Northwind BI solution

Index

[Introduction 6](#_Toc146055989)

[Planning a BI Solution 6](#_Toc146055990)

[Project Scope 6](#_Toc146055991)

[Required BI Software 6](#_Toc146055992)

[High-level project schema 7](#_Toc146055993)

[Project Infrastructure 8](#_Toc146055994)

[Project Management Infrastructure 8](#_Toc146055995)

[Design and Development Tools 8](#_Toc146055996)

[Development and Test Infrastructure 8](#_Toc146055997)

[Project Personnel Roles 8](#_Toc146055998)

[Project Stakeholders 9](#_Toc146055999)

[Planning SQL Server Business Intelligence Infrastructure 10](#_Toc146056000)

[BI Topology 10](#_Toc146056001)

[Single BI server 10](#_Toc146056002)

[Dedicated report server 11](#_Toc146056003)

[Dedicated ETL server 12](#_Toc146056004)

[Dedicated master data server 13](#_Toc146056005)

[Distributed report server 14](#_Toc146056006)

[Distributed MDM server 15](#_Toc146056007)

[Planning Data Warehouse Hardware 16](#_Toc146056008)

[Designing Data Warehouse 17](#_Toc146056009)

[Designing Data Warehouse Logical Schema 17](#_Toc146056010)

[Identify business processes 17](#_Toc146056011)

[Designing Dimension Models and Data Warehouse Tables 18](#_Toc146056012)

[Order dimensional model 18](#_Toc146056013)

[Dimension 19](#_Toc146056014)

[Customer 19](#_Toc146056015)

[Date 19](#_Toc146056016)

[Employee 20](#_Toc146056017)

[Product 20](#_Toc146056018)

[Fact 20](#_Toc146056019)

[Order 20](#_Toc146056020)

[Designing Data Warehouse Physical Implementation 21](#_Toc146056021)

[Designing File Storage 21](#_Toc146056022)

[Partitioning 21](#_Toc146056023)

[Indexes 21](#_Toc146056024)

[Compression 22](#_Toc146056025)

[Views 22](#_Toc146056026)

[Designing ETL Solution 23](#_Toc146056027)

[ETL Architecture 23](#_Toc146056028)

[Data Flow Diagrams 24](#_Toc146056029)

[Dimension 24](#_Toc146056030)

[Customer 24](#_Toc146056031)

[Date 25](#_Toc146056032)

[Employee 26](#_Toc146056033)

[Product 27](#_Toc146056034)

[Fact 28](#_Toc146056035)

[Order 28](#_Toc146056036)

[Source To Target Mapping 28](#_Toc146056037)

[Planning Data Extraction 29](#_Toc146056038)

[Data source profiling 29](#_Toc146056039)

[Identifying New and Modified Rows 29](#_Toc146056040)

[Planning Extraction Windows 30](#_Toc146056041)

[Planning Data Transformation 30](#_Toc146056042)

[Transact-SQL vs. Data Flow Transformations 30](#_Toc146056043)

[Handling Invalid Rows and Errors 30](#_Toc146056044)

[Logging Audit Information 30](#_Toc146056045)

[Planning Data Loads 31](#_Toc146056046)

[Minimizing Logging 31](#_Toc146056047)

[Loading Indexed Tables 31](#_Toc146056048)

[Loading Partitioned Fact Tables 31](#_Toc146056049)

[Planning BI Delivery Solution 32](#_Toc146056050)

[Reporting Requirements 32](#_Toc146056051)

[Monitoring and Optimizing BI Solution 33](#_Toc146056052)

[Creating a Performance Baseline 33](#_Toc146056053)

[Monitoring ETL 33](#_Toc146056054)

[Package Execution Reports 33](#_Toc146056055)

[Monitoring and optimizing Data Warehouse 33](#_Toc146056056)

[Considerations for Data Warehouse Performance 33](#_Toc146056057)

[Data warehouse workloads: 33](#_Toc146056058)

[SQL Server settings 34](#_Toc146056059)

[Monitoring tools 34](#_Toc146056060)

[Turning tools 34](#_Toc146056061)

[Using Resource Governor to Balance Resource Utilization 34](#_Toc146056062)

[Troubleshooting Data Warehouse Performance 35](#_Toc146056063)

[Monitoring DQS 35](#_Toc146056064)

[Monitoring MDS 35](#_Toc146056065)

[Monitoring and optimizing Power BI Report Server 36](#_Toc146056066)

[Considerations for Reporting Services Performance 36](#_Toc146056067)

[Reporting Services workloads 36](#_Toc146056068)

[Reporting Services settings 36](#_Toc146056069)

[Monitoring Tools 36](#_Toc146056070)

[Tune 36](#_Toc146056071)

[Using Caching and Snapshots to Optimize Performance 36](#_Toc146056072)

[Operating BI Solution 38](#_Toc146056073)

[ETL Operations 38](#_Toc146056074)

[Deploying and Configuring Packages 38](#_Toc146056075)

[Environments 38](#_Toc146056076)

[Timing of package execution 38](#_Toc146056077)

[Execution account 38](#_Toc146056078)

[Backing up the SSIS Catalog 38](#_Toc146056079)

[Data Warehouse Operations 39](#_Toc146056080)

[Reorganizing or rebuilding indexes 39](#_Toc146056081)

[Updating data distribution statistics 39](#_Toc146056082)

[Managing partitioned tables 39](#_Toc146056083)

[Managing database files 39](#_Toc146056084)

[Security 39](#_Toc146056085)

[Backing up the database 39](#_Toc146056086)

[DQS Operations 40](#_Toc146056087)

[Configuration 40](#_Toc146056088)

[Security 40](#_Toc146056089)

[Backing up DQS databases 40](#_Toc146056090)

[MDS Operations 41](#_Toc146056091)

[Security 41](#_Toc146056092)

[Backing up MDS database 41](#_Toc146056093)

[Cleaning up transactions and staging tables 41](#_Toc146056094)

[Reporting Services Operations 41](#_Toc146056095)

[Managing Scheduled Tasks 41](#_Toc146056096)

[Backing up Power BI Report Server 42](#_Toc146056097)

[Deployment 42](#_Toc146056098)

[Solution structure 42](#_Toc146056099)

[Parameters 42](#_Toc146056100)

[Manual deployment 43](#_Toc146056101)

[Manual testing 43](#_Toc146056102)

[Automating Operational Tasks 44](#_Toc146056103)

[SQL agent 44](#_Toc146056104)

[Subsystems maintenance plan 44](#_Toc146056105)

[Full backup 44](#_Toc146056106)

[Differential backup 45](#_Toc146056107)

[Log backup 45](#_Toc146056108)

[Disaster recovery plan 46](#_Toc146056109)

# Introduction

Northwind Traders is a fictitious organization that manages orders, products, customers, suppliers, and many other aspects of a small business. This example is distributed by Microsoft.

The goal of this project is to develop a BI solution that provides sales analytics for the management of Northwind Traders.

# Planning a BI Solution

## Project Scope

|  |  |  |
| --- | --- | --- |
| High business value |  | * See sales amount with discount by customers and products. * Compare sales amount with discount with previous year/quarter/month. * See discount by employees and their geography. * See new and returned customers. * Cluster customers based on spending volume. * Product ABC classification. * Product basket analysis. * See number of orders, sales amount, sales amount with discount in progress. |
| Low business value |  |  |
|  | Low feasibility | High feasibility |

## Required BI Software

|  |  |  |
| --- | --- | --- |
| Software | | Rationale |
| SQL Server Database Engine | * The CEO wants a consistent view of all business data, and a centralized data warehouse in SQL Server would provide this. | |
| SQL Server Integration Services | * The business data required for analysis and reporting is currently spread across a range of data sources. Integration Services will provide an ETL platform to populate and refresh the data warehouse. | |
| SQL Server Master Data Services | * The Sales VP has complained about inconsistent data, which could potentially be caused by a lack of central data management for key business entities. | |
| SQL Server Data Quality Services | * The specialists have difficulty ensuring that their analysis of sales data is accurate because of quality issues in the data. Data Quality Services could be used to cleanse records and improve the consistency, thereby reducing the inaccuracy of the sales analysis. | |
| Power BI Report Server | * Business users need the centralized portal for analysis and reporting. * The database administrators team needs to receive email notifications about errors in the ETL process. * Data stewards need to be notified of errors and raw values in data cleansing processes. | |

## High-level project schema

Data sources

Structured



Semi-structured

ETL

*SQL Server Integration Services*



Reporting & analysis

*Power BI Report Server*

Data Cleansing

*SQL Server*

*Data Quality Services*



Master data

management

*SQL Server*

*Master Data Services*



Data

Warehouse

*SQL Server*

*Database engine*



## Project Infrastructure

### Project Management Infrastructure

* Microsoft Office
* Microsoft Project
  + [Northwind BI Plan.mpp](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/Docs/Northwind%20BI%20Plan.mpp)
  + [Northwind BI Plan with prototype.mpp](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/Docs/Northwind%20BI%20Plan%20with%20prototype.mpp)
* A SharePoint Server site for the project.

