Adventure Works
DWH and Reporting Solution Overview
for HR questions task

1 The Task (Description from Customer)

The goal of this test is to build BI solution that will answer the following questions of HR department:

1. How many people employed in each department (day, month and year level)?

2. What is average age in each department (day, month and year level)?

3. What is average seniority in each department (day, month and year level)?

The project will be build using MS SQL 2012 or higher and include:

1. Data mart (DWH)

2. ETL which populate DWH (SSIS)

3. OLAP cube (SSAS)

4. Final reports (SSRS/Excel)

The data for project - Adventure works OLTP.

The project should be submitted as following:

1. Back up of data base

2. Copy of all solutions (SSIS, SSAS, SSRS)

3. Instructions for execution

2 Initial analysis and communication with the customer

2.1 Correspondence

Hi Sergey,

Good thinking, let's proceed with #4 - Weighed approach.

BR,
<Customer Name>

Get Outlook for Android

From: Sergey Vdovin
Sent: Tuesday, July 31, 16:52

Subject: Re: BI Task

To: <Customer Name>

Cc: <Customer Representative>

<Customer Name>, hello.

I want to emphasis again that in my experience business usually does not go to such elaborated discussions before there is a kind of playground and we are in a special business case right now.

I am considering one business aspect for the task below (for example), for technical implementations there are different implementations at different levels as well, like this:

To calculate the measures i can suggest 4 generally accepted business methods.
The methods vary in how the data of an employee affects the value of a measure for selected period (day, month, year) and department:

1. Period's start date alignment: the data of the employee is used in the aggregation if the employee was working in the department at the period's start date.
2. Period's end date alignment: the data of the employee is used in the aggregation if the employee was working in the department at the period's end date.

3. LFL(Like-For-Like approach): may makes sense if to consider, for example the Productivity Tax - the data of the employee is used in the aggregation if the employee was working in the department during the whole period.

4. Weighed approach: the data of the employee affects the aggregation proportionally to the number of days the employer was working in the department during the period.

We can implement all the methods withing the SSAS Cube and to choose the method we may use a Shell Dimension

Looking forward to the reply.

Sergey Vdovin

sergeyavdovin.com

2.2 Other Assumptions

2.2.1 DWH Architecture

The request contains mention of the DWH what usually means that we do not consider isolated data mart to solve a specific task but want to start a data warehouse project with solving the specific task – so we will define dimension and fact tables, which can be used further for solving other kinds of analytical tasks. Currently brand new DWH project perhaps should be started with Data Vault data warehouse methodology but within this initial cycle we will use the conventional Kimbal’s DWH approach + Persistent storage = 3 layers of the data warehouse: 1. Staging 2. Persistent 3. Data Mart

2.2.2 Reporting Layer Choice

For an analytical reporting task probably, Power BI desktop should be considered (free desktop client) – for the initial cycle we go with SSRS report.

3 Solution

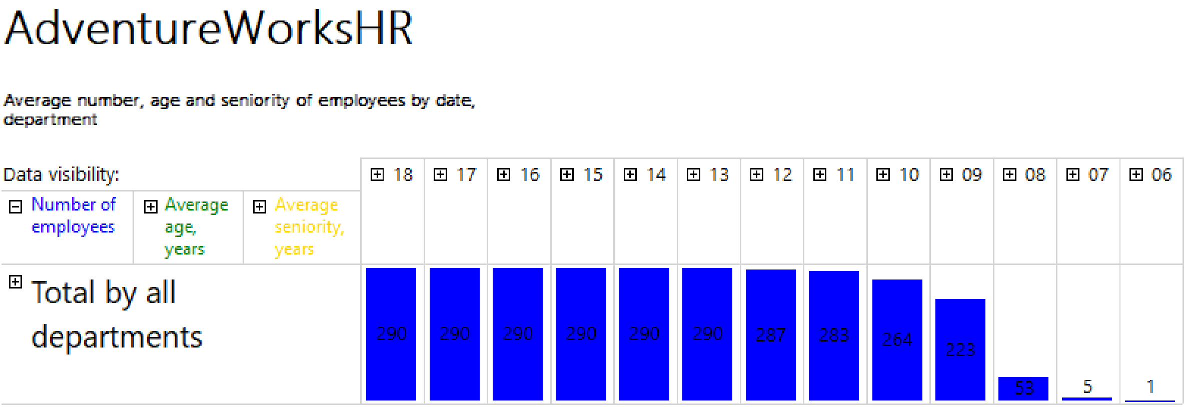
3.1 Reporting Services (SSRS) Layer

3.1.1 User Interface

The answers to the questions are available through SSRS report:

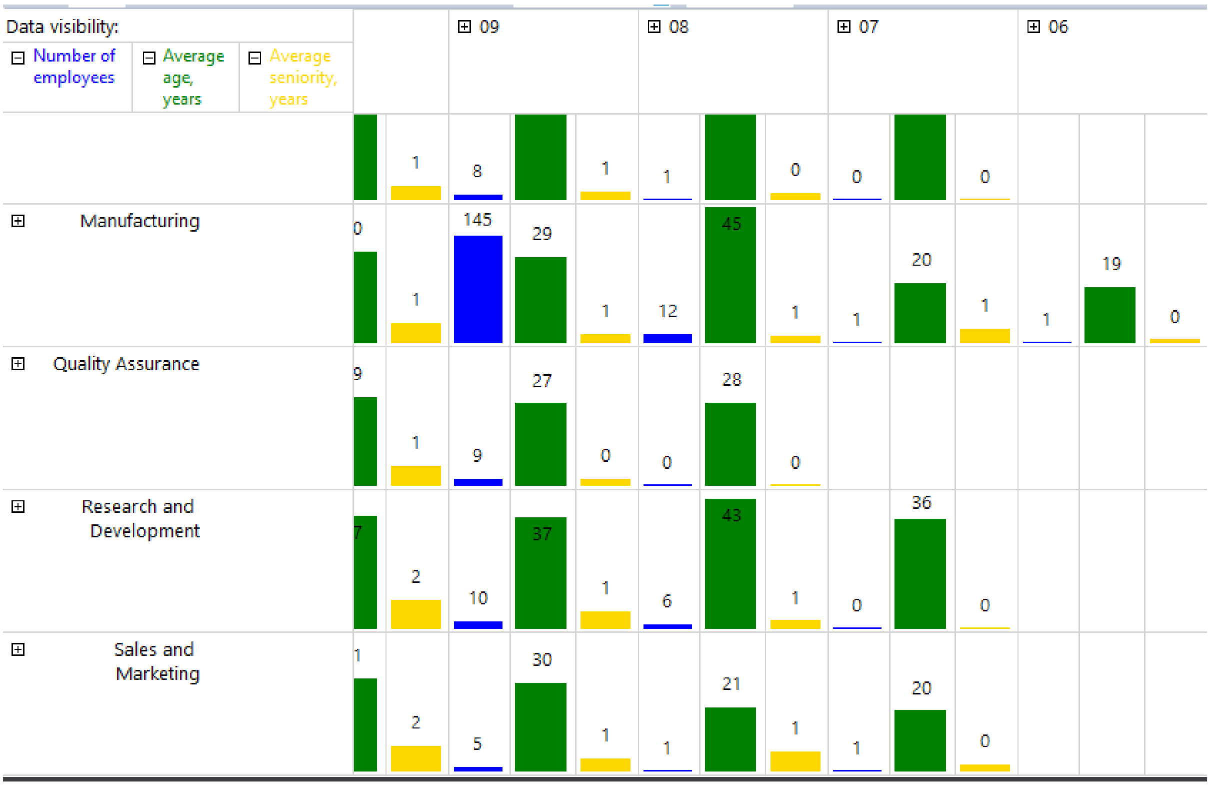
3.1.1.1 Initial View

After opening the report a user see a matrix with years and department groups.



