорта MS SQL Server переместить данные из полученного файла *.csv в целевую таблицу в созданной базе данных.

Проанализируйте процесс импорта данных в отчете, описав возникнувшие в процесса mapping проблемы в таблице формата: код проблемы – суть проблемы – варианты решения проблемы.

2. В свободных источниках интернета найдите подходящий для экспорта JSON-документ небольшого объема, просмотрите структуру JSON-документа. Перенесите данные JSON-документа в строку временной таблицы #test_json_import в текущей целевой базе данных. Ознакомьтесь с содержимым полученной таблицы. Сделайте значимый скриншот с результатами.

6. ПРОЕКТИРОВАНИЕ ПОТОКОВ УПРАВЛЕНИЯ SSIS ДЛЯ MS SQL SERVER (ETL-ПРОЦЕДУРЫ)

6.1. Базовое описание инструментария SSIS для проектирования ETL-процедур

На уровне хранилищ данных, учитывая большое количество и разнородность входящих потоков данных, процедуры импорта/экспорта данных могут быть существенно более сложными и многоступенчатыми. Процессы, способствующие решению этих сложных задач, называются ETL (extract transform load). Для демонстрации практического создания и реализации этих процессов в профессиональном программном обеспечении, рассмотрим пример создания ETL процедуры средствами MS SQL Server.

Для начала развернем необходимое для работы программное обеспечение. ETL процессы для хранилищ данных на базе СУБД MS SQL Server проектируются как пайплайны (pipeline) в IDE Visual Studio со специальной надстройкой, позволяющей создавать проекты ETL. Эта надстройка называется Службы SQL Server Integration Services (SSIS). SSIS это набор инструментов для интеграции данных корпоративного уровня и преобразований данных при работе с корпоративным хранилищами. Перечислим основные функции служб SSIS:

- копирование или скачивание файлов;
- загрузка данных в хранилища данных;
- очистка и шахта данных;
- управление данными SQL Server.

Ключевые компоненты служб SSIS это:

- набор встроенных задач как для источников данных, так и для самих потоков данных;
- графическая нотация для создания ETL процедур (пакетов);
- специальный репозиторий для хранения, запуска и мониторинга созданных ETL процедур.

Службы SSIS являются дополнительным компонентом MS SQL Server версии не ниже Developer (в версии Express они не работают) и могут быть установлены или при первичной установке сервера баз данных, или установлены уже на развернутый экземпляр, как дополнение.

Для установки и настройки экземпляра СУБД MS SQL Server с SSIS выполним следующие действия.

1. Необходимо загрузить установочный файл MS SQL Server Developer, перейдя по ссылке: <https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads>.

2. В мастере установки SQL Server следует выбрать Новая установка изолированного экземпляра SQL Server или добавление компонентов к существующей установке. Чтобы установить службы Integration Services, следует в разделе Общие компоненты выбрать Integration Services (рис. 35).

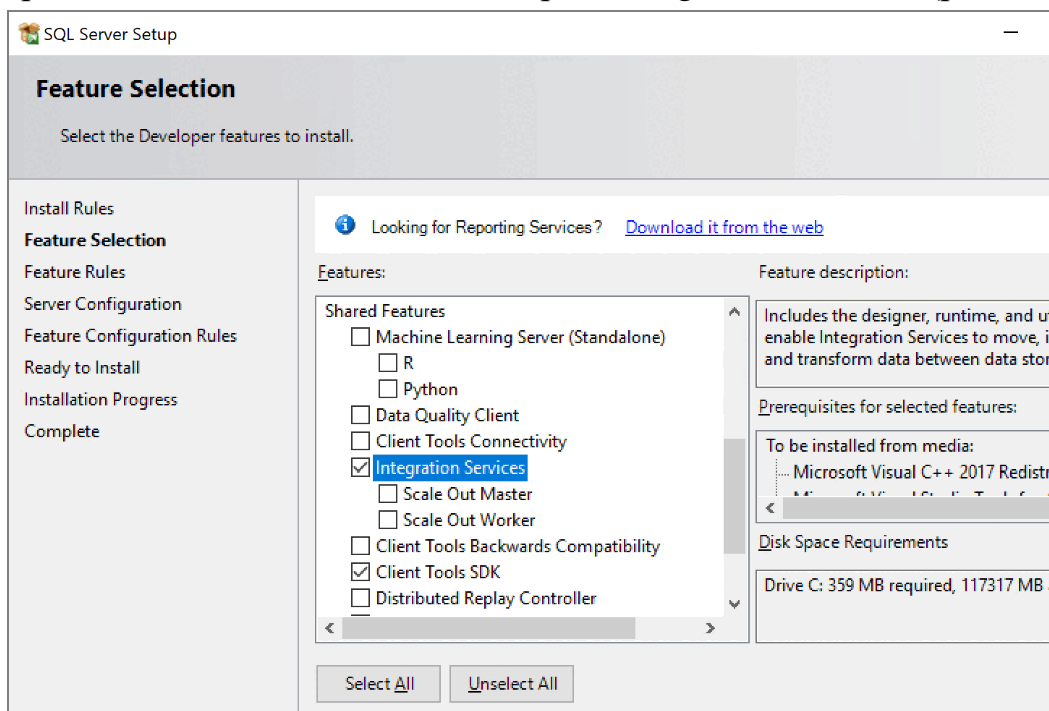


Рисунок 35. Установка экземпляра базы данных с SSIS

После успешной установки сервера перейдем к установке и настройке другого компонента, необходимого для работы. Как было отмечено выше, сама процедура использования графических нотаций для создания ETL процедур будет осуществляться в виде проекта IDE Visual Studio. Далее, для визуализации процесса работы с ПО будет использовано ПО Visual Studio 2019. Для работы с графическими нотациями ETL в установленную IDE следует установить пакет SSDT.

SQL Server Data Tools (SSDT) - это набор средств разработки для создания баз и хранилищ данных SQL Server или Azure, а также пакетов служб Integration Services (IS). Возможности проектов SQL расширяются до конвейеров CI/CD, что позволяет автоматизировать сборку и развертывание проектов базы данных с помощью инструментария непрерывного развертывания (рис. 36).

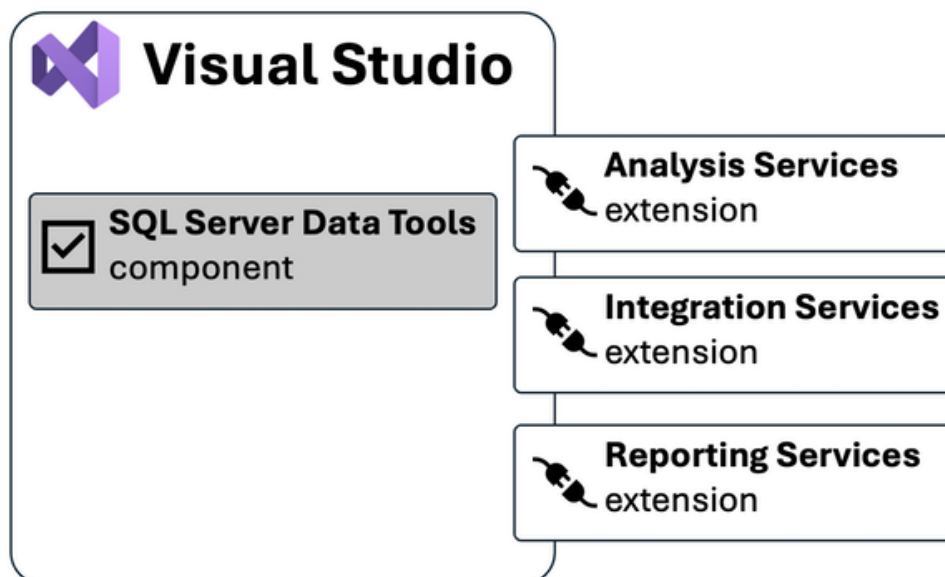


Рисунок 36. IDE Visual Studio и компонент SSDT

Для установки Visual Studio 2019 или более актуальной версии следует перейти по ссылке: <https://learn.microsoft.com/ru-ru/visualstudio/install/install-visual-studio?preserve-view=true&view=vs-2019>

SSDT в установленную IDE добавляется как дополнительный компонент. Для этого следует выполнить следующие действия:

1. Необходимо снова запустить установщик Visual Studio (Visual Studio Installer).
2. В установщике следует выбрать установленную версию Visual Studio и выбрать Изменить.
3. В разделе Хранение и обработка данных следует выбрать компонент SQL Server Data Tools (рис. 37).

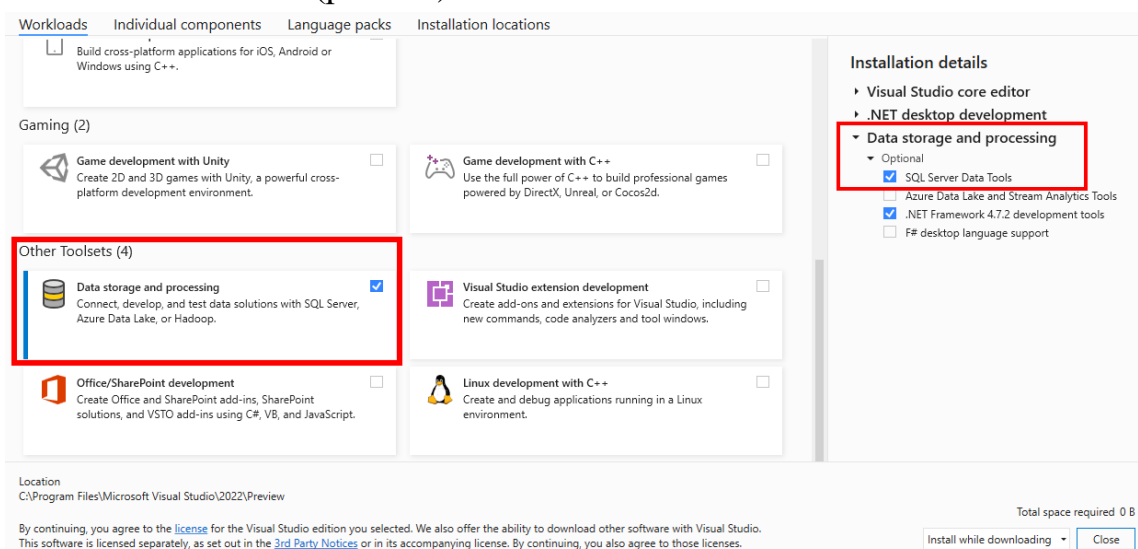


Рисунок 37. Установка компонента SSDT в IDE Visual Studio

После успешной установки и настройки всех необходимых сервисов, перейдем к проектированию ETL-процедур.

6.2. Реализация примера ETL-процедур типа «Поток управления-Поток данных»

В случае успешной установки, в IDE Visual Studio появится возможность создавать проекты ETL-процедур, а затем интегрировать их в имеющийся экземпляр хранилища данных.

Логика работы ETL-процедур в целом схожа с простыми операциями экспорта/импорта данных, рассмотренными в прошлой главе. Разница в большей комплексности решаемых задач и более широком инструментарии для их решения. Далее, в качестве примера, создадим небольшой проект с ETL-процедурой, используя развернутое программное обеспечение.

Предположим, что, в соответствии с определенной бизнес-логикой, в компании предпринимаются действия по архивированию устаревших файлов. С точки зрения выполняемой процедуры логика следующая. Файлы хранятся на сервере, в папке Input. После выполнения ETL-процедуры все файлы из папки Input индексируются и переносятся в папку Archive для завершения процесса архивирования (рис. 38).