### Design and Development Tools

* Microsoft Visio to support diagrammatic design.
* SQL Server Data Tools to develop data models, reports, and SSIS packages ([Download SQL Server Data Tools (SSDT)](https://learn.microsoft.com/en-us/sql/ssdt/download-sql-server-data-tools-ssdt?view=sql-server-ver15#install-ssdt-with-visual-studio-2019))
  + [Visual Studio](https://visualstudio.microsoft.com/vs/older-downloads/#visual-studio-2019-and-other-products) ([Configuration file](https://1drv.ms/u/s!AueE3aY8CojZlNcXQwPa7Xq1h1BaWw?e=1Vt8Uc))
  + Extensions
    - [SQL Server Integration Services Projects](https://marketplace.visualstudio.com/items?itemName=SSIS.SqlServerIntegrationServicesProjects)
    - [Microsoft Reporting Services Projects](https://marketplace.visualstudio.com/items?itemName=ProBITools.MicrosoftReportProjectsforVisualStudio)
* [Power BI Desktop RS January 2023](https://www.microsoft.com/en-us/download/details.aspx?id=55330)
* [Tabular Editor](https://sql.bi/53569)
* [DAX Studio](https://sql.bi/980)
* [Analyze in Excel for Power BI Desktop](https://sql.bi/675172)
* [ALM Toolkit](https://sql.bi/629988)
* Other Microsoft Visual Studio components to develop custom application components.
* Azure DevOps (Server or Service) to provide source control, issue tracking capabilities, build and deployment automation.

### Development and Test Infrastructure

* Environments:
  + Development Servers
  + Test Servers
* Software:
  + [Download SQL Server 2019 Developer Edition](https://go.microsoft.com/fwlink/?linkid=866662) ([Configuration file](https://1drv.ms/u/s!AueE3aY8CojZlNccCBWmm3oHgW9A0w?e=04eazX))
  + [Download Microsoft Power BI Report Server - January 2023](https://www.microsoft.com/en-us/download/details.aspx?id=55329)

### Project Personnel Roles

* **A project manager.** Coordinates project tasks and schedules and ensures that the project is completed on time and within budget.
* **A BI solution architect**. Has overall responsibility for the technical design of the data warehousing solution.
* **A data modeler.** Designs the data warehouse schema and analytical data models.
* **A database administrator.** Designs the physical architecture and configuration of the data warehouse database. In addition, database administrators who have responsibility for data sources that are used in the data warehousing solution must be involved in the project to provide access to the data sources that the ETL process uses.
* **An infrastructure specialist.** Implements the server and network infrastructure for the data warehousing solution.
* **An ETL developer.** Builds the ETL workflow for the data warehousing solution.
* **A report developer.** Creates the reporting elements of the BI solution.
* **Business users.** Provide requirements and help to prioritize the business questions that the data warehousing solution will answer. Often, the team includes a business analyst as a full-time member to help interpret the business questions and ensure that the solution design meets the needs of the users.
* **Testers.** Verify the business and operational functionality of the solution as it is developed.

### Project Stakeholders

* **Executive sponsor.** Ensures BI project supports business vision. Crucial for project success.
* **Business users.** Provide detailed understanding of business processes, terminology, and metrics.
* **Data stewards.** Use business knowledge to validate and maintain business entity data.

# Planning SQL Server Business Intelligence Infrastructure

## BI Topology

### Single BI server



* SQL Server DB Engine
  + Data Warehouse
  + Landing zone
  + Staging databases
  + SSIS catalog
  + Reporting Services catalog
  + DQS catalog
  + MDS catalog
  + Logs database
* SQL Server Integration Services
* Power BI Report Server
* Data Quality Services
* Master Data Services

SRVBI



CLIENT PC

* Microsoft Excel
  + Power Query
  + Power Pivot
  + Power View
* Report Builder
* Power BI Desktop
* Microsoft Edge

Notes:

* Minimal server hardware and software license requirements, but the server would require significant memory, CPU, and disk resources for all but the most lightweight BI workloads.
* The range of different workload types on the server would make it difficult to specify and configure hardware resources appropriately.
* The server could be clustered to provide high availability.

### Dedicated report server



* SQL Server DB Engine
  + Data Warehouse
  + Landing zone
  + Staging databases
  + SSIS catalog
  + DQS catalog
  + MDS catalog
  + Logs database
* SQL Server Integration Services
* Data Quality Services
* Master Data Services

SRVDWH



CLIENT PC

* Microsoft Excel
  + Power Query
  + Power Pivot
  + Power View
* Report Builder
* Power BI Desktop
* Microsoft Edge



* SQL Server DB Engine
  + Reporting Services catalog
* Power BI Report Server

SRVPBIRS

Notes:

* The reporting and DWH workloads are full separated.
* Using a dedicated reporting server makes it easier to manage several types of workloads.
* Additional report servers could be added to scale out reporting.
* One or both servers could be clustered to provide high availability.

### Dedicated ETL server



* SQL Server DB Engine
  + Data Warehouse
  + Landing zone

SRVDWH



CLIENT PC

* Microsoft Excel
  + Power Query
  + Power Pivot
  + Power View
* Report Builder
* Power BI Desktop
* Microsoft Edge



* SQL Server DB Engine
  + Reporting Services catalog
* Power BI Report Server

SRVPBIRS



* SQL Server DB Engine
  + Staging databases
  + SSIS catalog
  + DQS catalog
  + MDS catalog
  + Logs database
* SQL Server Integration Services
* Data Quality Services
* Master Data Services

SRVETL

Notes:

* The DWH, reporting and ETL workloads are full separated.
* The landing database could be hosted on the ETL server or data warehouse server depending on workloads.
* SRVDWH, SRVPBIRS, and SRVETL could be clustered to provide high availability.
* Additional report servers and ETL could be added to scale out.

### Dedicated master data server



* SQL Server DB Engine
  + Data Warehouse
  + Landing zone

SRVDWH



CLIENT PC

* Microsoft Excel
  + Power Query
  + Power Pivot
  + Power View
* Report Builder
* Power BI Desktop
* Microsoft Edge



* SQL Server DB Engine
  + Reporting Services catalog
* Power BI Report Server

SRVPBIRS



* SQL Server DB Engine
  + DQS catalog
  + MDS catalog
  + Staging databases
* Data Quality Services
* Master Data Services

SRVMDM



* SQL Server DB Engine
  + SSIS catalog
  + Logs database
* SQL Server Integration Services

SRVETL

Notes:

* The DWH, reporting, ETL and MDM workloads are full separated.
* The landing database could be hosted on the ETL server or data warehouse server depending on workloads.
* Servers could be clustered to provide high availability.
* Additional report servers and ETL could be added to scale out.

### Distributed report server



* SQL Server DB Engine
  + Data Warehouse
  + Landing zone

SRVDWH



CLIENT PC

* Microsoft Excel
  + Power Query
  + Power Pivot
  + Power View
* Report Builder
* Power BI Desktop
* Microsoft Edge



* Power BI Report Server

SRVPBIRS



* SQL Server DB Engine
  + DQS catalog
  + MDS catalog
  + Staging databases
* Data Quality Services
* Master Data Services

SRVMDM



* SQL Server DB Engine
  + SSIS catalog
  + Logs database
* SQL Server Integration Services

SRVETL



* SQL Server DB Engine
  + Reporting Services catalog

SRVPBIRSDB

Notes:

* The DWH, reporting, ETL and MDM workloads are full separated.
* The landing database could be hosted on the ETL server or data warehouse server depending on workloads.
* The report server uses a separate database server to host the report catalog.
* Servers could be clustered to provide high availability.
* Additional report servers and ETL could be added to scale out.