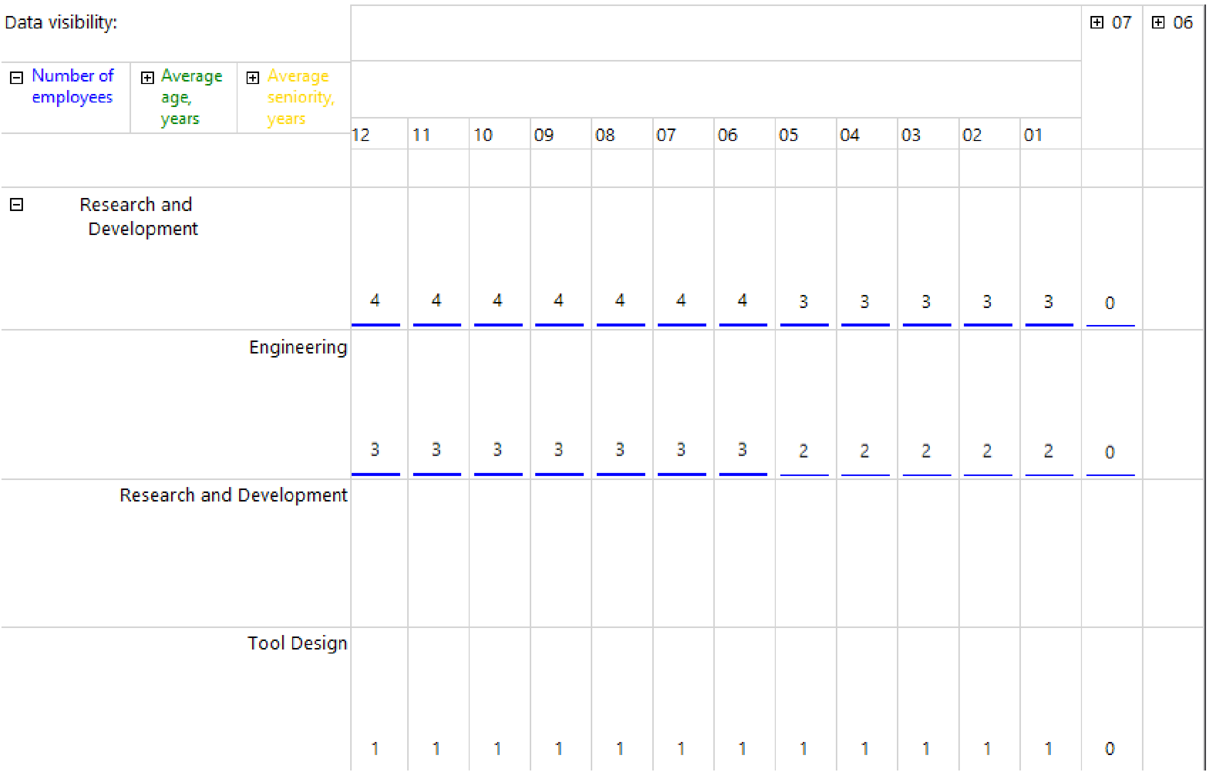
3.1.1.2 Adding different measures to the report, fixed headers

By pressing the Data Visibility elements, we can reach all 3 characteristics, the headers of rows and columns are fixed while scrolling:



3.1.1.3 Drill down functional

By pressing the headers we can navigate along the year-month-date and group-department hierarchies:



Data visibility:

⊟ Number of employees ⊞ Average age, years ⊞ Average seniority, years

		12	11	10	09	08	07	06	05	04	03	02	01	⊞ 07	⊞ 06
⊟ Research and Development		4	4	4	4	4	4	4	3	3	3	3	3	0	
	Engineering	3	3	3	3	3	3	3	2	2	2	2	2	0	
	Research and Development														
	Tool Design	1	1	1	1	1	1	1	1	1	1	1	1	0	

3.1.2 Techical Implementation

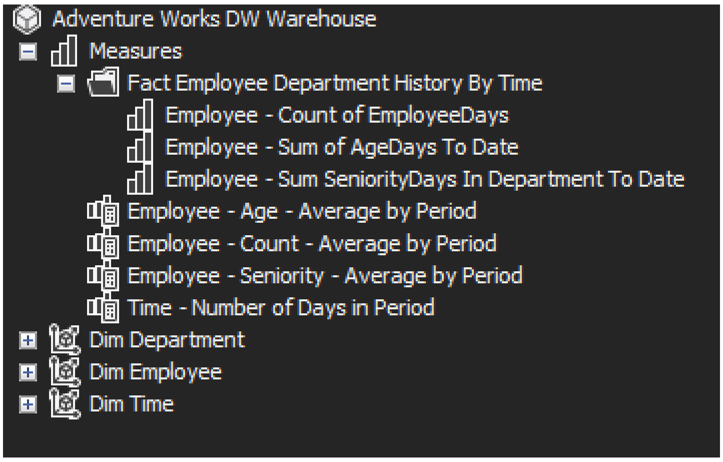
3.1.2.1 Server Aggregated Calculations

In order to use SSAS server aggregated calculations (when a formula defined at a different levels of hierarchies defines the values in the report totals) we use

- *Aggregate* SSRS function
- *MDX query in the report which returns information from all levels of hierarchies*
- *SSAS calculated measures which worke at all levels of hierarchies*

3.2 Analysis Services (SSAS) Layer

SSAS Database contains 1 measure group with 3 physical measures and 4 calculated measures



Adventure Works DW Warehouse
- Measures
  - Fact Employee Department History By Time
    - Employee - Count of EmployeeDays
    - Employee - Sum of AgeDays To Date
    - Employee - Sum SeniorityDays In Department To Date
  - Employee - Age - Average by Period
  - Employee - Count - Average by Period
  - Employee - Seniority - Average by Period
  - Time - Number of Days in Period
- Dim Department
- Dim Employee
- Dim Time

3.2.1 Default MDX Script with calculated members

Values to answer the HR questions are calculated in MDX (and in SQL below)

/*

The CALCULATE command controls the aggregation of leaf cells in the cube.

If the CALCULATE command is deleted or modified, the data within the cube is affected.

