Northwind BI solution

Оглавление

[Planning a BI Solution 4](#_Toc129364922)

[Project Scope 4](#_Toc129364923)

[Required BI Software 4](#_Toc129364924)

[Project conceptual schema 5](#_Toc129364925)

[Project Infrastructure 6](#_Toc129364926)

[Project Personnel Roles 6](#_Toc129364927)

[Business Stakeholders 6](#_Toc129364928)

[Planning SQL Server Business Intelligence Infrastructure 7](#_Toc129364929)

[BI Topology 7](#_Toc129364930)

[Single BI server 7](#_Toc129364931)

[Dedicated report server 8](#_Toc129364932)

[Dedicated ETL server 9](#_Toc129364933)

[Dedicated master data server 10](#_Toc129364934)

[Distributed report server 11](#_Toc129364935)

[Planning Data Warehouse Hardware 12](#_Toc129364936)

[Designing Data Warehouse 13](#_Toc129364937)

[Designing Data Warehouse Logical Schema 13](#_Toc129364938)

[Identify business processes 13](#_Toc129364939)

[Designing Dimension Models and Data Warehouse Tables 14](#_Toc129364940)

[Data source profiling 15](#_Toc129364941)

[Dimension 17](#_Toc129364942)

[Fact 18](#_Toc129364943)

[Designing Data Warehouse Physical Implementation 18](#_Toc129364944)

[Designing File Storage 18](#_Toc129364945)

[Partitioning 19](#_Toc129364946)

[Indexes 19](#_Toc129364947)

[Compression 19](#_Toc129364948)

[Views 19](#_Toc129364949)

[Designing ETL Solution 20](#_Toc129364950)

[Xmlcalendar 20](#_Toc129364951)

[Incremental load package 20](#_Toc129364952)

[Customer 21](#_Toc129364953)

[Date 22](#_Toc129364954)

[Employee 23](#_Toc129364955)

[Product 24](#_Toc129364956)

[Order 25](#_Toc129364957)

[Data Warehouse maintenance 26](#_Toc129364958)

[Planning BI Delivery Solution 27](#_Toc129364959)

[Reporting Requirements 27](#_Toc129364960)

[Monitoring and Optimizing BI Solution 28](#_Toc129364961)

[Creating a Performance Baseline 28](#_Toc129364962)

[Monitoring and optimizing Data Warehouse 28](#_Toc129364963)

[Considerations for Data Warehouse Performance 28](#_Toc129364964)

[Monitoring tools 28](#_Toc129364965)

[Turning tools 28](#_Toc129364966)

[Using Resource Governor to Balance Resource Utilization 28](#_Toc129364967)

[Troubleshooting Data Warehouse Performance 29](#_Toc129364968)

[Monitoring and optimizing Power BI Report Server 30](#_Toc129364969)

[Considerations for Reporting Services Performance 30](#_Toc129364970)

[Monitoring Tools 30](#_Toc129364971)

[Tune 30](#_Toc129364972)

[Using Caching and Snapshots to Optimize Performance 30](#_Toc129364973)

[Operating BI Solution 31](#_Toc129364974)

[ETL Operations 31](#_Toc129364975)

[Deploying and Configuring Packages 31](#_Toc129364976)

[Environments 31](#_Toc129364977)

[Considerations for Package Execution 31](#_Toc129364978)

[Package Execution Reports 31](#_Toc129364979)

[Backing up the SSIS Catalog 31](#_Toc129364980)

[Data Warehouse Operations 32](#_Toc129364981)

[Reorganizing or rebuilding indexes 32](#_Toc129364982)

[Updating data distribution statistics 32](#_Toc129364983)

[Managing partitioned tables 32](#_Toc129364984)

[Backing up the database 32](#_Toc129364985)

[Reporting Services Operations 32](#_Toc129364986)

[Managing Scheduled Tasks 32](#_Toc129364987)

[Backing up Power BI Report Server 32](#_Toc129364988)

[Deployment 33](#_Toc129364989)

[Solution structure 33](#_Toc129364990)

[Parameters 33](#_Toc129364991)

[Manual deployment 34](#_Toc129364992)

[Automating Operational Tasks 34](#_Toc129364993)

[SQL agent 34](#_Toc129364994)

[Subsystems maintenance plan 34](#_Toc129364995)