### Distributed MDM server



* SQL Server DB Engine
  + Data Warehouse
  + Landing zone

SRVDWH



CLIENT PC

* Microsoft Excel
  + Power Query
  + Power Pivot
  + Power View
* Report Builder
* Power BI Desktop
* Microsoft Edge



* Power BI Report Server

SRVPBIRS



* Data Quality Services
* Master Data Services

SRVMDM



* SQL Server DB Engine
  + SSIS catalog
  + Logs database
* SQL Server Integration Services

SRVETL



* SQL Server DB Engine
  + Reporting Services catalog

SRVPBIRSDB



* SQL Server DB Engine
  + DQS catalog
  + MDS catalog
  + Staging databases

SRVMDMDB

Notes:

* The DWH, reporting, ETL and MDM workloads are full separated.
* The landing database could be hosted on the ETL server or data warehouse server depending on workloads.
* The report server uses a separate database server to host the report catalog.
* Servers could be clustered to provide high availability.
* Additional report servers and ETL could be added to scale out.

## Planning Data Warehouse Hardware

[DW Hardware Spec.xlsx](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/Docs/DW%20Hardware%20Spec.xlsx&version=GBmaster&_a=contents)

# Designing Data Warehouse

## Designing Data Warehouse Logical Schema

### Identify business processes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Business processes** | **Dimensions** | | | |
| Date | Product | Customer | Employee |
| Orders | X | X | X | X |

Notes:

* There is no interest in shipment geography, so there is no reason to create dedicated geography dimension. Employee and Customer geography will be implemented as attributes.
* In Date dimension will be used opensource project [Производственный календарь в XML (xmlcalendar.ru)](http://xmlcalendar.ru/) for getting information about production calendar.

### Designing Dimension Models and Data Warehouse Tables

#### Order dimensional model

Date

(Order, Required, Shipped)

Year

Quarter

Month

Day

Year

Quarter

Mon

ISO Week

Weekday

Year

Quarter

Month

Week

Weekday

Customer

Country

City

Customer

Contact name

Contact title

Phone

Fax

DayOfQuarterNumber

DayOfYearNumber

StartOfYear

EndOfYear

YearQuarter

StartOfQuarter

EndOfQuarter

YearMonth

StartOfMonth

EndOfMonth

StartOfWeek

EndOfWeek

Holiday

Workday type

Workday hours

Product

Category

Product

Employee

Country

City

Employee

Title

Title of courtesy

### Dimension

#### Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN NAME | COLUMN TYPE | SCD TYPE | DATA TYPE | NULL |
| CustomerKey | Surrogate key |  | INT | NOT |
| CustomerAlterKey | Business key |  | NVARCHAR(5) | NOT |
| Customer | Drill-thought detail | 1 | NVARCHAR(50) | NOT |
| ContactName | Drill-thought detail | 1 | NVARCHAR(50) |  |
| ContactTitle | Slicer | 1 | NVARCHAR(50) |  |
| Country | Hierarchy | 2 | NVARCHAR(25) |  |
| City | Hierarchy | 2 | NVARCHAR(25) |  |
| Phone | Drill-thought detail | 1 | NVARCHAR(30) |  |
| Fax | Drill-thought detail | 1 | NVARCHAR(30) |  |
| StartDate | Metadata |  | DATETIME2 | NOT |
| EndDate | Metadata |  | DATETIME2 |  |
| Current | Metadata |  | BIT | NOT |
| LineageKey | Metadata |  | INT | NOT |

#### Date

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN NAME | COLUMN TYPE | SCD TYPE | DATA TYPE | NULL |
| DateKey | Surrogate key |  | DATE | NOT |
| DayOfMonth | Hierarchy |  | TINYINT | NOT |
| DayOfWeek | Hierarchy |  | NVARCHAR(5) | NOT |
| DayOfWeekNumber | Sort by column |  | TINYINT | NOT |
| DayOfQuarterNumber | Drill-thought detail |  | TINYINT | NOT |
| DayOfYearNumber | Drill-thought detail |  | SMALLINT | NOT |
| Year | Hierarchy |  | SMALLINT | NOT |
| StartOfYear | Drill-thought detail |  | DATE | NOT |
| EndOfYear | Drill-thought detail |  | DATE | NOT |
| Quarter | Hierarchy |  | NVARCHAR(5) | NOT |
| YearQuarter | Hierarchy |  | NVARCHAR(10) | NOT |
| YearQuarterNumber | Sort by column |  | INT | NOT |
| StartOfQuarter | Drill-thought detail |  | DATE | NOT |
| EndOfQuarter | Drill-thought detail |  | DATE | NOT |
| Month | Hierarchy |  | NVARCHAR(10) | NOT |
| Mon | Hierarchy |  | NVARCHAR(5) | NOT |
| MonthNumber | Sort by column |  | TINYINT | NOT |
| YearMonth | Hierarchy |  | NVARCHAR(10) | NOT |
| StartOfMonth | Drill-thought detail |  | DATE | NOT |
| EndOfMonth | Drill-thought detail |  | DATE | NOT |
| Week | Hierarchy |  | NVARCHAR(50) | NOT |
| IOSWeekNumber | Hierarchy |  | TINYINT | NOT |
| StartOfWeek | Drill-thought detail |  | DATE | NOT |
| EndOfWeek | Drill-thought detail |  | DATE | NOT |
| Holiday | Drill-thought detail |  | NVARCHAR(100) | NOT |
| WorkDayType | Slicer |  | NVARCHAR(25) | NOT |
| WorkDayHours | Measure |  | TINYINT | NOT |

#### Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN NAME | COLUMN TYPE | SCD TYPE | DATA TYPE | NULL |
| EmployeeKey | Surrogate key |  | INT | NOT |
| EmployeeAlterKey | Business key |  | INT | NOT |
| Employee | Drill-thought detail | 1 | NVARCHAR(35) | NOT |
| Title | Drill-thought detail | 1 | NVARCHAR(30) |  |
| TitleOfCourtesy | Drill-thought detail | 1 | NVARCHAR(10) |  |
| City | Hierarchy | 2 | NVARCHAR(25) |  |
| Country | Hierarchy | 2 | NVARCHAR(25) |  |
| StartDate | Metadata |  | DATETIME2 | NOT |
| EndDate | Metadata |  | DATETIME2 |  |
| Current | Metadata |  | BIT | NOT |
| LineageKey | Metadata |  | INT | NOT |

#### Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN NAME | COLUMN TYPE | SCD TYPE | DATA TYPE | NULL |
| ProductKey | Surrogate key |  | INT | NOT |
| ProductAlterKey | Business key |  | INT | NOT |
| Product | Hierarchy | 1 | NVARCHAR(50) | NOT |
| Category | Hierarchy | 1 | NVARCHAR(50) | NOT |
| LineageKey | Metadata |  | INT | NOT |

### Fact

#### Order

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | COLUMN TYPE | DATA TYPE | NULL |
| OrderKey | Degenerate dimension | INT | NOT |
| ProductKey | Dimension key | INT | NOT |
| CustomerKey | Dimension key | INT |  |
| EmployeeKey | Dimension key | INT |  |
| OrderDateKey | Dimension key | DATE | NOT |
| RequiredDateKey | Dimension key | DATE |  |
| ShippedDateKey | Dimension key | DATE |  |
| UnitPrice | Measure | MONEY |  |
| Quantity | Measure | INT |  |
| Discount | Measure | MONEY |  |
| SalesAmount | Measure | MONEY |  |
| SalesAmountWithDiscount | Measure | MONEY |  |
| LineageKey | Metadata | INT |  |