You should edit this command only if you manually specify how the cube is aggregated.

*/

CALCULATE;

CREATE MEMBER CURRENTCUBE.[Measures].[Employee - Age - Average by Period]

AS [Measures].[Employee - Sum of AgeDays To Date]/[Measures].[Employee - Count of EmployeeDays],

VISIBLE = 1 ;

CREATE MEMBER CURRENTCUBE.[Measures].[Employee - Count - Average by Period]

AS [Measures].[Employee - Count of EmployeeDays]/[Measures].[Time - Number of Days in Period],

VISIBLE = 1 ;

CREATE MEMBER CURRENTCUBE.[Measures].[Employee - Seniority - Average by Period]

AS [Measures].[Employee - Sum SeniorityDays In Department To Date]/[Measures].[Employee - Count of EmployeeDays],

FORMAT_STRING = "Short Date",

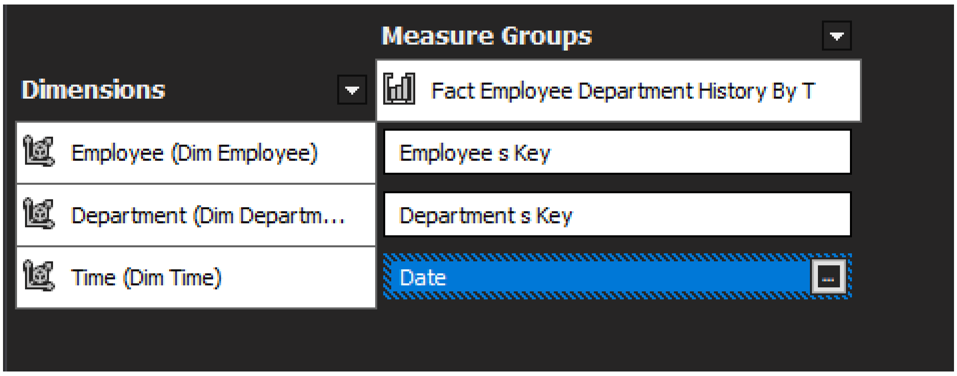
VISIBLE = 1 ;

CREATE MEMBER CURRENTCUBE.[Measures].[Time - Number of Days in Period]

```
AS count(descendants([Dim Time].[Hierarchy].currentmember,,leaves)),

VISIBLE = 1;
```

3.2.2 Dimensions usage matrix

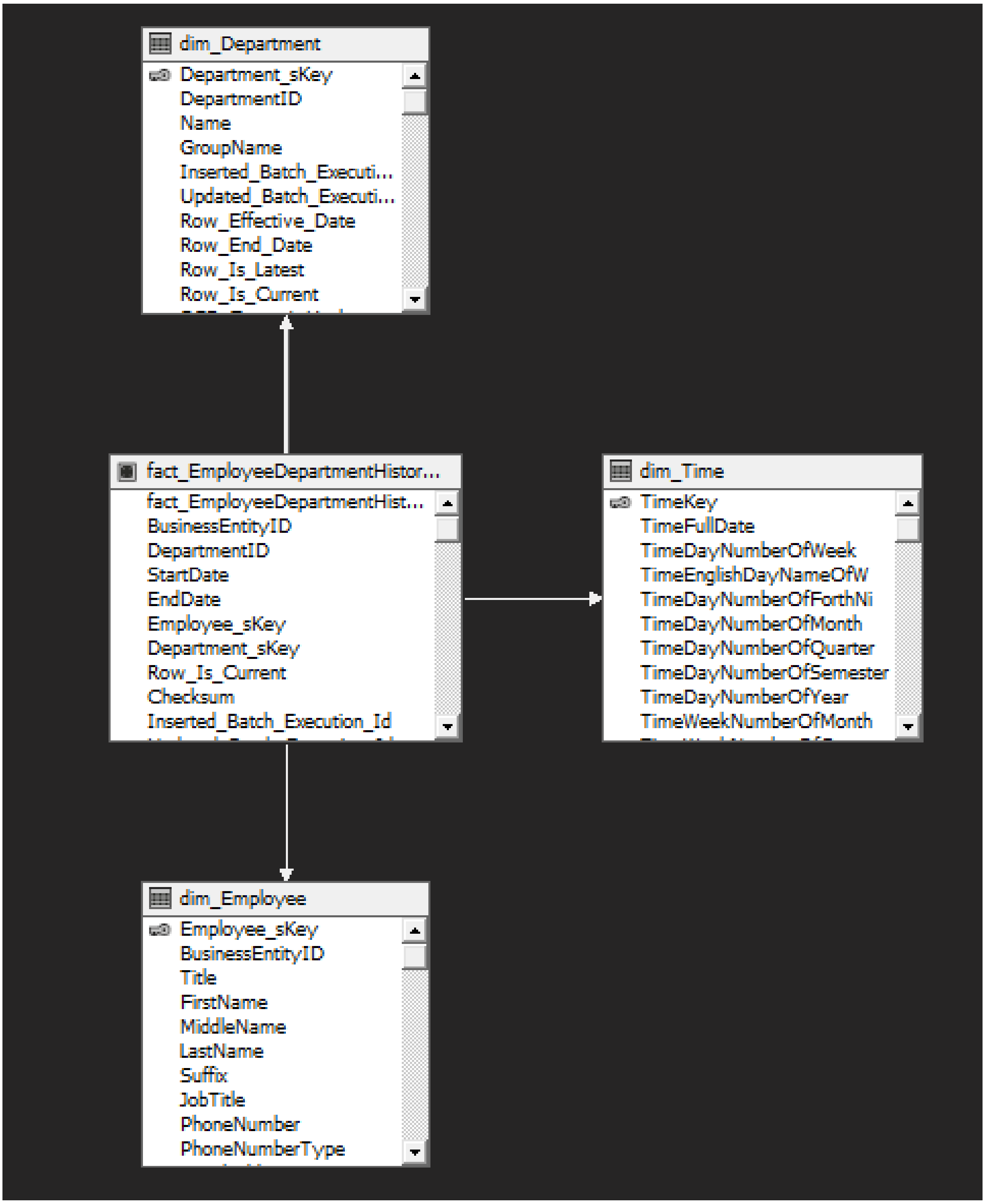
All relationships between measure groups and dimensions are regular, the relation is made with surrogate keys with the exception for the date dimension:



	Measure Groups
Dimensions	Fact Employee Department History By T
Employee (Dim Employee)	Employee s Key
Department (Dim Departm…	Department s Key
Time (Dim Time)	Date

3.2.3 Data Source View

Data source view contains references to 3 physical dimension tables and 1 fact view:



dim_Department
- Department_sKey
- DepartmentID
- Name
- GroupName
- Inserted_Batch_Executi...
- Updated_Batch_Executi...
- Row_Effective_Date
- Row_End_Date
- Row_Is_Latest
- Row_Is_Current

fact_EmployeeDepartmentHistor...
- fact_EmployeeDepartmentHist...
- BusinessEntityID
- DepartmentID
- StartDate
- EndDate
- Employee_sKey
- Department_sKey
- Row_Is_Current
- Checksum
- Inserted_Batch_Execution_Id