имя	^	р....	размер	и
..			DIR	28
01_Input			DIR	27
02_Archive			DIR	27

Рисунок 38. Папки с файлами для ETL-проекта

Внутри папки Input находятся файлы формата *.txt с данными для тестирования работы созданной процедуры ETL (рис. 39).

имя	^	р....	размер	изменено	вид
..			DIR	27.10.2024, 18:34	папка
CustomerInformation_01		txt	821 КБ	17.03.2012, 15:06	текст
CustomerInformation_02		txt	872 КБ	17.03.2012, 15:06	текст
CustomerInformation_03		txt	997 КБ	17.03.2012, 15:06	текст

Рисунок 39. Файлы с данными, подготовленными для переноса

Для начала, откроем Visual Studio и создадим проект Integration Services Project (он, как правило, находится внизу списка шаблонов проектов IDE), рис. 40. Название проекта можно определить самому, в данном примере будет использовано FillStageTables.

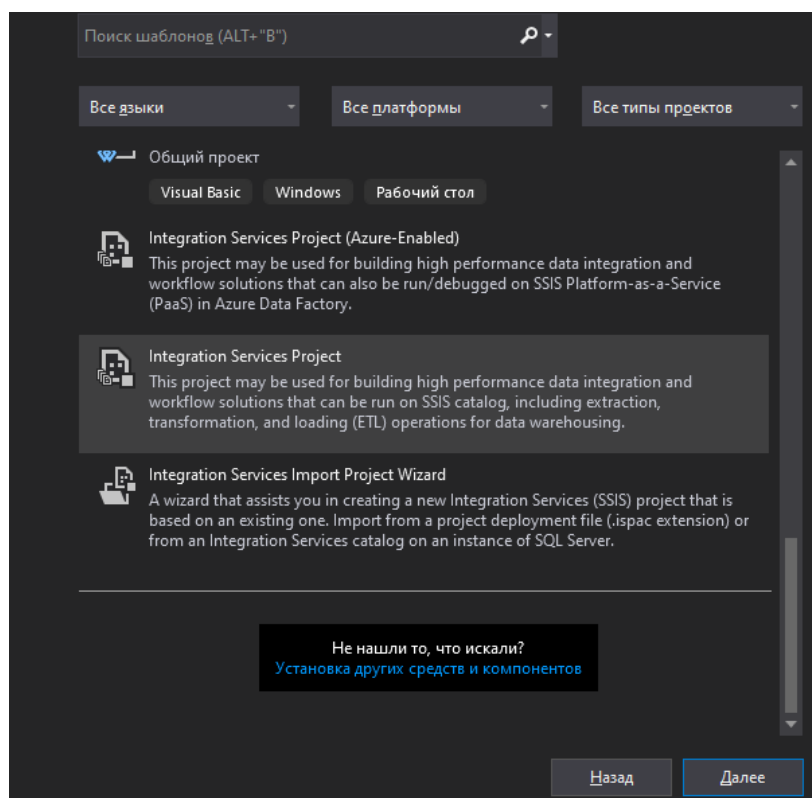


Рисунок 40. Создание ETL-проекта в Visual Studio

После создания проекта ознакомимся с рабочим инструментарием Visual Studio для проектов SSDT. Рабочая область состоит из трех основных частей, слева-направо: SSIS Toolbox, Конструктор и Обозреватель объектов со Свойствами объектов.

SSIS Toolbox позволяет выбрать и затем перенести в поле конструктора один из инструментов, необходимых для создания ETL-процедуры. Поскольку базовый принцип нотаций SSIS – минимальное использования кода, большинство инструментов представляет собой мастер с большим количеством настроек, что обеспечивает удобную работу в low-code среде.

Набор инструментов в SSIS Toolbox будет зависеть от флажка, установленного в Конструкторе. Это может быть набор инструментов для непосредственно управления ETL-процедурами (флаг Control Flow в Конструкторе, рис. 41), или набор инструментов для управления потоками данных внутри ETL-процедуры (флаг Data Flow в конструкторе, рис. 42). При этом отметим, что наборы инструментов отличаются друг от друга и инструменты Data Flow, как правило, являются частями более крупных элементов Control Flow.

Более подробно все доступные инструменты приведенных групп управления студенту предлагается изучить самостоятельно.

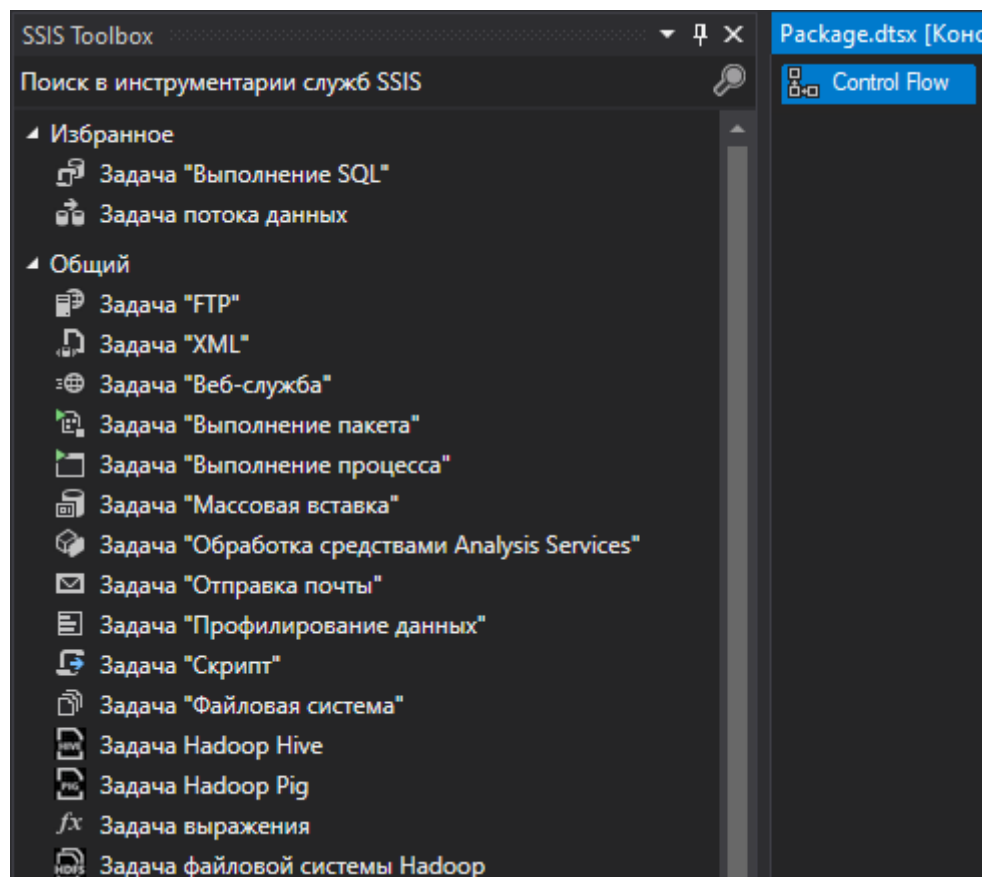


Рисунок 41. Инструменты группы Control Flow (фрагмент)

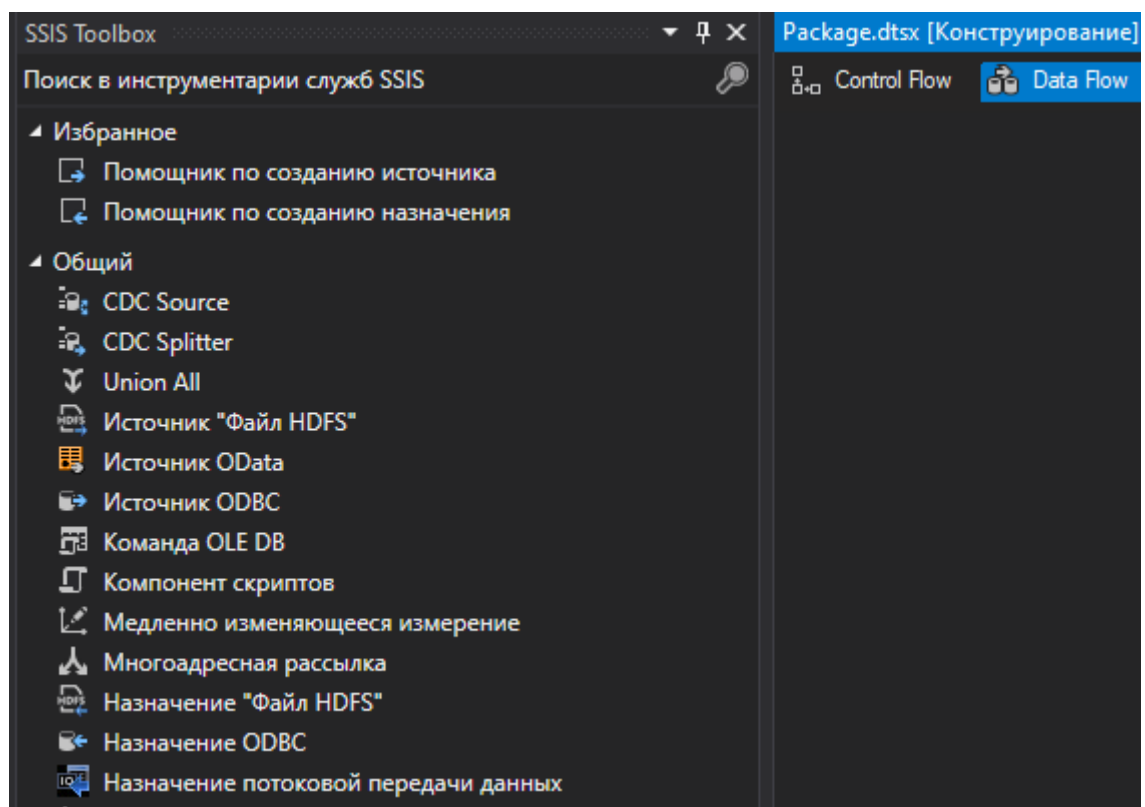


Рисунок 42. Инструменты группы Data Flow (фрагмент)

Для начала разберемся с соединениями с источниками данных, которые будут предоставлять или получать данные в результате работы ETL-процедуры.

Поскольку в данном примере обработке подвергаются файлы формата *.txt, в качестве источника данных будет использоваться «Плоские файлы». Попробуем подключиться к такому источнику, используя средства SSIS Toolbox. Для выбора и настройки источника данных следует воспользоваться инструментом Connection Managers, находящимся в нижней части среднего окна Конструктор. Следует нажать правой кнопкой мыши в указанном интерфейсом месте и в выпадающем меню выбрать мастер создания подключения «Плоские файлы» (New Flat File Connection..., рис. 43).

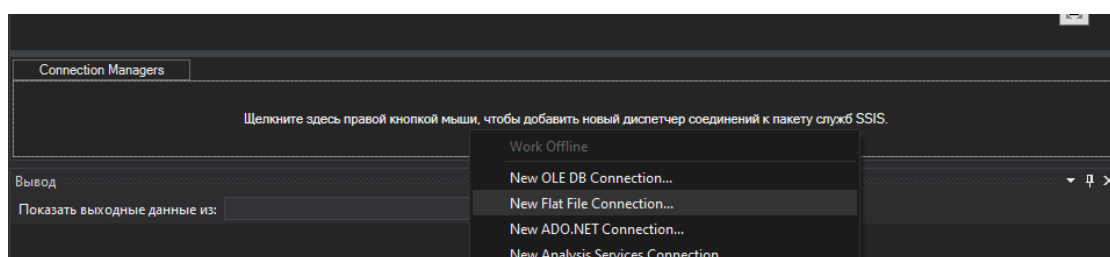


Рисунок 43. Создание нового подключения «Плоские файлы»

Изучим окно настроек мастера подключения. Ключевые элементы, подлежащие настройке и параметризации в ходе создания ETL-процедур, это вкладки «Общие» и «Столбцы».