## Designing Data Warehouse Physical Implementation

### Designing File Storage

|  |  |  |
| --- | --- | --- |
| Logical Drive | Filegroups | Notes |
| Disk1 | 1. PRIMARY 2. Default\_FG | * The system tables are in a dedicated filegroup to minimize fragmentation of data files. * The default filegroup is initialized with a single file of 100MB, which should be large enough for the tables it will contain for the near future. |
| Disk2 | Dimention\_Data | * Putting different tables used in the same join queries in different filegroups. This step will improve performance, because of parallel disk I/O searching for joined data. |
| Disk3 | Dimention\_Index | * Using different filegroups for heavily accessed tables and the nonclustered indexes will improve performance, because of parallel I/O if the files are located on different physical disks. |
| Disk4[[1]](#footnote-1) | Order\_Unkown\_Member\_Data | * This table is likely to grow extremely large, and so it is assigned to a dedicated logical drive to avoid competing for space. |
| Disk5 | Order\_1996\_Data | * As above |
| Disk6 | Order\_1997\_Data | * As above |
| Disk7 | Order\_Unkown\_Member\_Index | * Using different filegroups for heavily accessed tables and the nonclustered indexes will improve performance, because of parallel I/O if the files are located on different physical disks. |
| Disk8 | Order\_1996\_Index | * As above |
| Disk9 | Order\_1997\_Index | * As above |
| Disk10 | TempDB data files | * TempDB is given its own logical disk to minimize the effect of fragmentation on the data warehouse while enabling fast I/O to a suitable pre-sized TempDB. |
| Disk11 | Database log files | * The logfiles for TempDB, the Staging database, the NorthwindLogs database and the data warehouse are stored on this drive to isolate log activity from data I/O. |
| Disk12 | Backup volume | * This logical drive will be used for backup file storage |

### Partitioning

The **Order** table are partitioned on the **ShippedDateKey** column. This column was chosen because orders are not considered complete until shipped, so it is shippedto assume that the ship date is used as the cut off for each month’s orders.

Orders for the current month are partitioned into daily partitions, as this is to reflect the ETL load frequency. Orders from previous month are merged to form a single partition per month and Orders from previous years are merged to form a single partition per year, as this reduces management complexity.

### Indexes

The dimension tables each have a clustered index on the surrogate key and a non-clustered index on the alternate key, which includes SCD metadata columns where they exist. Additional non-clustered indexes are created on attribute columns basing on [dimensional models](#_Designing_Dimension_Models_1) and Database Engine Tuning Advisor.

The fact table has clustered columnstore index to maximize aggregations performance. Additional nonclustered indexes are created on dimension key columns to improve performance for joins.

### Compression

All row-stored indexes are compressed using PAGE.

Column-stored indexes compressed using COLUMNSTORE for current and previous year partitions and COLUMNSTORE\_ARCHIVE for archived data.

### Views

Each table has a matching view in the **Reports** schema. These views have user-friendly names and use the NOLOCK query hint to minimize locking.

# Designing ETL Solution

## ETL Architecture

Staging

Data Warehouse

Landing zone

Data Sources

Data Cleansing

Master Data Management

Northwind

(упрощённая)



xmlcalendar.ru

Landing



File system

bcp

bcp



DQS\_STAGING\_DATA



DQS



MDS



MDS



DWH

Facts

Dimensions

### Data Flow Diagrams

#### Dimension

##### Customer

Northwind (упрощённая)

Customers

Customer

Customer, ContactName, ContactTitle, Phone, Fax

Validate data with MDS

Extract data with BCP

Start audit

Filter modified rows

Filter on LastChgDateTime and succeeded validation

Add metadata columns

Lookup existing rows

Insert new rows (generate surrogate key)

Update SCD1 rows

Update and Insert SCD2 rows (generate surrogate key)

End audit

Data cleaning with DQS

##### Date

xmlcalendar

.\ru

Date

Validate data with MDS

Extract data from XML

Start audit

Filter modified rows

Get start and load dates from Landing.Orders

Add unknown member

Populate Date dimension from 01-01 first year to 12-31 last +1 year

End audit

Data cleaning with DQS

##### Employee

Northwind (упрощённая)

Employees

Employee

Employee, TitleOfCourtesy

Validate data with MDS

Extract data with BCP

Start audit

Filter modified rows

Filter on LastChgDateTime and succeeded validation

Add metadata columns

Lookup existing rows

Insert new rows (generate surrogate key)

Update SCD1 rows

Update and Insert SCD2 rows (generate surrogate key)

End audit

Data cleaning with DQS

##### Product

Northwind (упрощённая)

Products

Product

Product, Category

Validate data with MDS

Filter on LastChgDateTime and succeeded validation

Add metadata columns

Lookup existing rows

Insert new rows (generate surrogate key)

Update SCD1 rows

End audit

Categories

Extract data with BCP

Start audit

Filter modified rows

Data cleaning with DQS

#### Fact

##### Order

Northwind (упрощённая)

Orders

Update Partition Schema

Create load table

Start audit

Order

End audit

Order Details

Create filegroups

Set start load date

Switch partition

Insert or update data for previous dates

Drop load table

Extract data with bcp

Extract data with bcp

Create Hash

Extract data using hash

Lookup Customer Key

Lookup Employee Key

Lookup Product Key

Customer

Employee

Product

Calculate Sales Amount & Sales Amount with Discount

Replace NULLs in dimensions keys with unknown dimensions members

Optimize partitions Monthly

Optimize partitions Yearly

### Source To Target Mapping

[Source to target mapping.xlsx](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/Docs/Source%20to%20target%20mapping.xlsx&version=GBmaster&_a=contents)

## Planning Data Extraction

### Data source profiling

Table 1. Data sources.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Description | Profiling | Auth type | Account |
| Northwind (упрощённая) | SQL Server | Transactional data about customers, employees, products, orders | [Northwind Data Profile.xml](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/Docs/Northwind%20Data%20Profile.xml)[[2]](#footnote-2) | SQL login | Зинуков Денис Витальевич |
| [xmlcalendar.ru](http://xmlcalendar.ru/) | XML | Information about holidays in Russia | [xmlcalendar/data (github.com)](https://github.com/xmlcalendar/data) | Anonymous |  |

### Identifying New and Modified Rows

Northwind (упрощённая) do not have Data modification time fields, so hash values must be used to look for changes.

Hash values must be calculated using columns that used in Data Warehouse.

Table 8. Landing.Customers.

|  |  |
| --- | --- |
| Column | Used in Hash |
| CustomerID |  |
| CompanyName | + |
| ContactName | + |
| ContactTitle | + |
| City | + |
| Country | + |
| Phone | + |
| Fax | + |

Table 9. Landing.Products.

|  |  |
| --- | --- |
| Column | Used in Hash |
| ProductID |  |
| ProductName | + |
| SupplierID |  |
| CategoryID | + |
| UnitPrice |  |

Table 10. Landing.Categories.

|  |  |
| --- | --- |
| Column | Used in Hash |
| CategoryID |  |
| CategoryName | + |
| Description |  |

Table 11. Landing.Employees.

|  |  |
| --- | --- |
| Column | Used in Hash |
| EmployeeID |  |
| LastName | + |
| FirstName | + |
| Title | + |
| TitleOfCourtesy | + |
| City | + |
| Country | + |

Table 12. Landing.Orders.

|  |  |
| --- | --- |
| Column | Used in Hash |
| OrderID |  |
| CustomerID | + |
| EmployeeID | + |
| OrderDate | + |
| RequiredDate | + |
| ShippedDate | + |
| ShipCity |  |
| ShipCountry |  |

Table 13. Landing.Order Details.

|  |  |
| --- | --- |
| Column | Used in Hash |
| OrderID |  |
| ProductID |  |
| UnitPrice | + |
| Quantity | + |
| Discount | + |

### Planning Extraction Windows

Table 14. Planning extraction.

|  |  |  |
| --- | --- | --- |
|  | Northwind (Упрощённая) | xmlcalendar.ru |
| How frequently is new data generated in the source systems, and for how long is it retained? | Every day, permanently | Once or twice a year, permanently |
| What latency between changes in source systems and reporting is tolerable? | One day | |
| How long does data extraction take? | Less 1 minute | Less 1 minute |
| During what time periods are source systems least heavily used? | Workdays from 09:00:00 to 17:00:00 | No |

Table 15. Extraction Windows.

|  |  |
| --- | --- |
| Data source | Extraction windows |
| Northwind (Упрощённая) | Workdays from 17:00:00 to 09:00:00 and weekend all day |
| xmlcalendar.ru | All day |

## Planning Data Transformation

### Transact-SQL vs. Data Flow Transformations

Inside one database it is used Stored procedures for transformations.