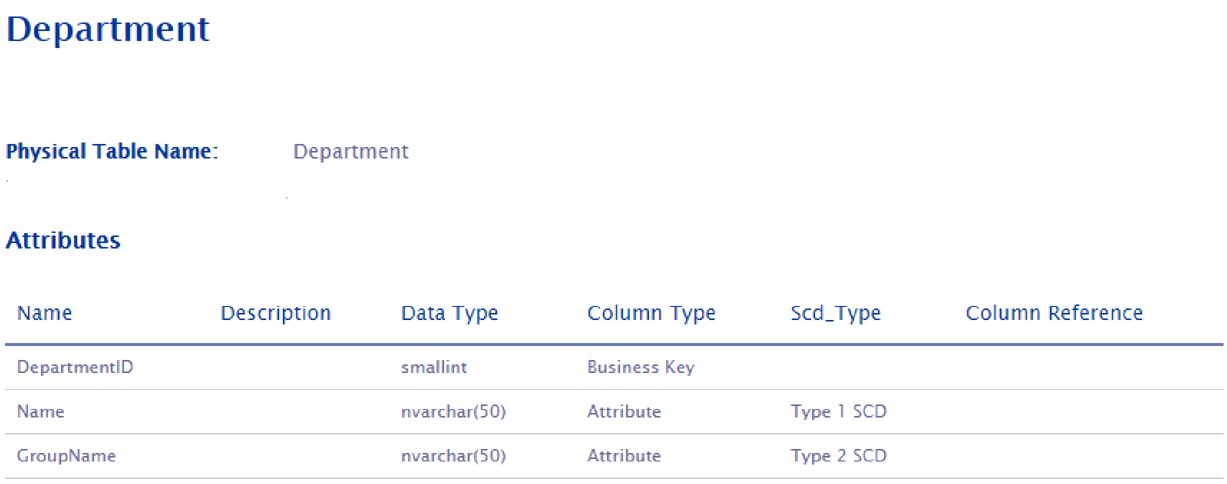
dim_Time
- TimeKey
- TimeFullDate
- TimeDayNumberOfWeek
- TimeEnglishDayNameOfW
- TimeDayNumberOfForthNi
- TimeDayNumberOfMonth
- TimeDayNumberOfQuarter
- TimeDayNumberOfSemester
- TimeDayNumberOfYear
- TimeWeekNumberOfMonth

dim_Employee
- Employee_sKey
- BusinessEntityID
- Title
- FirstName
- MiddleName
- LastName
- Suffix
- JobTitle
- PhoneNumber
- PhoneNumberType

3.3 DWH & ETL Layer – SQL Server Database Engine & SSIS

3.3.1 Dimensions - SCD1 SCD2

Both non time dimension contain SCD2 attributes:

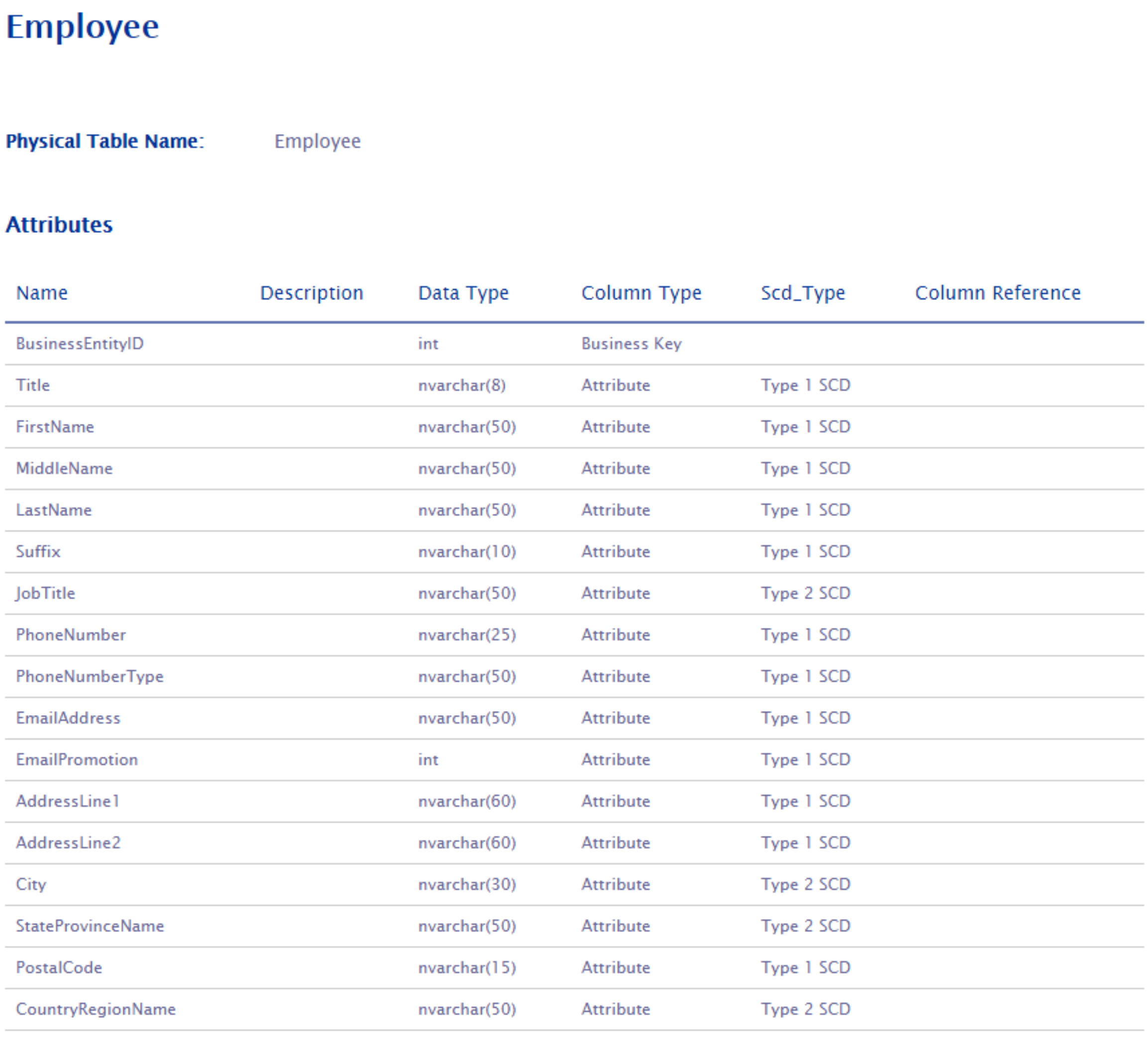


Department

Physical Table Name: Department

Attributes

Name	Description	Data Type	Column Type	Scd_Type	Column Reference
DepartmentID		smallint	Business Key		
Name		nvarchar(50)	Attribute	Type 1 SCD	
GroupName		nvarchar(50)	Attribute	Type 2 SCD	