Во вкладке «Общие» (рис. 44) указываются параметры соединения с выбранным источником данных. Обратите внимание, что Visual Studio и пакет SSDT использует мастер, аналогичный рассмотренному в главе 5 Мастеру экспорта и импорта данных, что существенно упрощает процесс ознакомления с этим инструментом. Выберите в строке Имя файла через кнопку Обзор... файл CustomerInformation.txt из комплекта файлов для лабораторной работы. Остальные параметры оставьте аналогичными рис. 44.

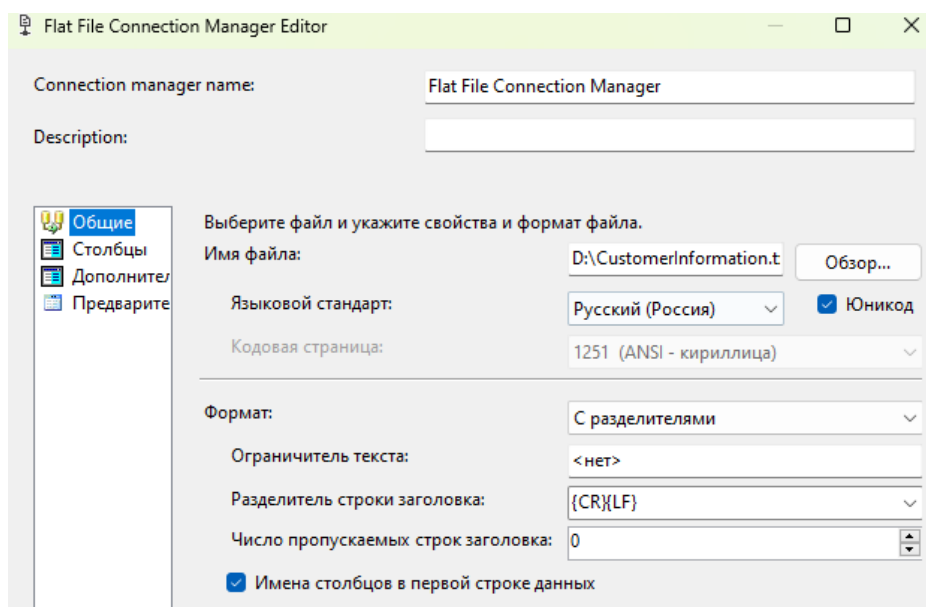


Рисунок 44. Мастер соединения с «плоскими файлами». Вкладка «Общие»

Также как и в случае Мастера экспорта и импорта, вкладка «Столбцы» позволяет отформатировать получаемый поток данных и применить маппинги (рис. 45). Убедитесь в том, что программа успешно «подтянула» данные из плоского файла.

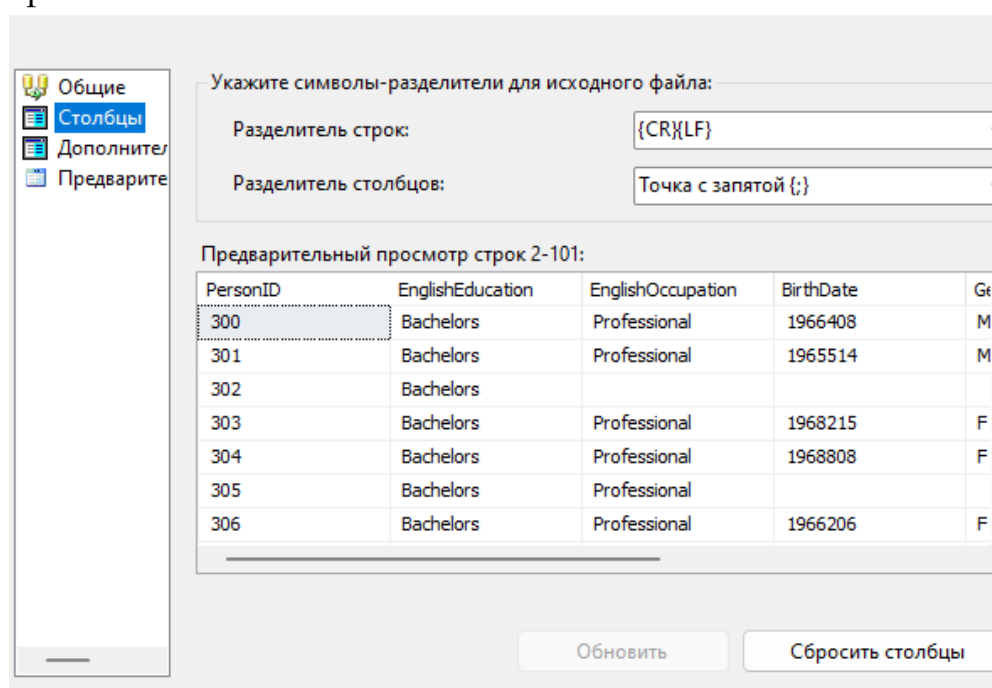


Рисунок 45. Мастер соединения с «плоскими файлами». Вкладка «Столбцы»

Работа с другими источниками данных также в целом не отличается от настройки их в Мастере экспорта и импорта данных MS SQL Server, речь о котором шла в главе 5 этой книги.

Создадим пайплайн для запланированного ETL-процесса. Основой пайплайна станет элемент из группы Control Flow Контейнер «Цикл по каждому элементу» (рис. 46). Для этого, зажав кнопку мыши вытащим его в поле

Конструктора. Этот контейнер будет осуществлять итерации, запланированные внутри до наступления события (например, End Of File, или, в нашем случае – закончатся файлы в папке на сервере).

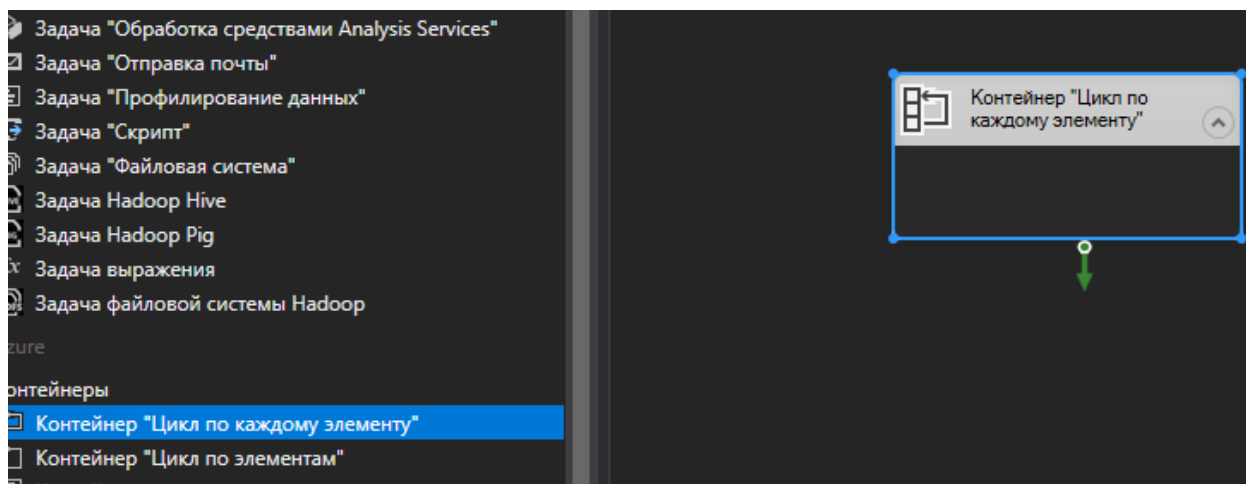


Рисунок 46. Создание контейнера для ETL-процедуры

Нажмем правой клавишей мыши на контейнер и перейдем в его свойства. Вкладка General содержит общую информацию и в нашей конфигурации не участвует. Во вкладке Collection, в поле Enumerator (перечислитель) следует выбрать подходящую для задачи функцию (в нашем случае это цикл по каждому файлу в папке). Далее следует указать на папку с файлами для переноса (Input) и на формат файлов, подлежащих переносу. Важно – не совершить грамматическую ошибку в названии! При обработке имени файла в папке будет использоваться Полное имя файла. Структура задания не подразумевает обработку подпапок, поэтому эту галочку следует снять (рис. 47).

General
Collection
Variable Mappings
Выражения

▼ Foreach Loop Editor

Enumerator
Перечислитель с циклом по каждому файлу

Expressions

Enumerator
Specifies the enumerator type.

Enumerator configuration

Папка:
D:\01_Input

Файлы:
CustomerInformation_*.txt

Получить имя файла

☐ Имя и расширение ☒ Полное ☐ Только имя

☐ Обработать подпапки

Рисунок 47. Настройки «Коллекции» контейнера с циклом

Настроенный цикл, «пробежавшись» по всем файлам в папке вернет полное имя каждого найденного в папке файла. Для того, чтобы эту информацию сохранить и иметь возможность использовать в дальнейшем, создадим переменную, куда это положим. Для создания переменной следует перейти во вкладку Variable Mappings (маппинги переменной). Далее следует создать новую переменную (New variable) и указать названия контейнера, где она будет использоваться (в нашем случае она будет использоваться только в контейнере цикла) ее имя и тип данных (рис. 48).