When it is needed to transfer data from one database to another or combine data from different databases it is used data flows.

### Handling Invalid Rows and Errors

Validation rules are implemented with DQS and MDS for dimensions data. Invalid rows are filtered on extraction from MDS.

For invalid dimension keys in fact tables is used unknown member surrogate key for dimension.

Event handling OnError:

* 1. Get parameter values
  2. Execute Error notification package
     1. Write event information to **[Integration].[ErrorLog]** table[[3]](#footnote-3)
     2. Send E-mail notification

### Logging Audit Information

Every table in Data Warehouse has foreign key to **[Integration].[Lineage]** table for auditing ETL.

Logging:

1. SSIS logs to Logs DB and txt-file. Events:
   1. OnError
   2. OnTaskFailed
   3. OnVariableValueChanged
   4. DiagnosticEx

## Planning Data Loads

### Minimizing Logging

1. Set recovery model simple for:
   1. Data Warehouse
   2. NorthwindLogs
   3. Landing
   4. DQS\_STAGING\_DATA
2. Use a bulk load operation to insert data:
   1. SSIS data flow destinations and the fast load option
   2. The bulk copy program (BCP)
   3. The BULK INSERT statement
   4. The INSERT … SELECT Statement
   5. The SELECT INTO Statement
   6. The MERGE Statement

### Loading Indexed Tables

1. Consider dropping and recreating indexes for large volumes of new data.

|  |  |
| --- | --- |
| Indexes | New data relative to existing table size |
| Clustered index only | 30% |
| Clustered index plus one non-clustered index | 25% |
| Clustered index plus two non-clustered indexes | 25% |
| Single non-clustered index only | 100% |
| Two non-clustered indexes | 60% |

1. Sort data by the clustering key and specify the ORDER hint

### Loading Partitioned Fact Tables

1. Switch loaded tables into partitions
2. The MERGE Statement for backdated data
3. Partition-align indexed view

# Planning BI Delivery Solution

Power BI usage scenarios: [On-premises reporting](https://learn.microsoft.com/en-us/power-bi/guidance/powerbi-implementation-planning-usage-scenario-on-premises-reporting)

## Reporting Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Audience | Tool | Notes |
| See sales amount with discount by customers and products | Sales VP | Power BI |  |
| Compare sales amount with discount with previous year/quarter/month | CEO, Sales VP | Power BI | [Standard time-related calculations](https://www.daxpatterns.com/standard-time-related-calculations/)  Additionally implement calculated group with context of current period and construction to see previous time periods in context of selected period. |
| See discount by employees and their geography | Sales Representative | Power BI |  |
| See new and returned customers | Sales VP | Power BI | [New and returning customers](https://www.daxpatterns.com/new-and-returning-customers/)   * Absolute.   A customer is considered new the first time they buy a product, regardless of any filter present in the report. |
| Cluster customers based on spending volume | Sales VP | Power BI | [Dynamic segmentation](https://www.daxpatterns.com/dynamic-segmentation/)   * Basic pattern.   2 variants based on Sales amount and Sales amount with Discount. |
| Product ABC classification | Sales VP | Power BI | [ABC classification](https://www.daxpatterns.com/abc-classification/)   * Dynamic ABC classification. * Finding the ABC class. |
| Product basket analysis | Inside Sales Coordinator | Power BI | [Basket analysis](https://www.daxpatterns.com/basket-analysis/)   * Optimized pattern |
| See number of orders, sales amount, sales amount with discount in progress | Inside Sales Coordinator | Power BI | [Events in progress](https://www.daxpatterns.com/events-in-progress/)   * Open orders with snapshot |
| Receive email notifications about errors in the ETL process | Database administrators team | Paginated report |  |
| Notification of errors and raw values in data cleansing processes | Data stewards | Paginated report |  |

# Monitoring and Optimizing BI Solution

## Creating a Performance Baseline

Before applying an effective health monitoring process, it is necessary to have a set of resource utilization metrics with which to compare future measurements. It is necessary to collect baseline resource utilization metrics for each server or service in the BI solution and collect the data over a sufficient period to reflect normal workload cycles. These workload cycles should include:

* Data warehouse load operations.
* Power BI reports processing operations.
* Normal everyday query activity.
* Peak-time query activity.

## Monitoring ETL

### Package Execution Reports

SSIS catalog reports:

1. **Integration Services Dashboard**. This report provides a central summary that shows details of package executions. For each package execution listed in this report, you can drill into three sub-reports: Overview, All Messages, and Execution Performance.
2. **All Executions.** This report provides details of all package executions on the server, and you can filter it to show executions within a specified date range.
3. **All Connections.** This report shows details of all connections that the package used during executions, including connection strings and whether the connection failed or succeeded.
4. **All Operations.** This report shows details of all operations that occurred on the server, including package deployments, executions, and other administrative operations.
5. **All Validations.** This report shows details of all validations that SSIS performed for packages.

[SSRS project Monitoring](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindPBIRS/Monitoring) reports:

1. **Monitoring**. This report provides information about loading duration, detailed information about Data Warehouse partitions, row validating statuses in DQS and MDS and errors in ETL process.
2. **SSIS log**. This report provides visual representation of sysssislog table.

## Monitoring and optimizing Data Warehouse

### Considerations for Data Warehouse Performance

#### Data warehouse workloads:

* **ETL data loads.** These workloads are usually regularly scheduled events that load new or updated data into the tables in the data warehouse.
* **Power BI reports data models processing.** These workloads usually occur after each ETL data load, to refresh Power BI reports data models with the new data.
* **Report queries.** These workloads occur when users create a report or a user-defined data model in Excel that requires the retrieval of data is retrieved from data warehouse tables.
* **Operational activities.** These workloads are usually scheduled operations, such as index maintenance or database backups.

#### SQL Server settings

SQL Server provides various configuration settings that affect the way the service uses hardware resources. In particular, can be used the Maximum Server Memory configuration setting to specify the maximum amount of system memory that SQL Server can use, and can be used the ALTER SERVER CONFIGURATION statement to control how SQL Server uses CPU threads and simulate non-uniform memory access (NUMA).

### Monitoring tools

* [Performance Monitor](https://learn.microsoft.com/en-us/sql/relational-databases/performance-monitor/monitor-resource-usage-system-monitor?view=sql-server-ver15)
* [SQL Server Profiler](https://learn.microsoft.com/en-us/sql/relational-databases/sql-trace/sql-trace?view=sql-server-ver15)
* [Data Collector](https://learn.microsoft.com/en-us/sql/relational-databases/data-collection/data-collection?view=sql-server-ver15)
* [Extended Events](https://learn.microsoft.com/en-us/sql/relational-databases/extended-events/extended-events?view=sql-server-ver15)

### Turning tools

* [Database Engine Tuning Advisor](https://learn.microsoft.com/en-us/sql/relational-databases/performance/database-engine-tuning-advisor?view=sql-server-ver15)
* [Query Store](https://learn.microsoft.com/en-us/sql/relational-databases/performance/monitoring-performance-by-using-the-query-store?view=sql-server-ver15)

### Using Resource Governor to Balance Resource Utilization

Table 16. SQL Server resource pools

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Minimum CPU % | Maximum CPU % | Minimum Memory % | Maximum memory % | Minimum I/O operations per second (IOPS) per disk volume | Maximum I/O operations per second (IOPS) per disk volume |
| Low Priority | 0 | 50 | 0 | 50 | 0 | 65 536 |
| High Priority | 20 | 90 | 20 | 90 | 65 536 | 0 |

Table 17. SQL Server workload groups

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Resource pool name | Importance | Maximum Requests | CPU Time (sec) | Memory Grant % | Memory Grant Time-out (sec) | Degree of Parallelism |
| User Queries | Low Priority | Low | 10 | 50 | 50 | 20 | 1 |
| ETL | High Priority | High | 100 | 80 | 80 | 30 | 4 |