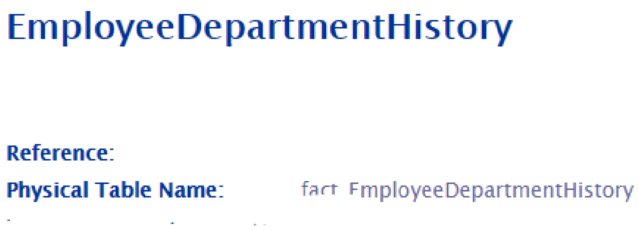


Employee

Physical Table Name: Employee

Attributes

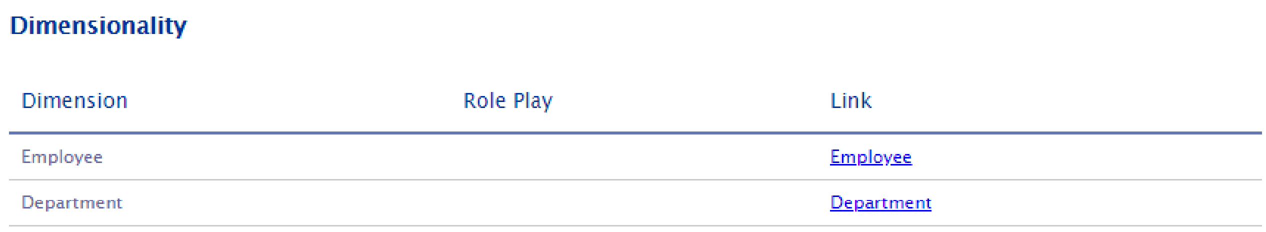
Name	Description	Data Type	Column Type	Scd_Type	Column Reference
BusinessEntityID		int	Business Key		
Title		nvarchar(8)	Attribute	Type 1 SCD	
FirstName		nvarchar(50)	Attribute	Type 1 SCD	
MiddleName		nvarchar(50)	Attribute	Type 1 SCD	
LastName		nvarchar(50)	Attribute	Type 1 SCD	
Suffix		nvarchar(10)	Attribute	Type 1 SCD	
JobTitle		nvarchar(50)	Attribute	Type 2 SCD	
PhoneNumber		nvarchar(25)	Attribute	Type 1 SCD	
PhoneNumberType		nvarchar(50)	Attribute	Type 1 SCD	
EmailAddress		nvarchar(50)	Attribute	Type 1 SCD	
EmailPromotion		int	Attribute	Type 1 SCD	
AddressLine1		nvarchar(60)	Attribute	Type 1 SCD	
AddressLine2		nvarchar(60)	Attribute	Type 1 SCD	
City		nvarchar(30)	Attribute	Type 2 SCD	
StateProvinceName		nvarchar(50)	Attribute	Type 2 SCD	
PostalCode		nvarchar(15)	Attribute	Type 1 SCD	
CountryRegionName		nvarchar(50)	Attribute	Type 2 SCD	



EmployeeDepartmentHistory

Reference:

Physical Table Name: fact_EmployeeDepartmentHistory



Dimensionality

Dimension	Role Play	Link
Employee		Employee
Department		Department

3.3.2 Fact view

Data for the measure group is calculated in a fact view:

```
ALTER view [fact].[fact_v_EmployeeDepartmentHistoryByTime]

 as

SELECT [fact_EmployeeDepartmentHistory_sKey]

   ,H.[BusinessEntityID]

   ,H.[DepartmentID]

   ,H.[StartDate]

   ,H.[EndDate]

   ,H.[Employee_sKey]

   ,H.[Department_sKey]

   ,H.[Row_Is_Current]

   ,H.[Checksum]

   ,H.[Inserted_Batch_Execution_Id]

   ,H.[Updated_Batch_Execution_Id]

   ,H.[Row_Effective_Date]

   ,H.[Row_End_Date]

   ,H.[Row_Is_Latest]

   ,[TimeKey]

   ,count(1) OVER (

      PARTITION BY H.[BusinessEntityID]

      ,[DepartmentID] ORDER BY [TimeKey] ROWS UNBOUNDED PRECEDING

      ) AS SeniorityDaysInDepartmentToDate
```

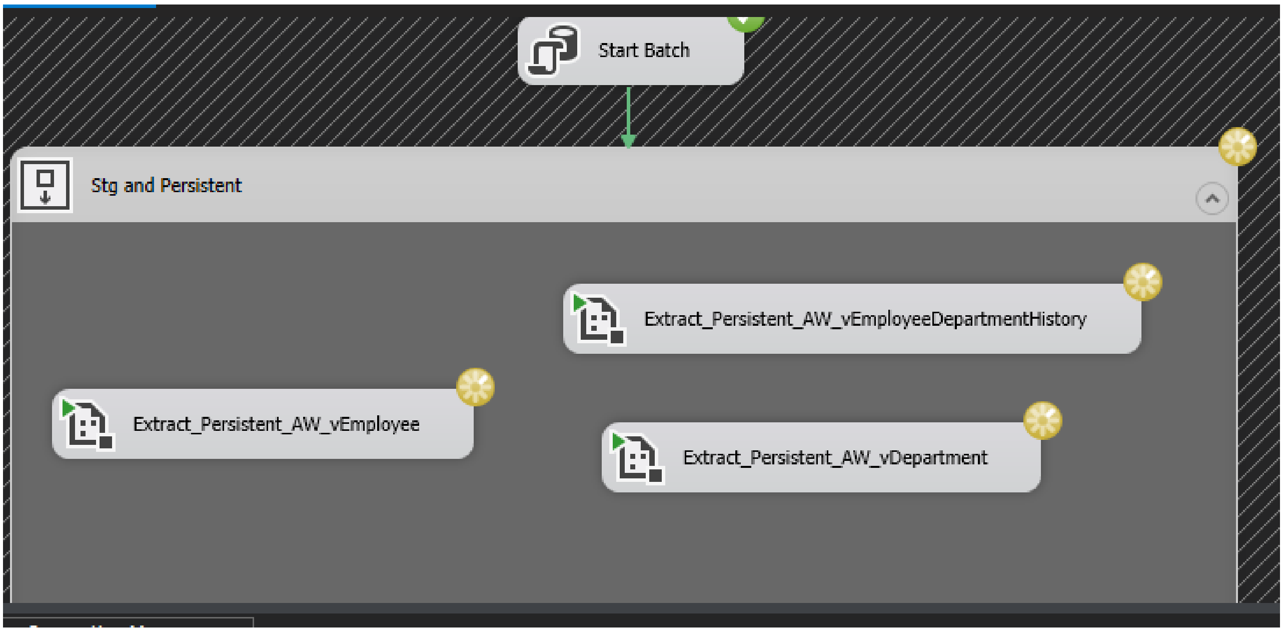
```
        --forgot to add birth date in the initial cycle - in POC getting in through another column

    ,datediff(day, cast(e.AddressLine2 AS DATE), CONVERT(DATE, CONVERT(NVARCHAR(10), [TimeKey], 112))) AS AgeToDate

FROM fact.[fact_EmployeeDepartmentHistory] H

LEFT JOIN [dim].[dim_Time] T ON [TimeKey] BETWEEN CONVERT(INT, CONVERT(NVARCHAR(10), [StartDate], 112))

        AND CONVERT(INT, CONVERT(NVARCHAR(10), isnull(EndDate, getdate()), 112))

LEFT JOIN [dim].[dim_Employee] E ON H.Employee_sKey = E.Employee_sKey
```