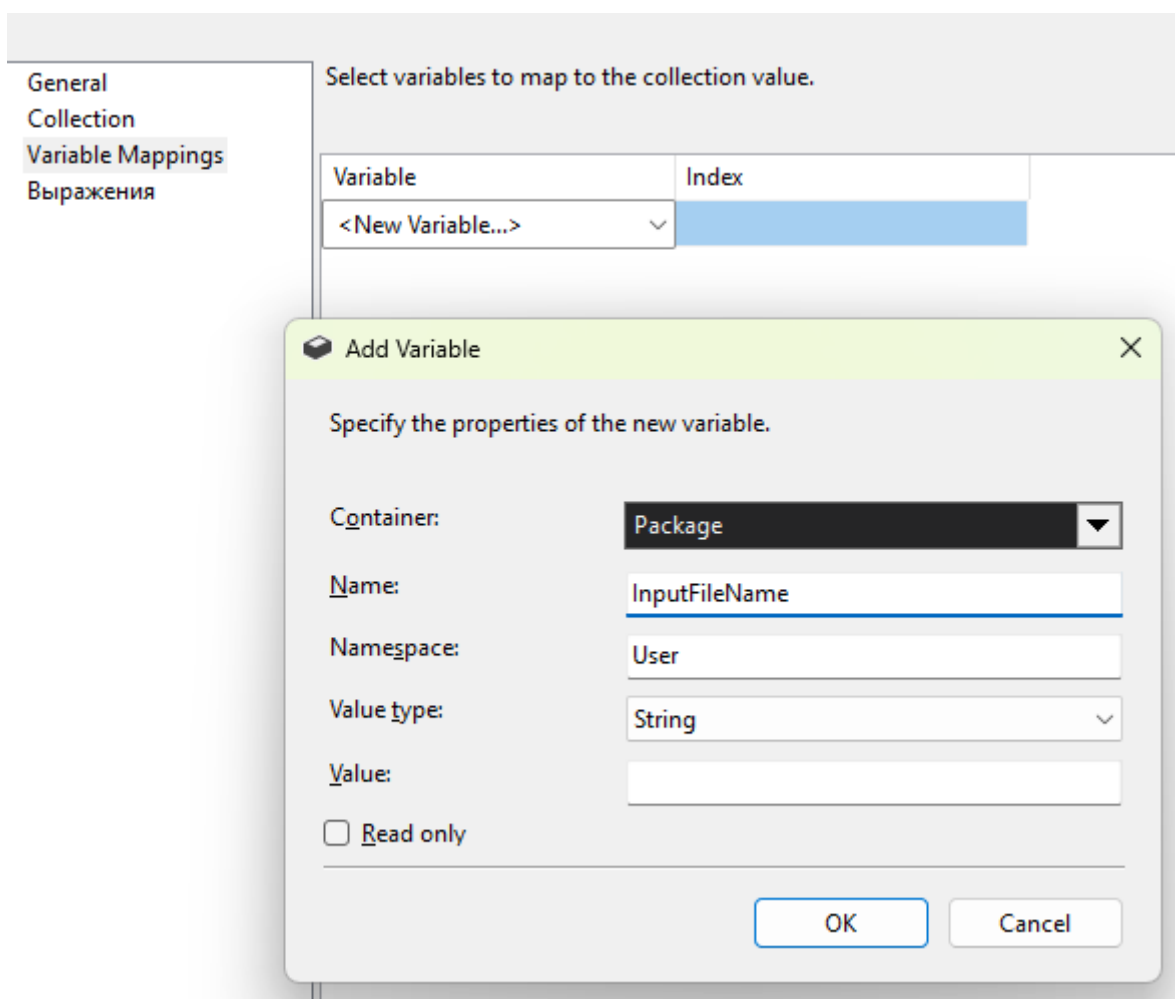


Рисунок 48. Создание переменной для цикла обработки

Результатом создания станет переменная и ее значение (пока цикл не запускался, ее значение Index, очевидно, 0), рис. 49.

Collection	
Variable Mappings	
Выражения	

Variable	Index
User::InputFileName	0

Рисунок 49. Результат создания переменной для цикла обработки

Созданная переменная будет по очереди передавать соединению «Плоские файлы» имя очередного файла из папки, который должен быть перемещен в рамках заданного цикла. Самое время передать созданному ранее соединению с «Плоским файлом» информацию о созданной в цикле переменной. Для этого, нажав правой клавишей мыши на кнопке Flat File Connection Manager, находящегося во вкладке Connection Managers, откроем его свойства (рис. 50).

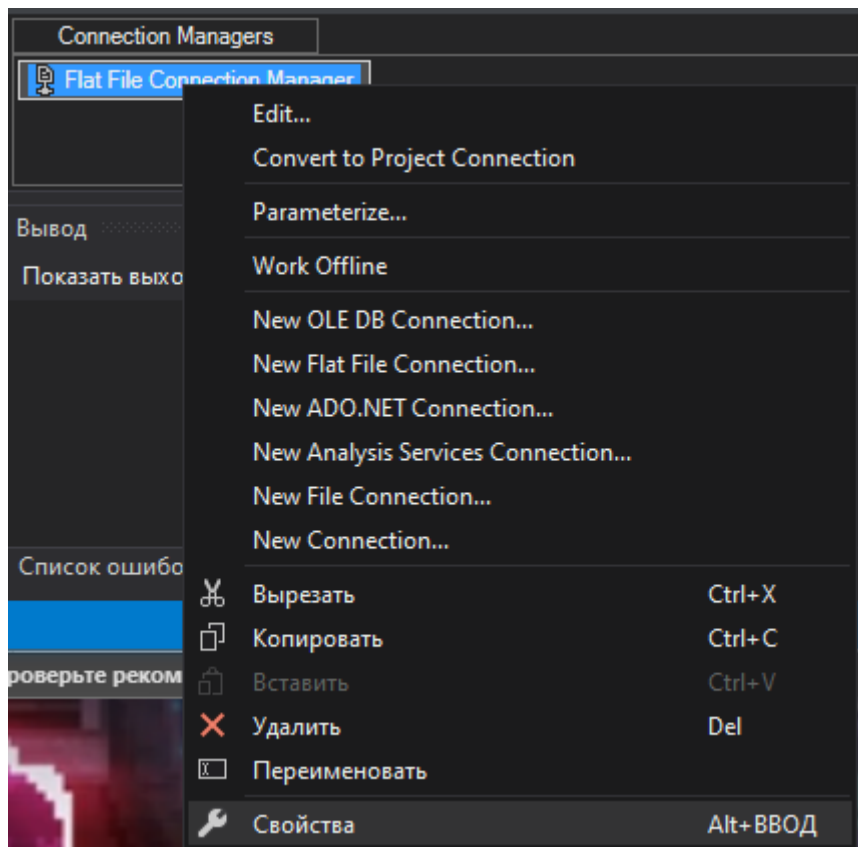


Рисунок 50. Свойства соединения Flat File Connection Manager

В свойствах нужно найти параметр Expressions (Выражения) и нажать напротив него на кнопку с тремя точками, рис. 51.

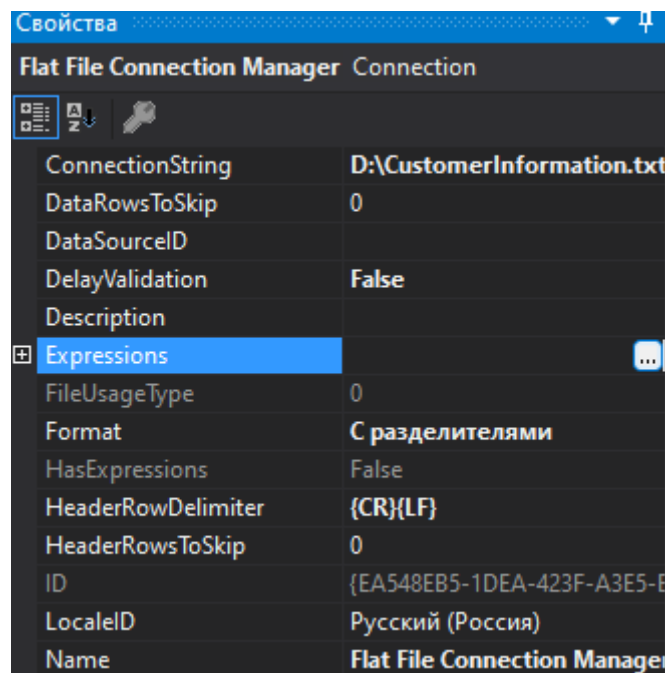


Рисунок 51. Изменение свойства Expressions для Flat File Connection Manager

Скопируйте выражение для нового свойства ConnectionString с рис. 52.

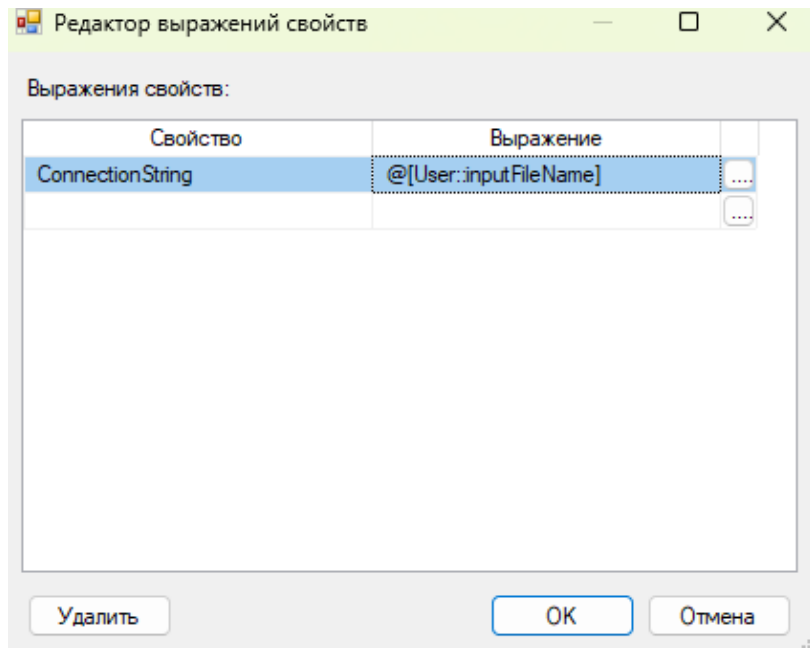


Рисунок 52. Новое выражение для Flat File Connection Manager

Это выражение присваивает в каждой итерации контейнера Foreach Loop Container значение ранее созданной переменной inputFileName строке соединения, динамически настраивая диспетчер на соединение каждый раз с новым файлом из целевой папки. Осталось запустить подготовленный поток данных (в нашем случае - файлов) в архивную папку. Для этого, внутри цикла нам понадобятся два последовательно исполняемых элемента – задача потока данных (для формирования потока) и задача «Файловая система», для перемещения потока в новую, архивную папку. Для начала, из SSIS Toolbox под флажком Control Flow, зажав левую клавишу мыши вытащим и поместим внутрь созданного в Конструкторе цикла элемент Задача потока данных (рис. 53).

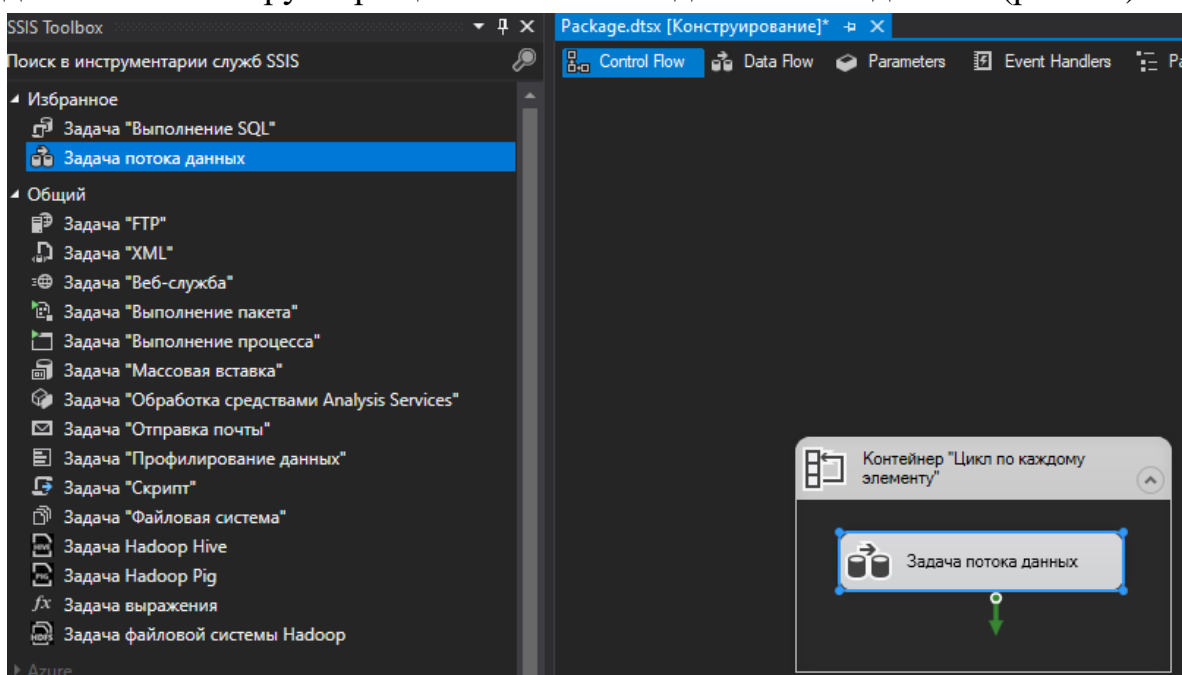


Рисунок 53. Добавление задачи потока данных к циклу

Далее, дважды щелкните кнопкой мыши на добавленной задаче потока данных, чтобы получить доступ к полю редактирования (в конструкторе откроется флажок Data Flow). В качестве потока данных для задачи будет выступать созданный ранее Flat File Connection. Но, поскольку напрямую как поток данных мы его в конструкторе использовать не можем, то сперва следует из SSIS Toolbox в Конструктор левой клавишей мыши вытащить элемент Источник «Неструктурированный файл» (рис. 54), а затем, нажав на него правой клавишей мыши и выбрав его редактирование (Edit) (рис. 55), во вкладке Connection Manager, с помощью выпадающего меню, выбрать ранее созданное Flat File Connection Manager (рис. 56). Таким образом мы настроили первый элемент цикла.