Conditions for dbo.fn\_classify:

IF USER\_NAME IN

AzPipelineAgent

SQLAGENT

RDLexec

THEN Workload group “ETL”

ELSE Workload group “User Queries”

### Troubleshooting Data Warehouse Performance

1. Performance monitor counters ([Performance Monitor Template](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/Monitoring/Performance%20Monitoring.xml&version=GBmaster))[[4]](#footnote-4):
   1. Memory: Available MBytes
   2. Paging File: % Usage
   3. Physical Disk:Avg. Disk sec/Read
   4. Physical Disk: Avg. Disk sec/Write
   5. Physical Disk: Disk Reads/sec
   6. Physical Disk: Disk Writes/sec
   7. Processor: % Processor Time
   8. SQLServer:Buffer Manager: Buffer cache hit ratio
   9. SQLServer:Buffer Manager: Page life expectancy
   10. SQLServer:General Statistics: User connections
   11. SQLServer:Memory Manager: Memory grants pending
   12. SQLServer:SQL Statistics: Batch requests/sec
   13. SQLServer:SQL Statistics: Compilations/sec
   14. SQLServer:SQL Statistics: Recompilations/sec
   15. System: Processor queue length
   16. SQLServer:Resource Pool Stats: CPU control effect %
   17. SQLServer:Workload Group Stats: CPU usage %
   18. SQLServer:Workload Group Stats: Reduced memory grants/sec
2. SQL Server Profiler ([SQL Server Profiler Template](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/Monitoring/Monitor%20Data%20Warehouse%20Query%20Activity.tdf&version=GBmaster)):
   1. events
      1. SQL: BatchCompleted
      2. SQL: StmtCompleted
   2. columns
      1. ApplicationName
      2. DatabaseName
      3. Duration
      4. EndTime
      5. LoginName
      6. Reads
      7. RowCounts
      8. SPID
      9. StartTime
      10. TextData
   3. filters
      1. DatabaseName LIKE %NorthwindDW%

## Monitoring DQS

The **Activity Monitoring** screen in Data Quality Client displays detailed information about each activity performed on a Data Quality Server. Enables the DQS administrators to control an activity or a process within an activity by terminating a running activity or stopping a running process within an activity, if required. The data is displayed for knowledge discovery, domain management, matching policy, cleansing, matching, and SQL Server Integration Services (SSIS)-based cleansing.

## Monitoring MDS

[Tracing - SQL Server Master Data Services](https://learn.microsoft.com/en-us/sql/master-data-services/tracing-master-data-services?view=sql-server-ver15)

## Monitoring and optimizing Power BI Report Server

### Considerations for Reporting Services Performance

#### Reporting Services workloads

* **Data retrieval and processing.** When responding to a request for a report (interactively or by a scheduled subscription, cache refresh, or snapshot), Reporting Services loads the definition of the report and uses its datasets to retrieve the required data. Reporting Services then generates the report in an intermediate format for rendering.
* **Report rendering.** When a user views or exports a report, or Reporting Services will deliver a report as a subscription, Reporting Services will use the appropriate rendering extension to render the report into the required format.

#### Reporting Services settings

* **WorkingSetMinimum.** You can add this setting to the RSReportServer.config file to specify the minimum amount of system memory that Reporting Services must have allocated before it will start to release memory resources. By default, this setting is 60 percent of the memory available on the server. If Reporting Services is within this value, the level of memory pressure is considered low.
* **MemorySafetyMargin.** You can use this value to specify a percentage of WorkingSetMaximum. If Reporting Services exceeds this amount of memory, the level of memory pressure is considered medium, and Reporting Services begins to refuse some requests for memory reduce memory allocations.
* **MemoryThreshold.** You can use this value to specify a percentage of WorkingSetMaximum that is higher than MemorySafetyMargin. If Reporting Services exceeds this amount of memory, the level of memory pressure is considered high, and Reporting Services begins to manage requests for memory aggressively.
* **WorkingSetMaximum.** You can add this setting to the RSReportServer.config file to set the maximum amount of system memory that Reporting Services can use. By default, this setting is not included in the RSReportServer.config file, and Reporting Services can access all memory available on the server.

### Monitoring Tools

* [Power BI Desktop](https://learn.microsoft.com/en-us/power-bi/guidance/monitor-report-performance)
* [Power BI report server logs](https://learn.microsoft.com/en-us/sql/reporting-services/report-server/reporting-services-log-files-and-sources?view=sql-server-ver15)

### Tune

* [Optimization guide for Power BI](https://learn.microsoft.com/en-us/power-bi/guidance/power-bi-optimization)

### Using Caching and Snapshots to Optimize Performance

Consider the following factors:

* Reporting Services stores cached reports in an intermediate format that includes data and layout information. It is still necessary to render the report to the requested format after recovering it from cache.
* Cached datasets and reports are based on specific parameter value combinations. Reporting Services creates a cached copy of the report or dataset for each combination of parameter values requested.
* Can be configured a cached object to expire after a specified interval (in minutes), or at a time specified in a schedule. Schedules can be specific to an individual cached object or shared across multiple objects. When a cached object expires, Reporting Services removes it from the cache and the next request results in a new execution with live data.
* Can be preloaded a cached object by creating a cache refresh schedule, or by scheduling a subscription for a cached report with a NULL delivery extension.
* In the event of a server restart, Reporting Services recreates cached objects.

# Operating BI Solution

## ETL Operations

### Deploying and Configuring Packages

Table 18. Deployment model.

|  |  |
| --- | --- |
| Deployment Model | Project |
| Storage | SSIS catalog |
| Dynamic configuration | Maps environment variables in the SSIS catalog to project-level parameters and connection managers. |
| Troubleshooting | Logs events are and saves them to the catalog automatically. It can be viewed by using built-in reports, reports in [SSRS project Monitoring](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindPBIRS/Monitoring), txt file in file system, views such as **catalog.executions** and **catalog.event\_messages** and tables in database NorthwindLogs dbo. Lineage. |

### Environments

To change execution context in straightforward way it is recommended to create environment with variables mapped to parameters and to connection manager properties.

* Script for creating [CreateEnvironment.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateEnvironment.sql&version=GBmaster&_a=contents).
* Script for mapping [SetEnvironmentVars.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/SetEnvironmentVars.sql&version=GBmaster&_a=contents)

### Timing of package execution

At weekends nights ETL will be longer because of maintenance operations:

* Every night from Saturday to Sunday creates full database backup
* Every first night in month from Friday to Saturday merges daily partitions in monthly. Additionally, every January merges monthly partitions in yearly.

### Execution account

**SQL Server agent account** on server with SQL Server integration services and SSIS catalog.

### Backing up the SSIS Catalog

Considerations for managing the SSIS catalog:

1. Back up SSIS catalog[[5]](#footnote-5)
2. Back up the Master key
3. Generate scripts to recreate:
   1. The ##MS\_SSISServerCleanupJobLogin## login.
   2. The dbo.sp\_ssis\_startup stored procedure.
   3. The SSIS Server Maintenance Job SQL Server Agent job.

## Data Warehouse Operations

### Reorganizing or rebuilding indexes[[6]](#footnote-6)

Reorganizing or rebuilding indexes uses for partitions in read-write filegroups. Implemented in stored procedure [OptimizeIndexes.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Maintenance/Store%20Procedures/OptimizeIndexes.sql) with the following logic.

IF avg\_fragmentation\_in\_percent between 5 and 30 THEN reorganize

IF avg\_fragmentation\_in\_percent > 30 THEN rebuild

### Updating data distribution statistics[[7]](#footnote-7)

Uses the **sp\_updatestats** stored procedure.

### Managing partitioned tables

The manage partitions for each fact table occurs in the fact load package.

Steps:

1. Every day:
   1. Create filegroup if does not exist.
   2. Create partition for new portion of data.
2. Every first night from Friday to Saturday in month merges day partitions for the previous month in 1 month partition.
3. Every first night from Friday to Saturday in year merges month partitions for the previous year in 1 year partition.