# Planning a BI Solution

## Project conceptual schema

Data sources

Northwind (упрощённая)



xmlcalendar.ru

ETL



Reporting and Analysis

Data Cleansing



Data management



Data

Warehouse



## Project Infrastructure

### Project Management Infrastructure

* Office productivity applications.
* 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!AueE3aY8CojZjpFN7KdU2STPcDtUMA?e=CEFZNN))
  + 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)
* 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

* Development Servers
* Test Servers
* [Download SQL Server 2019 Developer Edition](https://go.microsoft.com/fwlink/?linkid=866662) ([Configuration file](https://1drv.ms/u/s!AueE3aY8CojZlMppqryUZmonKVHj-Q?e=1vriaj))

## 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.

## Business 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

## Project Scope

|  |  |  |
| --- | --- | --- |
| High business value |  | * See sales amount with discount by customers and products. * Compare sales amount with discount with previous year/quarter/month/week. * 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. |

# Planning SQL Server Business Intelligence Infrastructure

## BI Topology

### Single BI server



* SQL Server DB Engine
  + Data Warehouse
  + Landing database
  + SSIS catalog
  + Reporting Services catalog
  + DQS catalog
  + MDS catalog
* 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 database
  + SSIS catalog
  + DQS catalog
  + MDS catalog
* 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 database

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
  + SSIS catalog
  + DQS catalog
  + MDS catalog
* 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 database

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
* Data Quality Services
* Master Data Services

SRVMDM



* SQL Server DB Engine
  + SSIS catalog
* 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 database

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
* Data Quality Services
* Master Data Services

SRVMDM



* SQL Server DB Engine
  + SSIS catalog
* 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.

## 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

### Data source profiling

Data sources

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Profiling |
| Northwind (упрощённая) | SQL Server | 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)[[1]](#footnote-1) |
| [Производственный календарь в XML (xmlcalendar.ru)](http://xmlcalendar.ru/) | XML | Information about holidays in Russia | [xmlcalendar/data (github.com)](https://github.com/xmlcalendar/data) |

#### Customer

Table 1. Northwind (упрощённая).dbo.Customers.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CustomerID | CompanyName | ContactName | ContactTitle | City | Country | Phone | Fax |
| ALFKI | Alfreds Futterkiste | Maria Anders | Sales Representative | Berlin | Germany | 030-0074321 | 030-0076545 |
| ANATR | Ana Trujillo Emparedados y helados | Ana Trujillo | Owner | Mexico D.F. | Mexico | (5) 555-4729 | (5) 555-3745 |
|  | SCD1 | SCD1 | SCD1 | SCD2 | SCD2 | SCD1 | SCD1 |
| Business key | Drill-thought detail | Drill-thought detail | Slicer | Hierarchy | | Drill-thought detail | Drill-thought detail |

#### Product

Table 2. Northwind (упрощённая).dbo.Products.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ProductID | ProductName | SupplierID | CategoryID | UnitPrice |
| 1 | Chai | 1 | 1 | 18,00 |
| 2 | Chang | 1 | 1 | 19,00 |
|  | SCD1 |  |  |  |
| Business key | Drill-thought detail | Foreign key | Foreign key (Hierarchy) | Drill-thought detail |

Table 3. Northwind (упрощённая).dbo.Categories.

|  |  |  |
| --- | --- | --- |
| CategoryID | CategoryName | Description |
| 1 | Beverages | Soft drinks, coffees, teas, beers, and ales |
| 2 | Condiments | Sweet and savory sauces, relishes, spreads, and seasonings |
|  | SCD1 |  |
| Business key | Drill-thought detail | Drill-thought detail |

#### Employee

Table 4. Northwind (упрощённая).dbo.Employees.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| EmployeeID | LastName | FirstName | Title | TitleOfCourtesy | City | Country |
| 1 | Davolio | Nancy | Sales Representative | Ms. | Seattle | USA |
| 2 | Fuller | Andrew | Vice President, Sales | Dr. | Tacoma | USA |
|  | SCD1 | SCD1 | SCD2 | SCD1 | SCD2 | SCD2 |
| Business key | Drill-thought detail | Drill-thought detail | Drill-thought detail | Drill-thought detail | Hierarchy | |