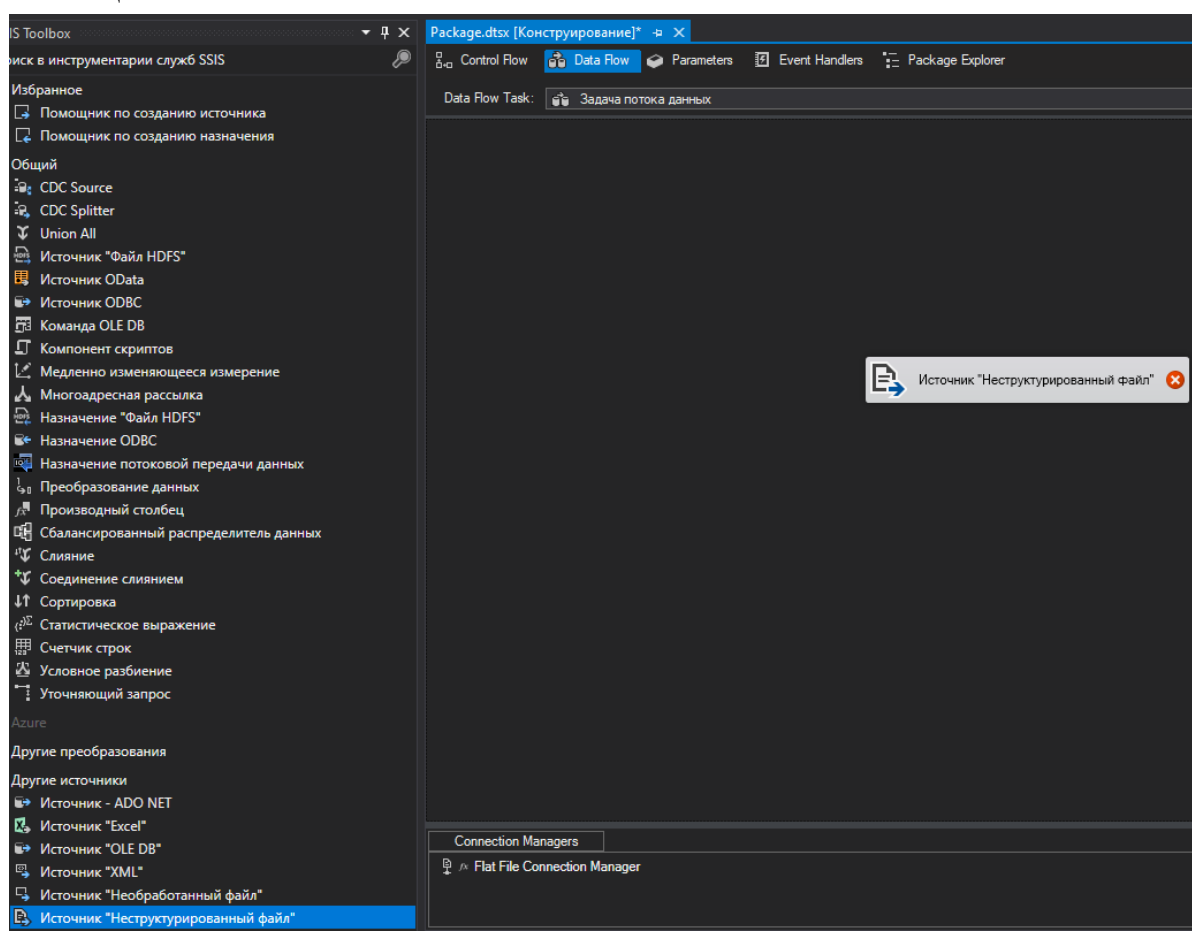


Рисунок 54. Создание Источника «Неструктурированный файл»

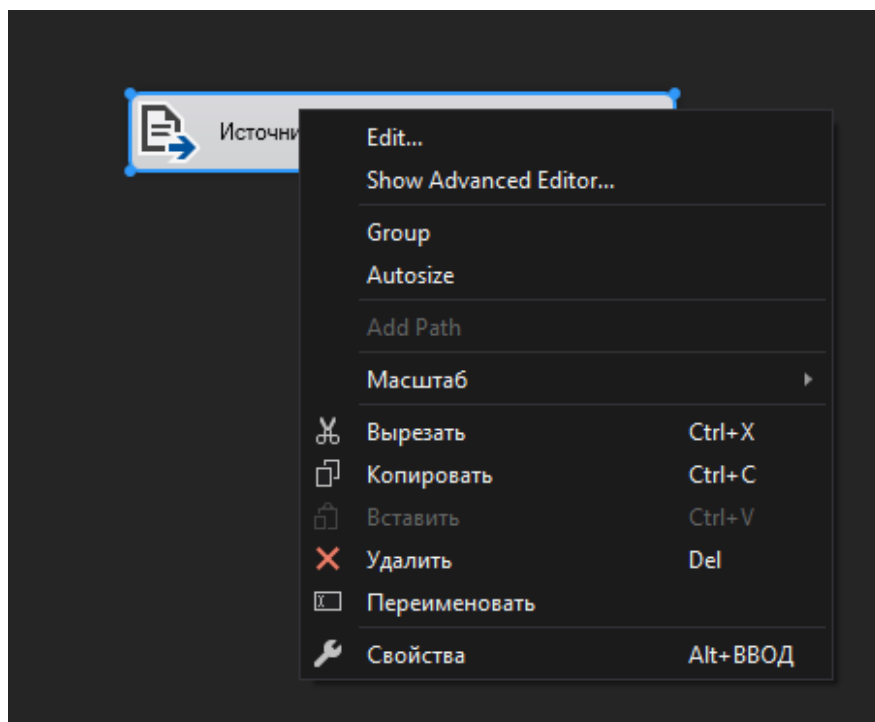


Рисунок 55. Редактирование Источника «Неструктурированный файл»

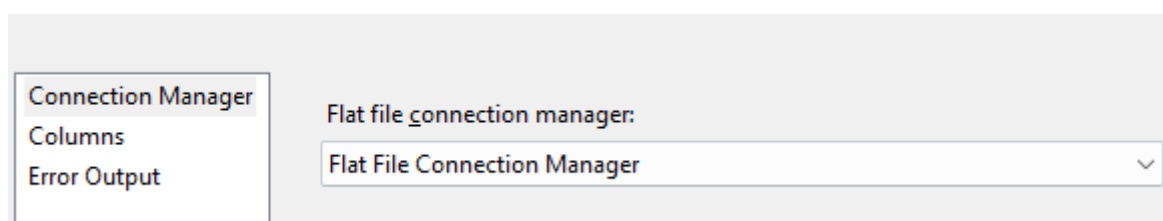


Рисунок 56. Настройка Connection Manager для Источника «Неструктурированный файл»

Вернемся в Конструкторе во вкладку Control Flow и добавим в Контейнер с циклом второй элемент обработки – Задачу «Файловая система» (рис. 57).

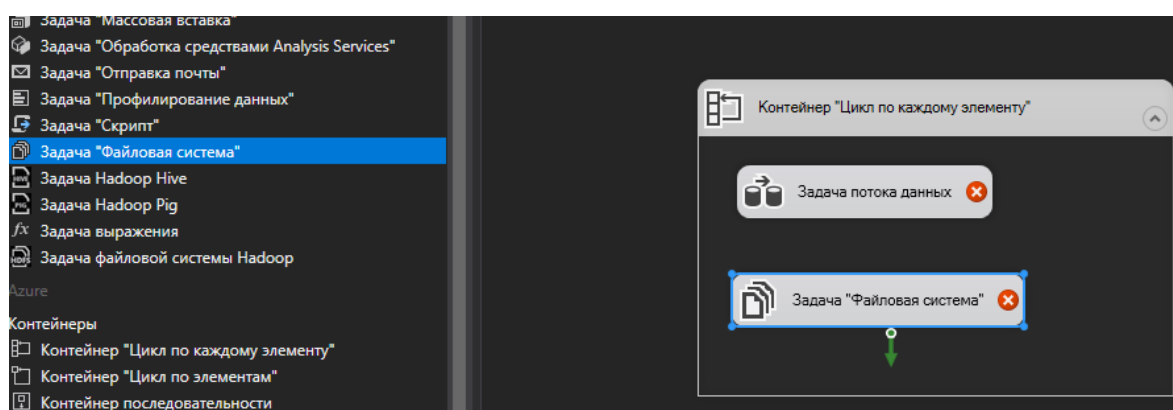


Рисунок 57. Добавление в цикл задачи «Файловая система»

Зажав левую клавишу мыши, сверху вниз следует соединить две созданные внутри цикла задачи в последовательность, как это показано на рис. 58.

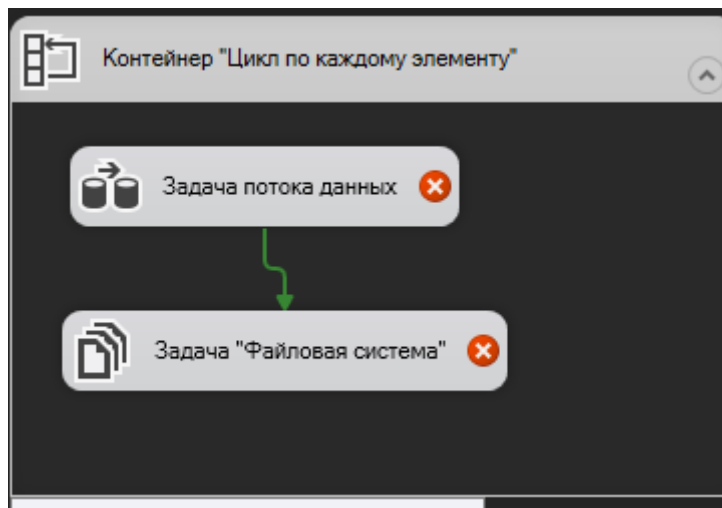


Рисунок 58. Формирование последовательности выполнения задач в цикле

Осталось лишь настроить новую добавленную задачу файловой системы и обработку исключений. Через нажатие правой клавиши мыши откроем Редактор задачи «Файловая система». Изменим некоторые настройки во вкладке «Общие» (рис. 59)

Настроить свойства, необходимые для выполнения операций файловой системы, таких как создание, перемещение или удаление файлов или каталогов.	
Общие	
Name	Archive Input File
Description	Задача "Файловая система"
Операция	
Operation	Переместить файл
Соединение с источником	
IsSourcePathVariable	True
SourceVariable	User::InputFileName
Соединение с назначением	
IsDestinationPathVariable	False
DestinationConnection	
OverwriteDestination	True
Name Определение имени задачи.	