### Managing database files

Data Warehouse will create new filegroups and files for fact data in default file system folder. DBA must use stored procedure **[Maintenance].[InsertDatabaseFilesData]** to point location for new database files. The information about partitions, filegroups and files places in tables [dbo].[DatabaseFiles] in NorthwindLogs DB and [Maintenance].[DatabaseFiles] in Data Warehouse.

### Security

Roles:

* The **UserBI** role can select data from schema Reports.

### Backing up the database

It is believed that the data of the year before last will no longer change.

Partial backup strategy:

1. Changes compression to COLUMNSTORE\_ARCHIVE for partitions with fact data and nonclustered indexes of the year before last.
2. Sets filegroups with fact data and nonclustered indexes of the year before last READONLY.
3. Every night creates READONLY filegroups backup if it was not done.
4. Every night from Saturday to Sunday creates full backup for READ\_WRITE\_FILEGROUPS.
5. Every night but from Saturday to Sunday creates differential backup for READ\_WRITE\_FILEGROUPS.

## DQS Operations

### Configuration

* **General Settings**: Specify the threshold values for data cleansing and data matching, and whether to enable notifications for profiling in Data Quality Client. These threshold values are used by DQS during the computer-assisted cleansing and matching activities in a data quality project.
* **Log Settings**: The log files in DQS record the activities performed in DQS and are useful for tracking operational issues during maintenance and troubleshooting. You can filter the messages that you want to be logged for various DQS features (domain management, knowledge discovery, cleansing, matching, and reference data services) and DQS modules based on the severity level of the events.

### Security

Roles:

* The **DQS Administrator** (dqs\_administrator role) can do everything in the scope of the product. The administrator can edit and execute a project, create and edit a knowledge base, terminate an activity, stop a process within an activity, and can change the configuration and Reference Data Services settings. The DQS Administrator cannot, however, install the server or add new users. The database administrator must do that.
* The **DQS KB Editor** (dqs\_kb\_editor role) can perform all of the DQS activities, except for administration. The KB Editor can edit and execute a project, and create and edit a knowledge base. They can see the activity monitoring data, but cannot terminate or stop an activity or perform administrative duties.
* The **DQS KB Operator** (dqs\_kb\_operator role) can edit and execute a project. They cannot perform any kind of knowledge management; they cannot create or change a knowledge base. They can see the activity monitoring data, but cannot terminate an activity or perform administrative duties.

The database administrator (DBA) creates DQS users and associates them with DQS roles in SQL Server Management Studio. The DBA manages their permissions by adding SQL Logins as users of the DQS\_MAIN database and associating each user with one of the DQS roles. Each role is granted permissions to a set of stored procedures on the DQS\_MAIN database. The three DQS roles are not available for the DQS\_PROJECTS and DQS\_STAGING\_DATA databases.

### Backing up DQS databases

Considerations for managing the DQS catalogs:

* Back up DQS\_MAIN[[8]](#footnote-8)
* Back up DQS\_PROJECTS[[9]](#footnote-9)
* [Export Knowledge Bases to .dqsb File](https://learn.microsoft.com/en-us/sql/data-quality-services/install-windows/export-and-import-dqs-knowledge-bases-using-dqsinstaller-exe?view=sql-server-ver15#export)
* Generate scripts to recreate:
  + The ##MS\_dqs\_db\_owner\_login## login
  + The ##MS\_dqs\_service\_login## login
  + The DQInitDQS\_MAIN stored procedure in the master database

## MDS Operations

### Security

Types of Users:

* Those who access data in the Explorer functional area.
* Those who have the ability to perform administrative tasks in areas other than Explorer.

[Security - SQL Server Master Data Services](https://learn.microsoft.com/en-us/sql/master-data-services/security-master-data-services?view=sql-server-ver15)

Additionally, in MDS database created roles:

* The VSTS\_AgentService\_G39071 role can select and execute in schema stg. This role is necessary for ETL.
* The RDLexec role can select from master data views and schema stg. This role is necessary for monitoring reports in Power BI Report Server.

### Backing up MDS database

Considerations for managing the DQS catalog:

* Back up MDS database[[10]](#footnote-10)
* [Create Model Deployment Packages for each model](https://learn.microsoft.com/en-us/sql/master-data-services/create-a-model-deployment-package-by-using-mdsmodeldeploy?view=sql-server-ver15)

### Cleaning up transactions and staging tables

Cleaning up transactions and staging tables implemented with stored procedures in Data Warehouse and automated with DWH maintenance package. Transactions and data in staging tables older than 7 days are cleared by default.

## Reporting Services Operations

### Managing Scheduled Tasks

Considerations for schedules:

* Schedules depend on the SQL Server Agent. If SQL Server Agent is not running, scheduled tasks will not run.
* Using shared schedules enables you to centrally pause, resume, and modify multiple scheduled tasks in a specific location. Using object-specific schedules requires managing each schedule individually.
* A report server uses the time zone of the computer that hosts it.
* If you change the time zone of a server that hosts Power BI Report Server, you must restart the Reporting Services service for the time zone change to take effect. When you change the time zone of a report server, existing schedules retain the same times in the new time zone. For example, a task that was scheduled to run at 2:00 in the old time zone will be scheduled to run at 2:00 in the new time zone.

### Backing up Power BI Report Server

Considerations for managing the Report Server catalog:

* Back up Encryption key
* Back up ReportServer database
* Back up Configuration files
  + Rsreportserver.config
  + Rssvrpolicy.config
  + Web.config

## Deployment

### Solution structure

|  |  |  |
| --- | --- | --- |
| Project name | Project type | Description |
| Landing | Database | Landing database. Stores data from data sources as is. |
| DQS\_STAGING\_DATA | Database | Staging database. Stores data after DQS cleansing. |
| NorthwindLogs | Database | Database used for storing SSIS logs and Data Warehouse metadata. |
| Monitoring | SSRS | Paginated reports for monitoring. |
| NorthwindDW | Database | Data Warehouse database. |
| NorthwindETL | SSIS | ETL |

### Parameters

Table 19. NorthwindETL parameters.

|  |  |  |
| --- | --- | --- |
| Project parameter | Data Type | Description |
| BackupFilesPath | String | Path for backup files. Used in DWH maintenance package to create backup for Data Warehouse. |
| DBFilesPath | String | Default path for new database files which are created in fact load packages. |
| DQSDatabaseName | String | Name of staging database where writing data after DQS cleansing. |
| DQSServerName | String | Name of server with staging database where writing data after DQS cleansing. |
| DWHDatabaseName | String | Name of Data Warehouse database. |
| DWHServerName | String | Name of Data Warehouse server. |
| EndLoadDate | String | Parameter for testing incremental load |
| ExternalFilesPath | String | Path for files needed for ETL process:   1. Data files (bcp) 2. Logs in txt format 3. Dimensional rows with no change 4. Maintenance SQL scripts |
| LandingDatabaseName | String | Name of database used for raw data from data sources. |
| LandingServerName | String | Name of server used for raw data from data sources. |
| LogsDatabaseName | String | Name of database used for storing SSIS logs and Data Warehouse metadata |
| LogsServerName | String | Name of server with database used for storing SSIS logs and Data Warehouse metadata |
| MDSDatabaseName | String | Name of database used for Master data services catalog. |
| MDSServerName | String | Name of server with database used for Master data services. |
| RetrainWeeks | Int16 | Storage depth for Data Warehouse backups in weeks. |
| XMLCalendarFolder | String | Path to local repository of xmlcalendar project. |

Table 20. NorthwindDW parameters.

|  |  |  |
| --- | --- | --- |
| Project parameter | Data Type | Description |
| DQSDatabaseName | String | Name of staging database where writing data after DQS cleansing. |
| DQSServerName | String | Name of server with staging database where writing data after DQS cleansing. |
| DomainName | String | Name of Data Warehouse database. |
| LogsDatabaseName | String | Name of database used for storing SSIS logs and Data Warehouse metadata |
| LogsServerName | String | Name of server with database used for storing SSIS logs and Data Warehouse metadata |
| MDSDatabaseName | String | Name of database used for Master data services. |
| MDSServerName | String | Name of server with database used for Master data services. |