#### Order

Table 5. Northwind (упрощённая).dbo.Orders

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OrderID | CustomerID | EmployeeID | OrderDate | RequiredDate | ShippedDate | ShipCity | ShipCountry |
| 10248 | VINET | 5 | 1996-07-04 00:00:00.000 | 1996-08-01 00:00:00.000 | 1996-07-16 00:00:00.000 | Reims | France |
| 10249 | TOMSP | 6 | 1996-07-05 00:00:00.000 | 1996-08-16 00:00:00.000 | 1996-07-10 00:00:00.000 | Münster | Germany |
| Degenerate dimension | Dimension key | Dimension key | Dimension key | Dimension key | Dimension key |  |  |

Table 6. Northwind (упрощённая).dbo.Order Details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderID | ProductID | UnitPrice | Quantity | Discount |
| 10248 | 11 | 14,00 | 12 | 0 |
| 10248 | 42 | 9,80 | 10 | 0 |
| Degenerate dimension | Dimension key | Measure | Measure | Measure |

### 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 |  | INT | NOT |
| AlterDateKey | Business 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 | INT | NOT |
| RequiredDateKey | Dimension key | INT |  |
| ShippedDateKey | Dimension key | INT |  |
| 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[[2]](#footnote-2) | 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 | 1. Landing 2. Staging DB data files 3. Logs DB data files | * A separate staging database, landing filegroup and logs database is used, and its data files are stored on a dedicated logical disk to minimize the effects of fragmentation and I/O on the data warehouse data and activity. |
| Disk11 | 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. |
| Disk12 | Database log files | * The logfiles for TempDB, the Staging database, the Logs database and the data warehouse are stored on this drive to isolate log activity from data I/O. |
| Disk13 | 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 likely 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) 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

[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)

## Xmlcalendar

For each xml files

Update local repository

xmlcalendar/data

Github.com

Landing.Holidays

xml file

Sort holidays

Sort days

Left join by holiday Id

Add year to holiday date change datatypes

## Incremental load package

Load Dimensions

Load Facts

Customer

Employee

Product

Date

DQS status subscription activation

DWH maintenance

Order

Truncate landing

## Customer

Northwind (упрощённая)

Customers

Filter on modified date

Clean data with DQS

Load and validate customers data with MDS

Filter on LastChgDateTime and succeeded validation

Add metadata columns

Lookup existing rows

Start audit

Insert new rows (generate surrogate key)

Update SCD1 rows

Customer

Update and Insert SCD2 rows (generate surrogate key)

End audit

DQS

MDS

Customer, ContactName, ContactTitle, Phone, Fax

## Date

xmlcalendar

.\ru

Clean data with DQS

Load and validate holidays data with MDS

Add unknown member

Start audit

Date

End audit

Get start and load dates from Landing.Orders

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

DQS

MDS

## Employee

Northwind (упрощённая)

Employees

Filter on modified date

Clean data with DQS

Load and validate employees data with MDS

Add metadata columns

Lookup existing rows

Start audit

Insert new rows (generate surrogate key)

Update SCD1 rows

Employee, TitleOfCourtesy

Employee

Update and Insert SCD2 rows (generate surrogate key)

End audit

Filter on LastChgDateTime and succeeded validation

DQS

MDS

## Product

Northwind (упрощённая)

Products

Filter on modified date

Clean data with DQS

Load and validate products & categories data with MDS

Add metadata columns

Lookup existing rows

Start audit

Insert new rows (generate surrogate key)

Update SCD1 rows

Product, Category

Product

End audit

Categories

Filter on modified date

Clean data with DQS

Filter on LastChgDateTime and succeeded validation

DQS

MDS

## Order

Northwind (упрощённая)

Orders

Create new partition

Create staging table

Load data to staging table

Start audit

Switch partition with unknown shipped date to fact table

Order

End audit

Order Details

Create filegroups if necessary

Get start load date for incremental load

Switch partition with new shipped date to fact table

Insert or update data for previous dates

Drop staging table

## Data Warehouse maintenance

MDS clean up staging tables

Backup

Copy Data Warehouse metadata to Logs DB

Update statistics

Set filegroups read only

MDS clean up transactions

Optimize indexes

Check DWH integrity

# 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/week | CEO, Sales VP | Power BI | [Standard time-related calculations](https://www.daxpatterns.com/standard-time-related-calculations/) |
| 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/) |
| Cluster customers based on spending volume | Sales VP | Power BI | [Dynamic segmentation](https://www.daxpatterns.com/dynamic-segmentation/) |
| Product ABC classification | Sales VP | Power BI | [ABC classification](https://www.daxpatterns.com/abc-classification/) |
| Product basket analysis | Inside Sales Coordinator | Power BI | [Basket analysis](https://www.daxpatterns.com/basket-analysis/) |
| 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/) |
| 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 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 7. 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 8. 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))[[3]](#footnote-3):
   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 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#optimizing-the-data-model)

### 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 9. 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 Logs 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.