OK Отмена Справка

Рисунок 59. Изменение общих свойств задачи «Файловая система»

Свойство Operation следует изменить на «Переместить файл» (из папки в папку), в SourceVariable следует поместить ранее созданное название переменной, а в свойство DestinationConnection выбрать ранее созданный Flat File Connection Manager (рис. 60).

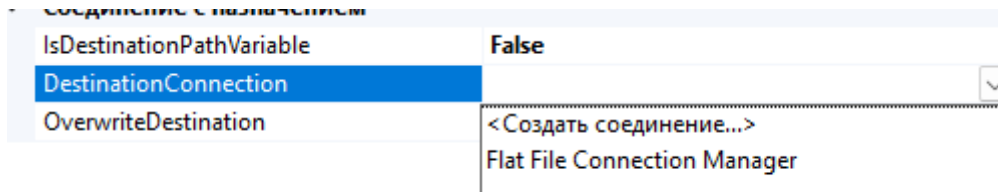


Рисунок 60. Добавление соединения до целевой папки

В конфигурации добавленного соединения в строке Usage type (используемый тип) следует выбрать Existing Folder (созданная папка), а в строке Folder (папка), указать физический путь к целевой папке, куда файлы должны быть перемещены в ходе исполнения цикла (Archive), рис. 61.

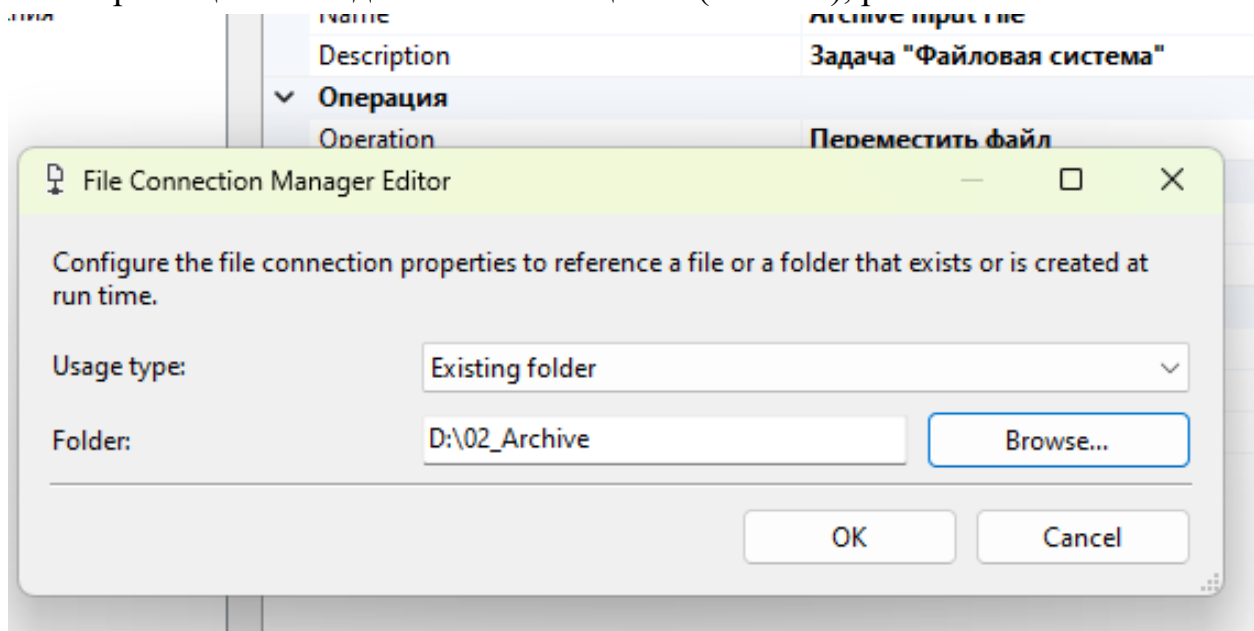


Рисунок 61. Настройки File Connection Manager Editor

Ранее, мы настроили задачу «Файловая система» так, чтобы свойству ConnectionString значение присваивалось динамически. В ходе исполнения цикла это должно привести к ошибке: задача не сможет проверить правильность подключения к файлу, потому что переменной inputFileNames не было присвоено никакое значение. Для того, чтобы эта ошибка не помешала компиляции полученной ETL-процедуры, в свойствах объекта Archive Input File следует изменить свойство DelayValidation (пропустить валидацию) на True (рис. 62).

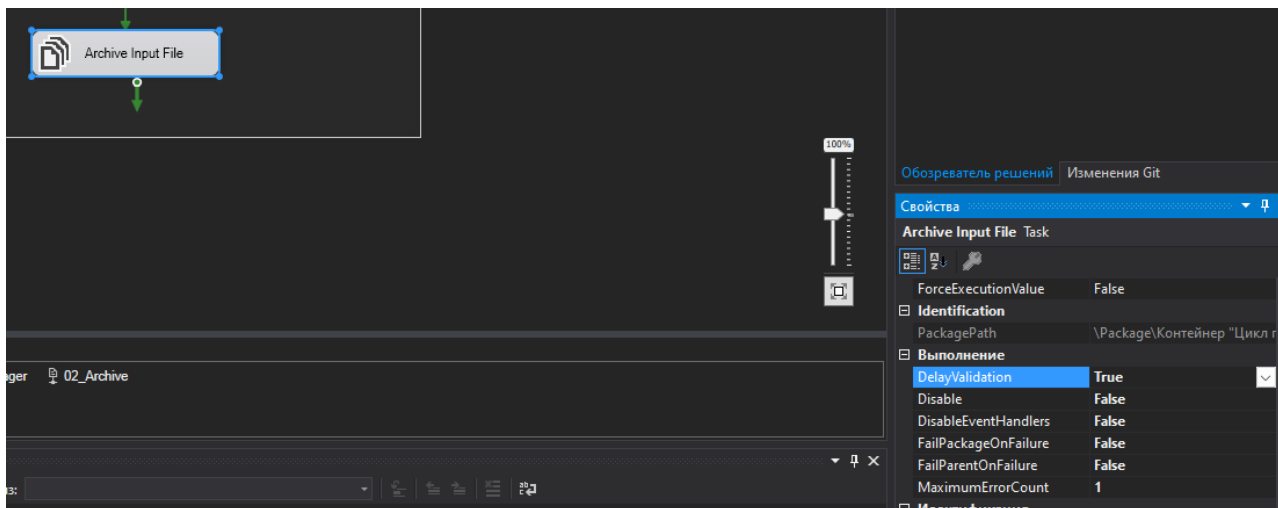


Рисунок 62. Отмена валидации результатов цикла

Кнопкой с зеленой стрелкой запустим ETL-процедуру и убедимся в том, что она работает как запланировано (папка Input в итоге пустая, а все файлы были перемещены в папку Archive), рис. 63.

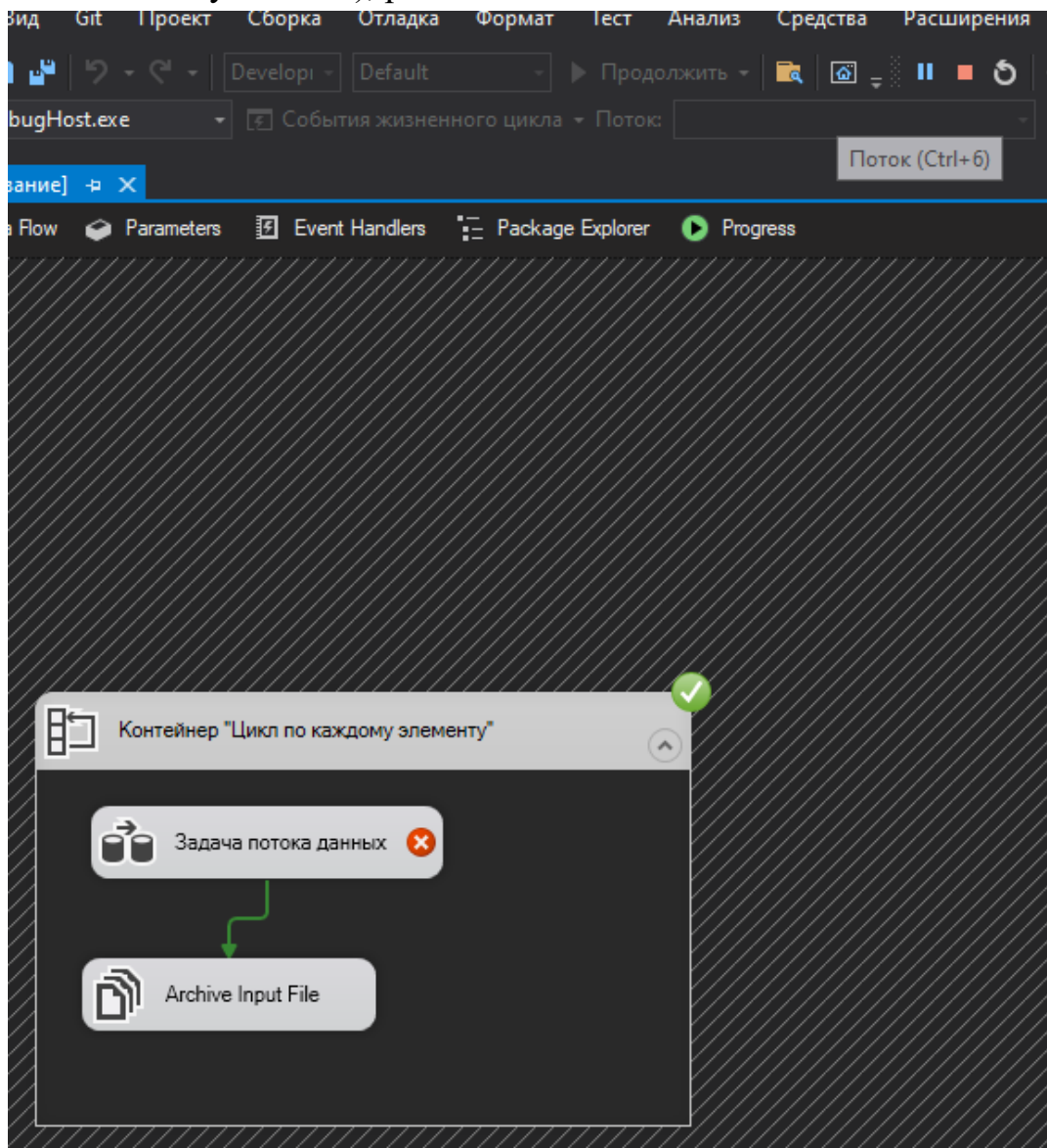


Рисунок 63. Запуск ETL-процедуры на исполнение, отладка

Остановите кнопкой с красным квадратом процесс отладки и сохраните проект. Таким образом, нами был создан ETL-пайплайн для работы с файлами на сервере.

6.3. Контрольные задания по теме

Самостоятельно создайте ETL пайплайн средствами MS Visual Studio, экспортирующий данные одной из таблиц одной базы данных на сервере в другую. При этом следует для целевой таблицы переименовать все столбцы, оставив структуру данных при этом неизменной.

Проверьте работу разработанного пайплайна с маппингами. В случае успешной реализации добавьте в отчет скриншот с настройками подключения источников, а также с графической моделью самого разработанного пайплайна.