### Manual deployment

1. Build solution NorthwindBI.sln.
2. Install DQS Server.
3. Install MDS.
4. Deploy DQS\_STAGING\_DATA project on DQS server.
5. Deploy NorthwindLogs on Logs server.
6. Deploy NorthwindDW on DWH server.
7. Move database files using information from [Designing File Storage](#_Designing_File_Storage).
8. Import DQS knowledge bases ([docs](https://learn.microsoft.com/en-us/sql/data-quality-services/install-windows/export-and-import-dqs-knowledge-bases-using-dqsinstaller-exe?view=sql-server-ver15#import), [docs](https://learn.microsoft.com/en-us/sql/data-quality-services/import-a-knowledge-base-from-a-dqs-file?view=sql-server-ver15), [archive](https://1drv.ms/u/s!AueE3aY8CojZlNQyCm6apJXVbjh38g?e=rfxCBC)).
9. Import MDS models ([docs](https://learn.microsoft.com/en-us/sql/master-data-services/deploy-a-model-deployment-package-by-using-the-wizard?view=sql-server-ver15), [archive](https://1drv.ms/u/s!AueE3aY8CojZlNQxwPqrzkTIGLwWJQ?e=nHHWkN)).
10. Create SSIS environment ([CreateEnvironment.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateEnvironment.sql&version=GBmaster&_a=contents)).
11. Deploy NorthwindETL on ETL server.
12. Map SSIS environment variables with parameters and connection manager properties ([SetEnvironmentVars.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/SetEnvironmentVars.sql)).
13. Create roles in MDS database ([CreateRoles.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateRoles.sql)).
14. Create users in databases ([CreateUsers.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateUsers.sql)).
15. Copy script [SetFilegroupsReadOnly.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/SetFilegroupsReadOnly.sql) to file system location $(ExternalFilesPath[[11]](#footnote-11))\Scripts
16. Deploy Monitoring to Power BI report server or SQL Server reporting services.
17. Deploy Power BI reports.

Also, the were created build and release pipelines in Azure DevOps. All necessary cmd scripts are under the [link](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/cmd).

### Manual testing

1. Run [ExecuteTestPackage.bat](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/cmd/ExecuteTestPackage.bat)
2. Open [Package Execution Reports](#_Package_Execution_Reports_1) to check results

## Automating Operational Tasks

### SQL agent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Job | Steps | Description | Environment | Schedule | Notifications |
| Northwind (упрощённая) | Northwind (упрощённая) Package | Ingest data from Northwind (упрощённая) | Release | 01:00:00 AM every day | E-mail to DBA team |
| Xmlcalendar | Xmlcalendar Package | Ingest data from xmlcalendar | Release | 01:00:00 AM on Sunday | E-mail to DBA team |
| Incremental load | Incremental load Package | Load new portion of data | Release | 02:00:00 AM every day | E-mail to DBA team |

### Subsystems maintenance plan

#### Full backup

Steps:

1. Check database integrity task
   1. Databases:
      1. DQS\_MAIN
      2. DQS\_PROJECTS
      3. DQS\_STAGING\_DATA
      4. NorthwindLogs
      5. master
      6. MDS
      7. msdb
      8. ReportServer
      9. ReportServerTempDB
      10. SSISDB
   2. Include indexes
   3. Physical only
2. Backup database task
   1. Databases:
      1. DQS\_MAIN
      2. DQS\_PROJECTS
      3. DQS\_STAGING\_DATA
      4. NorthwindLogs
      5. master
      6. MDS
      7. msdb
      8. ReportServer
      9. SSISDB
   2. Type: Full
   3. Backup set will expire after 14 days
   4. Destination: Disk
   5. Compression: on
3. Maintenance cleanup task
   1. Cleanup Database Backup files
   2. Age: older than 4 weeks

Schedule:

1. Occurs every week on Sunday at 1:00:00.

#### Differential backup

Steps:

1. Check database integrity task
   1. Databases:
      1. DQS\_MAIN
      2. DQS\_PROJECTS
      3. DQS\_STAGING\_DATA
      4. NorthwindLogs
      5. master
      6. MDS
      7. msdb
      8. ReportServer
      9. ReportServerTempDB
      10. SSISDB
   2. Include indexes
   3. Physical only
2. Backup database task
   1. Databases:
      1. DQS\_MAIN
      2. DQS\_PROJECTS
      3. DQS\_STAGING\_DATA
      4. NorthwindLogs
      5. MDS
      6. msdb
      7. ReportServer
      8. SSISDB
   2. Type: Differential
   3. Backup set will expire after 14 days
   4. Destination: Disk
   5. Compression: on

Schedule:

1. Occurs every day every 12 hour(s) between 0:00:00 and 23:59:59.

#### Log backup

Steps:

1. Backup database task
   1. Databases:
      1. MDS
      2. ReportServer
      3. SSISDB
   2. Type: Transaction log
   3. Backup set will expire after 14 days
   4. Destination: Disk
   5. Compression: on

Schedule:

1. Occurs every day every 30 minute(s) between 0:00:00 and 23:59:59.

## Disaster recovery plan

1. Installation
   1. Install SQL server 2019 using [Configuration file](https://1drv.ms/u/s!AueE3aY8CojZlMppqryUZmonKVHj-Q?e=1vriaj)
   2. Install DQS Server
   3. Install MDS
   4. Install Power BI Report Server
2. Restore DQS using [Backing Up and Restoring DQS Databases](https://learn.microsoft.com/en-us/sql/data-quality-services/backing-up-and-restoring-dqs-databases?view=sql-server-ver15#BackupRestore)
3. Restore MDS using [SQL Server Master Data Services (MDS) Database Restore Steps](https://www.mssqltips.com/sqlservertip/4295/sql-server-master-data-services-mds-database-restore-steps/)
4. Restore SSIS using [Moving the SSISDB Catalog on a new SQL Server instance](https://www.sqlshack.com/moving-the-ssisdb-catalog-on-a-new-sql-server-instance/)
5. Restore Report Server:
   1. Restore ReportServer database
   2. Create ReportServerTempDB database
   3. Run the CatalogTempDB.sql script in the \Program Files\Microsoft Power BI Report Server\PBIRS\ReportServer folder
   4. Restore Report server Encryption Keys using [Restore encryption keys - Report Server Configuration Manager](https://learn.microsoft.com/en-us/sql/reporting-services/install-windows/ssrs-encryption-keys-back-up-and-restore-encryption-keys?view=sql-server-ver15#bkmk_restore_configuration_manager)
   5. Restore Report server configuration files in the \Program Files\Microsoft Power BI Report Server\PBIRS\ReportServer folder
6. Restore NorthwindLogs database
7. Restore Landing database
8. Restore DQS\_STAGING\_DATA database
9. Restore msdb database
10. Restore Data Warehouse with script [RestoreDatabase.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/RestoreDatabase.sql)
11. Create users with script [CreateUsers.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateUsers.sql)

1. For every fact partition is needed dedicated disk. [↑](#footnote-ref-1)
2. For view xml file is needed [Data Profile Viewer](https://learn.microsoft.com/en-us/sql/integration-services/control-flow/data-profile-viewer?view=sql-server-ver15) [↑](#footnote-ref-2)
3. There is information from [Integration].[ErrorLog] in monitoring.rdl. [↑](#footnote-ref-3)
4. Additional information about performance counters is under the link [BI performance counters (sqlshack.com)](https://www.sqlshack.com/bi-performance-counters/). [↑](#footnote-ref-4)
5. Can be automated using the [maintenance plan](#_Subsystems_maintenance_plan) shown below. [↑](#footnote-ref-5)
6. This operation is implemented in DWH maintenance package. [↑](#footnote-ref-6)
7. This operation is implemented in DWH maintenance package. [↑](#footnote-ref-7)
8. Can be automated using the [maintenance plan](#_Subsystems_maintenance_plan) shown below. [↑](#footnote-ref-8)
9. Can be automated using the [maintenance plan](#_Subsystems_maintenance_plan) shown below. [↑](#footnote-ref-9)
10. Can be automated using the [maintenance plan](#_Subsystems_maintenance_plan) shown below. [↑](#footnote-ref-10)
11. Table 11. NorthwindETL parameters. [↑](#footnote-ref-11)