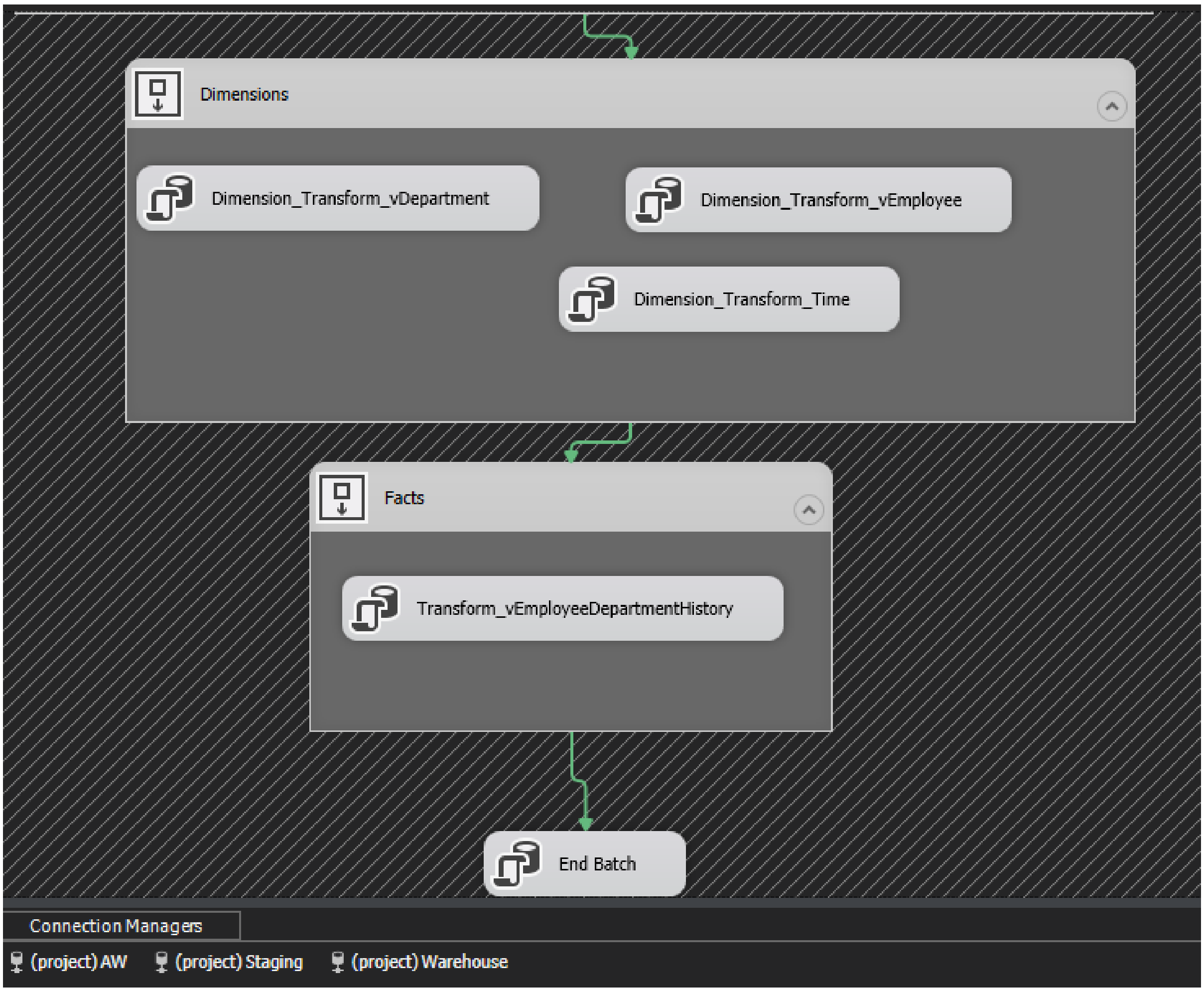
Here analytical function performs ~10 times faster than conventional approach with joining tables

3.3.3 SSIS

Although it was discussed that all transformation logic should be implemented via SSIS current infrastructure configuration does not allow to do so and while delivering the result we may return to the SSIS implementation during next cycles.

Currently in SSIS we have master package and staging, all other transformations – in T-SQL.


Start Batch
Stg and Persistent
Extract_Persistent_AW_vEmployeeDepartmentHistory
Extract_Persistent_AW_vEmployee
Extract_Persistent_AW_vDepartment

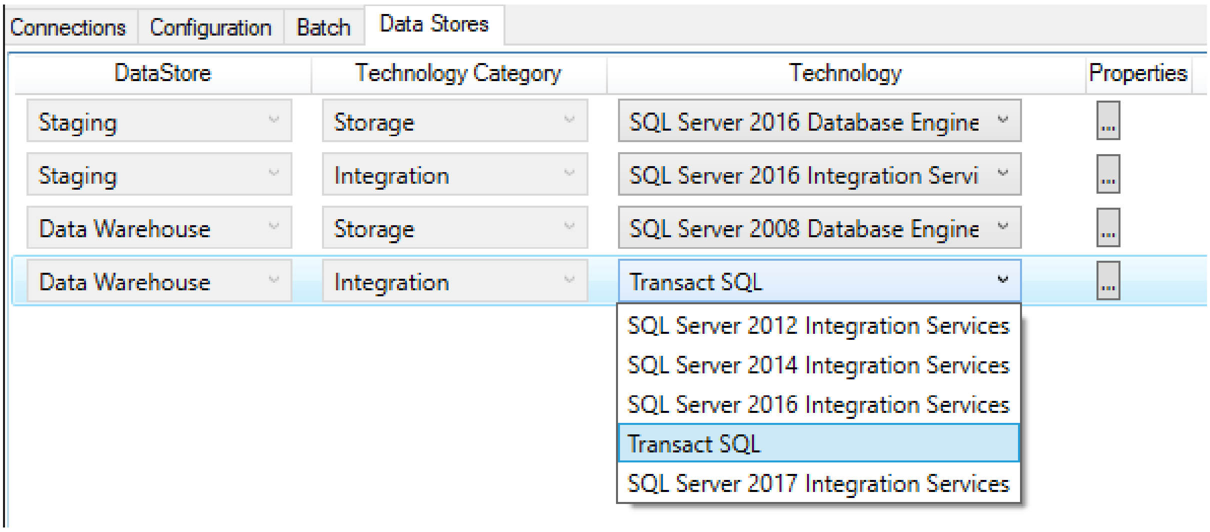


3.3.3.1 *SSIS vs T-SQL for Transformations*

First of all i want to emphasis that i am absolutely ok to work with SSIS for this logic and i do use SSIS on daily basis for that tasks and a lot of people use it as well. I just wanted to discuss that currently T-SQL is more popular for execution of this kind of tasks:

3.3.3.1.1 Data Warehouse Automation (DWA) Tools

In Data Warehouse Automation tools, for example - in WhereScape T-SQL it is the only option for SQL Server and in Dimodelo it is the default option:

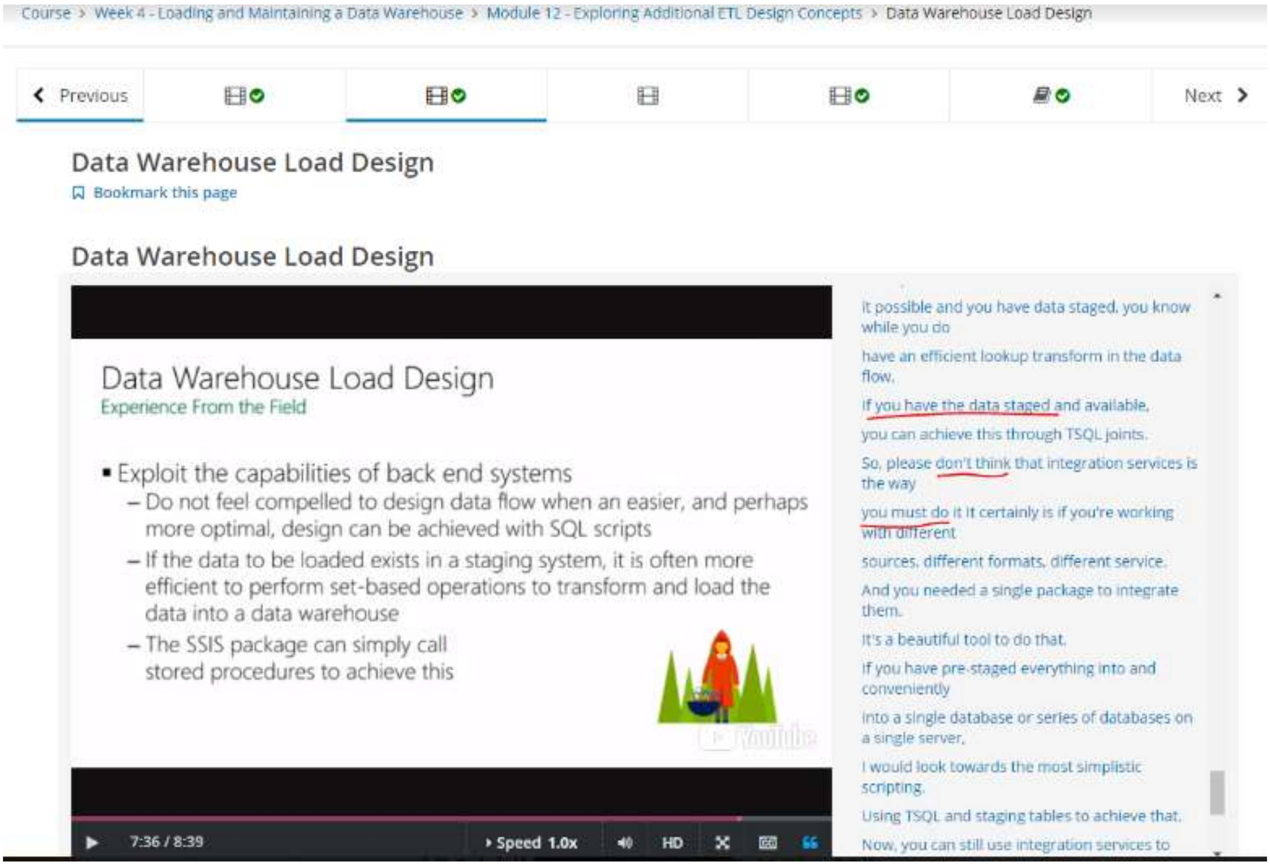

Connections
Configuration
Batch
Data Stores

DataStore	Technology Category	Technology	Properties
Staging	Storage	SQL Server 2016 Database Engine	...
Staging	Integration	SQL Server 2016 Integration Servi	...
Data Warehouse	Storage	SQL Server 2008 Database Engine	...
Data Warehouse	Integration	Transact SQL	...

SQL Server 2012 Integration Services
SQL Server 2014 Integration Services
SQL Server 2016 Integration Services
Transact SQL
SQL Server 2017 Integration Services

3.3.3.1.2 Relational Data Warehouse course from Microsoft

And even in the official Delivering a Relational Data Warehouse course from Microsoft the presenter tells that he prefers to use T-SQL for that transformations and encourages us to consider it as well:



Course > Week 4 - Loading and Maintaining a Data Warehouse > Module 12 - Exploring Additional ETL Design Concepts > Data Warehouse Load Design
Previous
Next
Bookmark this page
7:36 / 8:39
Speed 1.0x
HD

Data Warehouse Load Design

Data Warehouse Load Design

Data Warehouse Load Design
Experience From the Field

- Exploit the capabilities of back end systems
  - Do not feel compelled to design data flow when an easier, and perhaps more optimal, design can be achieved with SQL scripts
  - If the data to be loaded exists in a staging system, it is often more efficient to perform set-based operations to transform and load the data into a data warehouse
  - The SSIS package can simply call stored procedures to achieve this

it possible and you have data staged, you know while you do

have an efficient lookup transform in the data flow.

If you have the data staged and available,

you can achieve this through TSQL joints.

So, please don't think that integration services is the way

you must do it It certainly is if you're working with different

sources, different formats, different service.

And you needed a single package to integrate them.

It's a beautiful tool to do that.