7. ИМПОРТ И ОБРАБОТКА ДАННЫХ В NOSQL ХРАНИЛИЩАХ ДАННЫХ НА ПРИМЕРЕ MONGODB

7.1. Работа с СУБД MongoDB в режиме подключения SSH. Создание базовых элементов документного хранилища данных

Практика работы с noSQL хранилищами данных инструментально несколько отличается от работы с реляционными хранилищами. Связано это, в первую очередь с тем, что для взаимодействия с СУБД будут использованы другие языки программирования. Что касается программного обеспечения, облегчающего и автоматизирующего процессы импорта и экспорта данных, то предпочтение тут уделяется универсальным программным продуктам, способным обеспечивать работу администратора данных в условиях большого многообразия типов noSQL СУБД.

В данной главе будут рассмотрены базовые основы взаимодействия с документным noSQL хранилищем MongoDB, являющимся на данный момент наиболее популярным в использовании представителем класса noSQL хранилищ. В рамках практикума будут изучены основы создания, обработки и импорта документов в СУБД.

Для работы в рамках практикума понадобится развернуть и настроить ряд программных продуктов. Основой будет являться ядро СУБД MongoDB, развернутое в виртуальной машине Linux. Виртуальную машину размером приблизительно 4 Гб можно скачать по следующему адресу: <https://files.sberdisk.ru/s/6NOnSWZPKkGvtso>. Для развертывания виртуальной машины на компьютерах с операционной системой MS Windows необходимо установить актуальную версию ПО Oracle VirtualBox. Видео по настройке и работе с виртуальной машиной на компьютерах с операционной системой MacOS доступно по ссылке: <https://www.youtube.com/watch?v=qvtr34SKkBw>

Также, для дальнейшей работы в рамках практикума понадобится установка клиента PuTTY (<https://www.putty.org>), и нативного клиента для работы с СУБД MongoDB, - MongoDB Compass.

Разберемся с процедурой работы с консолью СУБД MongoDB и с созданием элементов хранения этого типа хранилища данных.

Следуя указаниям выше, запустите виртуальную машину с экземпляром MongoDB, введя в консоли Linux следующие реквизиты пользователя: логин db_user, пароль dbStudy. Из псевдографического интерфейса запущенной виртуальной машины следует запустить сервис СУБД MongoDB.

Следующим действием нужно выяснить выданный виртуальной машиной

IP адрес для того, чтобы иметь возможность подключиться к сервису СУБД из-за пределов виртуальной машины. Для этого следует в командной строке Linux на виртуальной машине ввести команду `ip addr show`. Результат выполнения команды, в виде требуемого IP адреса и порта подключения показан на рис. 64.

```
db_user@dbserver:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:59:58:52 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic enp0s3
        valid_lft 476sec preferred_lft 476sec
    inet6 fe80::a00:27ff:fe59:5852/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:a6:5d:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic enp0s8
        valid_lft 476sec preferred_lft 476sec
    inet6 fe80::a00:27ff:fea6:5d70/64 scope link
        valid_lft forever preferred_lft forever
```

Рисунок 64. Получение IP-адреса и порта виртуальной машины Linux

Далее следует вернуться в «домашнюю» операционную систему и запустить клиент PuTTY. Благодаря этому клиенту, зная IP-адрес нужного сервиса можно будет подключиться к СУБД MongoDB через протокол безопасного соединения SSH. Настройки сессии соединения (не забудьте указать полученные ранее IP-адрес и порт в соответствующих элементах соединения) показаны на рис. 65.

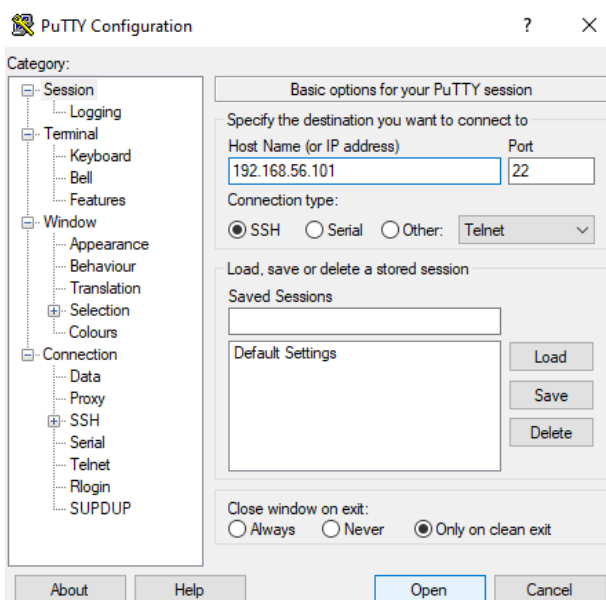


Рисунок 65. Настройки клиента PuTTY для подключения к сервису СУБД

После нажатия кнопки **Open**, в случае указания правильных атрибутов соединения, будет установлено соединение с сервисом СУБД и через консоль можно будет отправлять команды уже непосредственно этому сервису.

Для начала проверим, есть ли какие-нибудь действующие хранилища на данном экземпляре документного хранилища. Для этого следует исполнить команду `show dbs`.

Процесс создания новых хранилищ данных с включенными в них элементами, такими как коллекции и документы весьма прост. Связано это с фактическим отсутствием схем данных, обусловленных свойствами `noSQL` модели хранения. Так, для создания нового хранилища `test` с коллекцией `collection1` и документом, содержащим значение (в нашем примере «Соколов») достаточно выполнить последовательность из двух команд: `use test` (консоль переключится на новое хранилище данных) и `db.collection1.insert ({name: “Соколов”})`. Последняя команда создаст коллекцию и сразу же впишет в нее новый документ. Для удаления созданной коллекции используем команду `db.collection1.drop ()`. Обратите внимание – вместе с единственной коллекцией будет удалено и само хранилище данных.

7.2. Экспорт данных в хранилище MongoDB

Существует несколько способов импорта и экспорта данных в документное хранилище с использованием консольных команд или оболочек с графическим интерфейсом. Наиболее распространенный формат передачи файлов в документной `noSQL` парадигме, — это файлы формата `JSON` или файл формата `*.csv`. В данной практике будет использована оболочка из комплекта программного обеспечения `MongoDB Compass`. Ее следует предварительно установить на «домашнюю» операционную систему компьютера.

Для начала работы требуется включить сервис `MongoDB` в виртуальной машине аналогично прошлому практикуму (см. п. 7.1). В оболочке управления СУБД `Compass` через соединение `SSH Tunnel`, указав в качестве IP адреса и порта полученные ранее данные (см. п. 7.1), а в качестве реквизитов пользователя логин `db_user`, пароль `dbStudy` следует подключиться к ядру документного хранилища данных. Вариант подключения показан на рис. 66.

URI ⓘ Edit Connection String ☒

mongodb://localhost:27017/

▼ Advanced Connection Options

General Authentication TLS/SSL **Proxy/SSH Tunnel** Advanced

SSH Tunnel/Proxy Method

None **SSH with Password** SSH with Identity File Socks5

SSH Hostname

192.168.56.101

SSH Port

22

SSH Username

db_user

SSH Password

***** Optional

Рисунок 66. Вариант настройки подключения к хранилищу данных MongoDB

Для осуществления процедуры экспорта данных создадим целевое хранилище provider. Данное хранилище это фрагмент BigData компании-провайдера услуг сотовой связи, обеспечивающих пользователи доступом к сотовой связи и сервисам сети Интернет. В структуре данного хранилища следует создать 5 коллекций, в которые впоследствии будет осуществлен экспорт данных в виде документов. Коллекция calls содержит информацию об осуществленных пользователями звонках. Хранилище данных и его первую коллекцию создадим, используя форму Compass Create Database (нажатие правой кнопки мыши на Databases в левой части главной формы Compass), рис. 67.

Create Database

Database Name

provider

Collection Name

calls

☐ Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. [Learn More](#)

☐ Use Custom Collation

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. [Learn More](#)

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

Cancel

Create Database

Рисунок 67. Создание хранилища данных и коллекции в ПО Compass

Далее, через инструмент добавления коллекций в существующее хранилище данных (значок «+» на рис. 68) следует добавить остальные коллекции модели хранилища данных. Коллекция `internet` содержит информацию о сессиях работы в сети Интернет. Коллекция `messages` – информацию о мгновенных сообщениях, переданных пользователями по сотовой сети связи. Коллекция `users` содержит информацию о клиента провайдера, а коллекция `tariffs` – информацию о тарифах сотовой связи, предлагаемых пользователям.

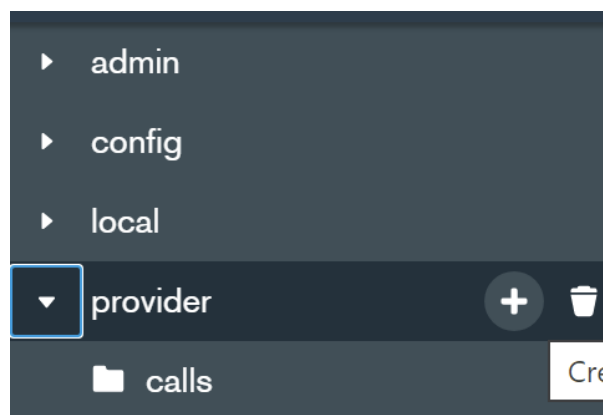


Рисунок 68. Добавление коллекций к хранилищу данных

Далее, последовательно вызывая мастера управления созданными коллекциями, нажимая на них левой кнопкой мыши, добавим соответствующие названиям коллекций документы. Массивы данных для экспорта представлены в архиве compassdata.zip и их можно скачать, перейдя по ссылке: https://msuniversity.ru/uploads/msu_file/file/40315/compassdata.zip. В Мастере следует нажать кнопку Import Data и в открывшейся форме указать тип файла (в нашем случае это *.csv), после чего выбрать конкретный файл для экспорта в ниспадающем меню (рис. 69).