If you have pre-staged everything into and conveniently

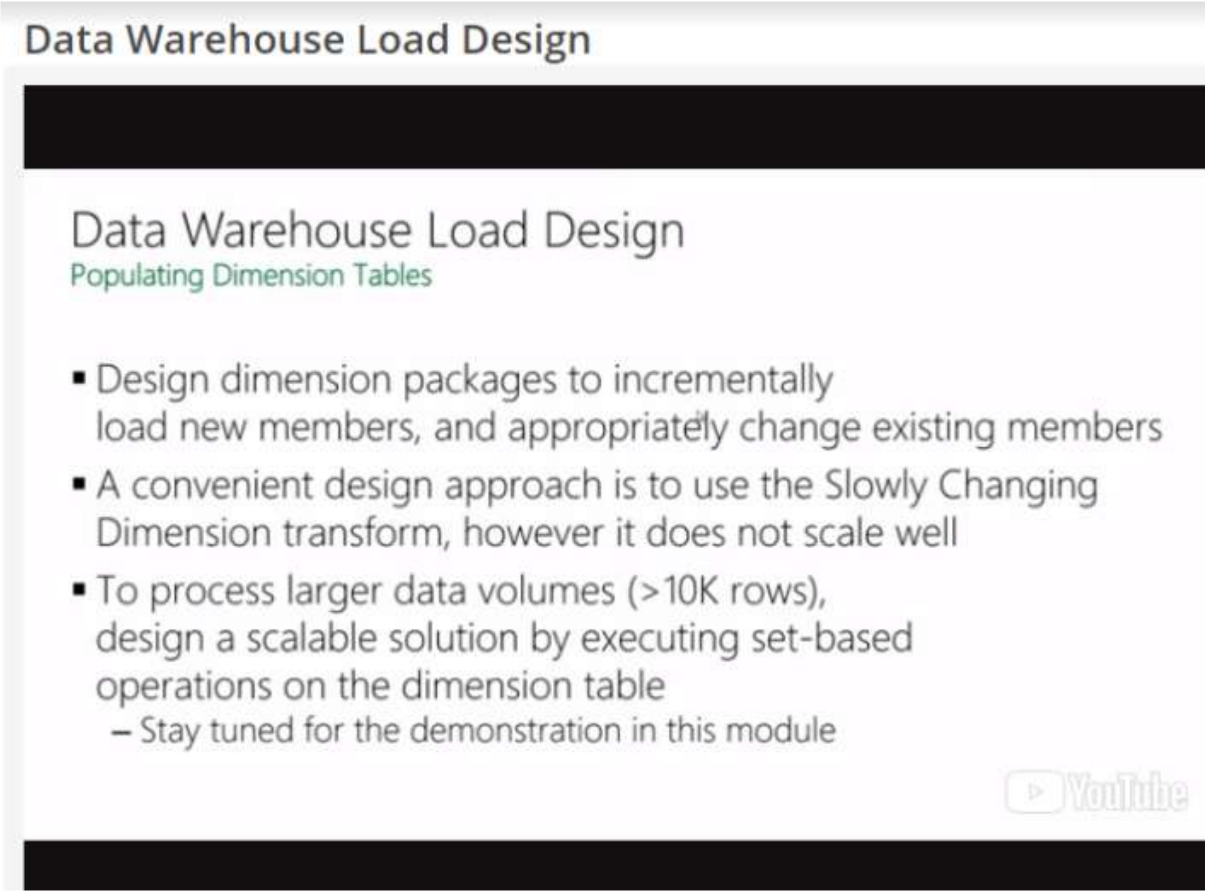
into a single database or series of databases on a single server,

I would look towards the most simplistic scripting.

Using TSQL and staging tables to achieve that.

Now, you can still use integration services to

Moreover for scalability it is directly advised to use T-SQL and not SSIS:



Data Warehouse Load Design

Data Warehouse Load Design

Populating Dimension Tables

- Design dimension packages to incrementally load new members, and appropriately change existing members
- A convenient design approach is to use the Slowly Changing Dimension transform, however it does not scale well
- To process larger data volumes (>10K rows), design a scalable solution by executing set-based operations on the dimension table
  - Stay tuned for the demonstration in this module

transcript>>>>>*So if you have volumes in excess typically of around 10,000 rows,*

you look to produce some more scalable solutions using

set based operations.

Set Based Operations are very efficient for

a relational engine to do and to achieve this,

you could create some temporary staging tables,

you could load into them by your own data flow logic.

Here are the new members, here are the members with type one

changes, type two changes, and

if you can persist those into these temporary staging tables,

then on success of the data flow you could execute a SQL task

that via joins to these staging tables could perform set based

inserts and updates <<<<

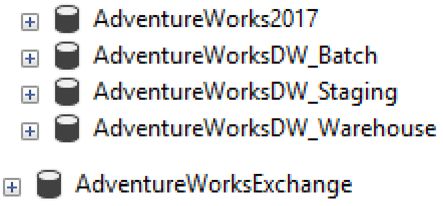
3.3.4 Extraction

Data from source system in the first place is extracted to dedicated AdventureWorksExchange database and then to staging. It is an emulation that the source system is usually have some export routines.

4 Deployment

All the databases should be deployed on one server on default instances, to run the report one may go directly to 5.3 – the achieve of the SSAS database contains processed database.

4.1 SQL Server Database Engine



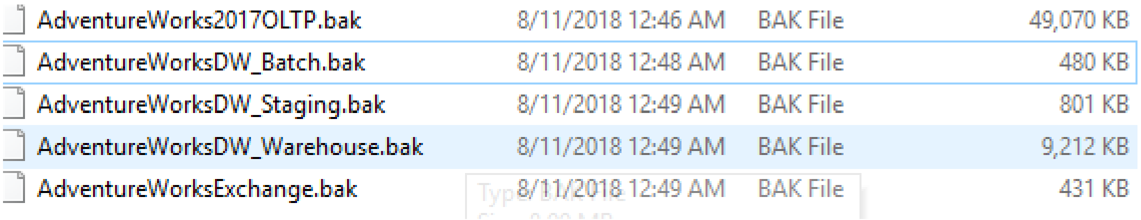
⊞ AdventureWorks2017
⊞ AdventureWorksDW_Batch
⊞ AdventureWorksDW_Staging
⊞ AdventureWorksDW_Warehouse

⊞ AdventureWorksExchange

Adventure works can be downloaded from official microsoft's GitHub repository:

https://github.com/Microsoft/sql-server-samples/releases

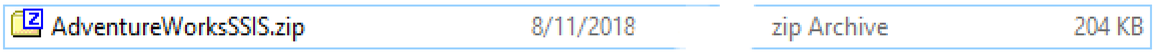
other files are included:



AdventureWorks2017OLTP.bak	8/11/2018 12:46 AM	BAK File	49,070 KB
AdventureWorksDW_Batch.bak	8/11/2018 12:48 AM	BAK File	480 KB
AdventureWorksDW_Staging.bak	8/11/2018 12:49 AM	BAK File	801 KB
AdventureWorksDW_Warehouse.bak	8/11/2018 12:49 AM	BAK File	9,212 KB
AdventureWorksExchange.bak	8/11/2018 12:49 AM	BAK File	431 KB

4.2 SQL Server Integration Services

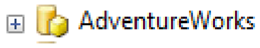
Visual Studio 2015 solution can be found in achieve:



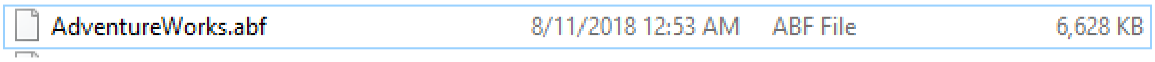
AdventureWorksSSIS.zip	8/11/2018	zip Archive	204 KB

To run the ETL process one should launch the Master Package

4.3 SQL Server Analysis Services



⊞ AdventureWorks



AdventureWorks.abf	8/11/2018 12:53 AM	ABF File	6,628 KB

The load of the multidimensional model currently is manual – one may run the full process on the database.

4.4 SQL Server Reporting Services

To launch the report one can unpack the solution (Visual Studio 2017) and launch the Report in the Business Intelligence Development Environment