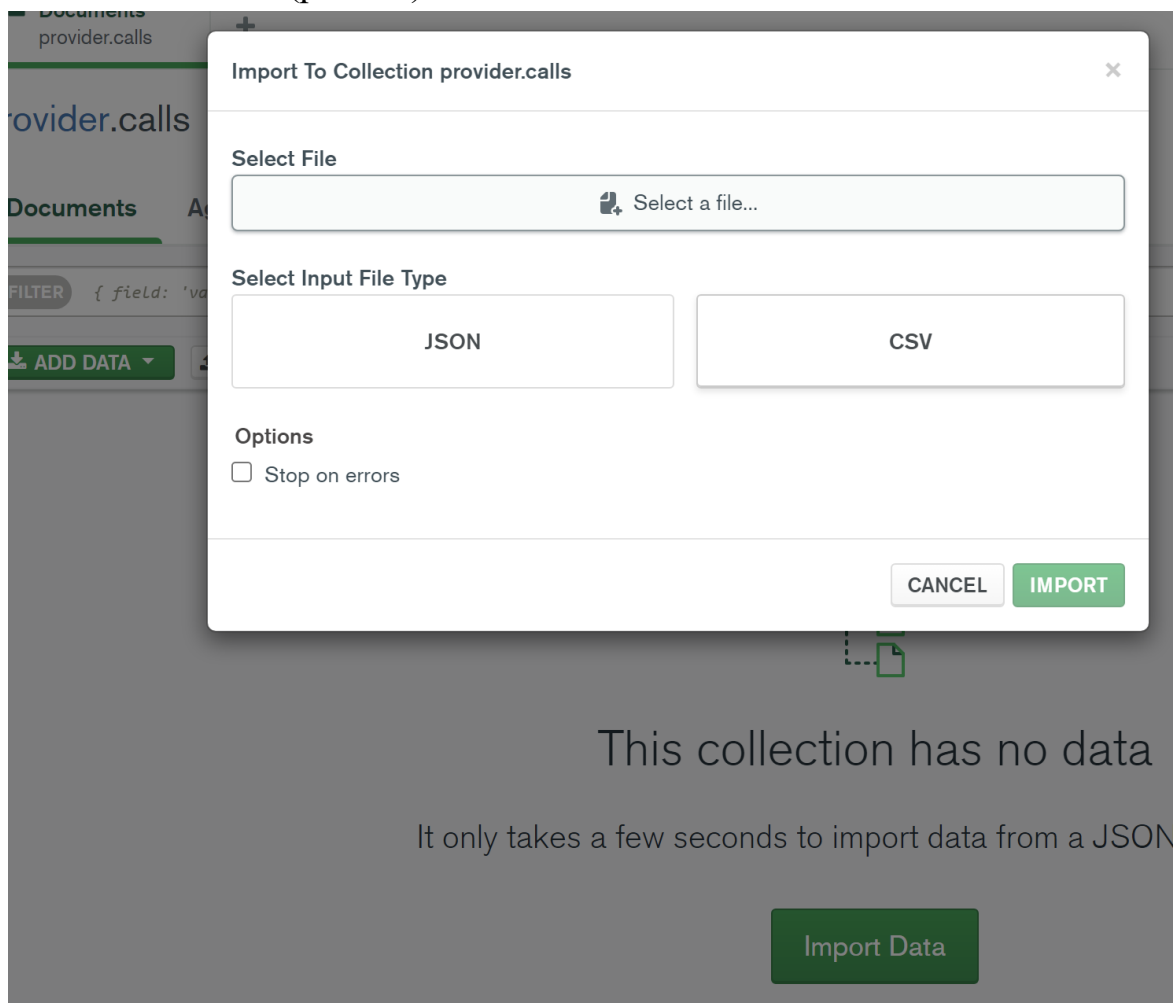


Рисунок 69. Настройка процедуры экспорта данных в хранилище MongoDB

Последовательно заполните данными все созданные ранее коллекции хранилища provider используя мастер экспорта.

7.3. Запросы к данным в хранилище MongoDB

Для написания и исполнения запросов к данным в документном хранилище mongodb удобно использовать командную строку mongosh, встроенную в программное средство Compass, которое использовалось в прошлой практической работе (рис. 70). После открытия командной строки

убедитесь, что подключены к хранилищу provider, выполнив команду `use provider`.

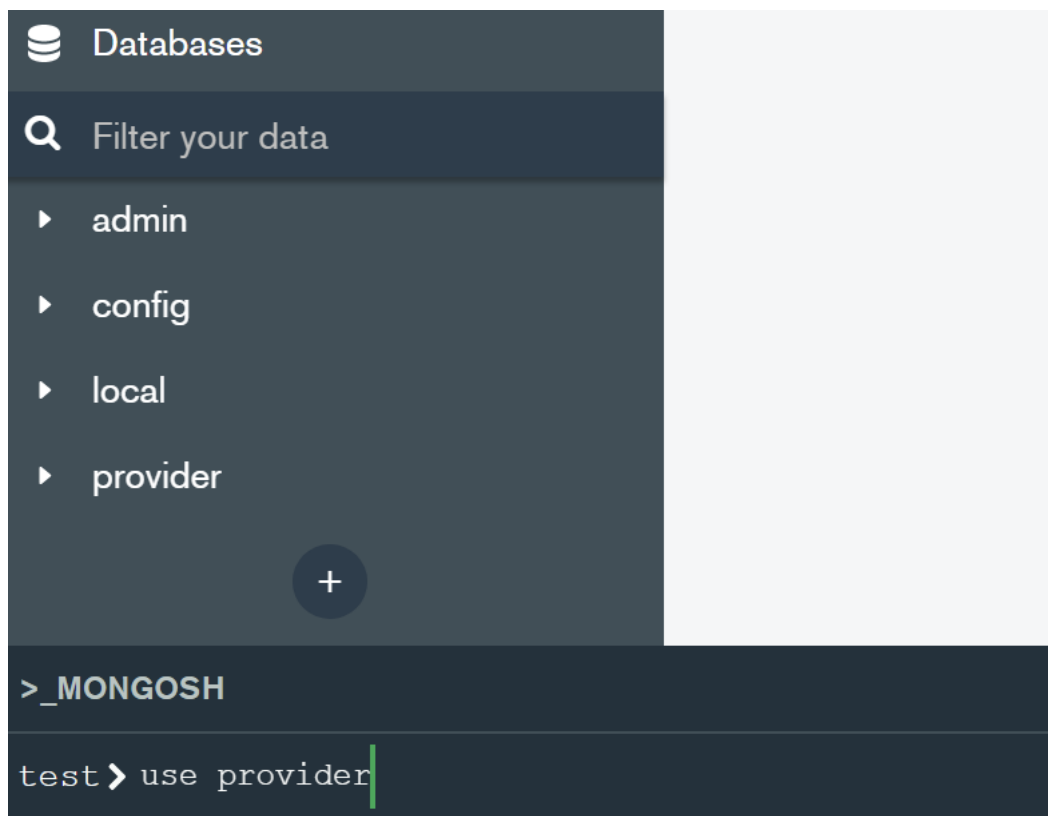


Рисунок 60. Командная консоль запросов к документному хранилищу mongosh

Все последующие запросы будут адресованы ранее созданному (см. п. 7.1 и 7.2) хранилищу provider. Следует обратить особое внимание на то, что, вследствие особенностей функционирования noSQL хранилищ данных прямых запросов на соединение данных нескольких документов не поддерживается, большинство аналитических запросов к документному хранилищу – это запросы с агрегацией и группировкой. Далее будет приведено несколько примеров реализации этих функций в СУБД MongoDB.

Первая аналитическая задача, подлежащая практическому решению: применив функции группировки и агрегатную функцию `count` (определение количества экземпляров) необходимо определить, какое количество сессий интернета пользователи провайдера открывали по дням, за которые были собраны данные. В языке BSON, который использует СУБД MongoDB для группировки значений используется функция `$group` (сгруппируем сессии интернета в документе `internet`), а для осуществления агрегации используется инструкция `aggregate` в связке непосредственно с агрегатной функцией (в нашем случае, чтобы посчитать число экземпляров используем функцию `$count`). Команда запроса показана в листинге 9. Скопируйте ее в командную строку, выполните и оцените полученный результат.

Листинг 9. Скрипт запроса к хранилищу с агрегацией и группировкой

```
db.internet.aggregate ([{$group: {_id: "$session_date",  
countNumberOfDocumentsForState: {$count: {}}}}])
```

В следующем запросе, с использованием группировки, условия ограничения и сортировки выведем выборку из документа users, содержащую все уникальные имена пользователей, отсортированные в возрастающем порядке. Скрипт запроса показан в листинге 10.

Листинг 10. Скрипт запроса к хранилищу с группировкой и сортировкой

```
db.users.aggregate ([{$group: {_id: "$first_name"}}]).sort ({_id: 1})
```

Для изучения агрегатных функций на столбцы документов с арифметическими значимыми числовыми значениями, в текущей базе данных создадим новый документ sales, заполнив его восемью экземплярами (см. листинг 11).

Листинг 11. Скрипт создания документа sales

```
db.sales.insertMany([  
  { "_id" : 1, "item" : "abc", "price" :  
    NumberDecimal("10"), "quantity" : NumberInt("2"), "date"  
    : ISODate("2014-03-01T08:00:00Z") },  
  { "_id" : 2, "item" : "jkl", "price" :  
    NumberDecimal("20"), "quantity" : NumberInt("1"), "date"  
    : ISODate("2014-03-01T09:00:00Z") },  
  { "_id" : 3, "item" : "xyz", "price" :  
    NumberDecimal("5"), "quantity" : NumberInt("10"), "date"  
    : ISODate("2014-03-15T09:00:00Z") },  
  { "_id" : 4, "item" : "xyz", "price" :  
    NumberDecimal("5"), "quantity" : NumberInt("20"),  
    "date" : ISODate("2014-04-04T11:21:39.736Z") },  
  { "_id" : 5, "item" : "abc", "price" :  
    NumberDecimal("10"), "quantity" : NumberInt("10"),  
    "date" : ISODate("2014-04-04T21:23:13.331Z") },
```

```
{  "_id"    : 6,  "item"    : "def",  "price"    :
NumberDecimal("7.5"),  "quantity":  NumberInt("5"  )  ,
"date"    : ISODate("2015-06-04T05:08:13Z") },
{  "_id"    : 7,  "item"    : "def",  "price"    :
NumberDecimal("7.5"),  "quantity":  NumberInt("10")  ,
"date"    : ISODate("2015-09-10T08:43:00Z") },
{  "_id"    : 8,  "item"    : "abc",  "price"    :
NumberDecimal("10"),  "quantity"  :  NumberInt("5"  )  ,
"date"    : ISODate("2016-02-06T20:20:13Z") }
```

Сконструируем для нового документа аналитический запрос посложнее. Сгруппируем предлагаемые покупателям продукты по названию, сосчитаем сумму продаж по каждой полученной группе и выведем в результате суммы только для тех групп, где полученное значение превышает значение 100. В запросе потребуется использовать инструкцию `aggregate`, функцию группировки `$group`, агрегатную функцию суммы значений `$sum` в сочетании с функцией умножения (`$multiply` для того, чтобы попарно перемножить количество проданного товара на стоимость единицы), а также функцией `$match` для задания условия ограничения для полученного результата (листинг 12).

Листинг 12. Скрипт запроса к хранилищу с группировкой, агрегацией и ограничением

```
db.sales.aggregate ([{$group:      {_id:      "$item",
totalSaleAmount:  {  $sum:  {  $multiply:  [  "$price",
"$quantity"  ]}}}}, {$match: {"totalSaleAmount": {$gte:
100}}}] )
```

Для выполнения следующего запроса следует предварительно ознакомиться с идентификаторами типов данных BSON, приведенными по ссылке <https://www.mongodb.com/docs/manual/reference/bson-types/>.

Вернитесь к хранилищу `provider` и в результатном запросе выведите пользователей из коллекции `users`, у которых значение `first_name` имеет тип данных `double` (числовой) (листинг 13).

Листинг 13. Скрипт запроса к хранилищу с группировкой, агрегацией и ограничением

```
db.users.find ({first_name: {$type : 1}})
```

Оцените результат запроса, попробуйте осуществить поиск по другим приведенным в таблице типам данных.

Список литературы

1. Kroenke D.M., Auer D.J. Database Processing: Fundamentals, Design and Implementation. PEARSON, 2019. - 691 с.: ил..
2. Кренке Д. Теория и практика построения баз данных - Спб.: Питер, 2005. - 859 с.: ил..
3. Ponniah P. Data Warehousing: Fundamentals for IT Professionals. Wiley, 2010. - 570 с.: ил..
4. Kroenke D.M., Auer D.J. Database Concepts. англ. - PEARSON, 2018. - 579 с.: ил..
5. Петкович Д. Microsoft SQL Server 2012. Руководство для начинающих. - Спб.: БХВ-Петербург, 2013. - 817 с.: ил..
6. Сарка Д., Лах М., Йеркич Г. Microsoft SQL Server 2012. Реализация хранилищ данных - Спб.: Наука, 2014. - 805 с.: ил..
7. Мартишин С.А., Симонов В.Л., Храпченко М.В. Базы данных. Практическое применение СУБД SQL и NOSQL-типа для проектирования информационных систем – Москва.: Инфра-М, 2016. – 367 с.: ил..



Сведения об авторе

Смирнов Михаил Вячеславович, кандидат экономических наук, доцент кафедры “Предметно-ориентированные информационные системы” Института кибербезопасности и цифровых технологий РТУ-МИРЭА.