### Considerations for Package Execution

**Timing of package execution**

Every night load new portion of data.

Every first night between Friday and Saturday in month starts optimization partitions of fact tables. In January month and year.

Every night between Saturday and Sunday is created Full backup.

**Execution account**

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

### 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.

### Backing up the SSIS Catalog

Considerations for managing the SSIS catalog:

1. Back up SSIS catalog[[4]](#footnote-4)
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, updating data distribution statistics, backing up the database are implemented in [DWH maintenance package](#_Data_Warehouse_maintenance).

### Reorganizing or rebuilding indexes

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

Uses the sp\_updatestats stored procedure.

### Managing partitioned tables

The manage partitions for each fact table occurs in the [fact load package](#_Order).

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.

### 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.

## 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

1. **Encryption key.** For a report server, you can back up the encryption key by using the Report Server Configuration tool, or the rskeymgmt.exe command line utility. When you back up the key, you must specify a password, which you must provide when you restore the key from the backup.
2. **ReportServer database.** The ReportServer database is the primary application database for the report server. Uses the full recovery model, and therefore supports full, differential, and transaction log backups.
3. **ReportServerTempDB.** The ReportServerTempDB database contains temporary data such as cached reports and datasets. Back up not required. In the event of failure, you can create a new database with the appropriate name, and then run the CatalogTempDB.sql script in the \Program Files\Microsoft Power BI Report Server\PBIRS\ReportServer folder.

## Deployment

### Solution structure

|  |  |  |
| --- | --- | --- |
| Project name | Project type | Description |
| DQS\_STAGING\_DATA | Database | Staging database. Stores data after DQS cleansing. |
| Logs | Database | Database used for storing SSIS logs and Data Warehouse metadata. |
| Monitoring | SSRS | Paginated reports for monitoring. |
| Northwind (упрощённая) | Database | Data source structure |
| NorthwindDW | Database | Data Warehouse database. |
| NorthwindETL | SSIS | ETL |

### Parameters

Table 11. 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 |
| 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. |
| RetrainWeeks | Int16 | Storage depth for Data Warehouse backups in weeks. |

Table 12. 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. |
| DWHDatabaseName | 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. Deploy DQS\_STAGING\_DATA project on DQS server.
3. Deploy Logs on logs server.
4. Deploy NorthwindDW on DWH server.
5. Create SSIS environment ([CreateEnvironment.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateEnvironment.sql&version=GBmaster&_a=contents)).
6. Deploy NorthwindETL on ETL server.
7. 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)).
8. Create roles in MDS database ([CreateRoles.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateRoles.sql)).
9. Create users in databases ([CreateUsers.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/CreateUsers.sql)).
10. Copy script [SetFilegroupsReadOnly.sql](https://dev.azure.com/zinykov/20767/_git/Northwind_BI_Solution?path=/NorthwindDW/Scripts/SetFilegroupsReadOnly.sql) to file system location $(ExternalFilesPath[[5]](#footnote-5))\Scripts
11. Deploy Monitoring to Power BI report server or SQL Server reporting services.
12. Deploy Power BI reports.
13. For testing solution execute Test package using environment created in point 5.

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).

## 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](#_Xmlcalendar) | Ingest data from xmlcalendar | Release | 01:00:00 AM on Sunday | E-mail to DBA team |
| Incremental load | [Incremental load Package](#_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. master
      4. MDS
      5. msdb
      6. ReportServer
      7. ReportServerTempDB
      8. SSISDB
   2. Include indexes
   3. Physical only
2. Backup database task
   1. Databases:
      1. DQS\_MAIN
      2. DQS\_PROJECTS
      3. master
      4. MDS
      5. msdb
      6. ReportServer
      7. 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. master
      4. MDS
      5. msdb
      6. ReportServer
      7. ReportServerTempDB
      8. SSISDB
   2. Include indexes
   3. Physical only
2. Backup database task
   1. Databases:
      1. DQS\_MAIN
      2. DQS\_PROJECTS
      3. MDS
      4. msdb
      5. ReportServer
      6. 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.

1. 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-1)
2. For every fact partition is needed dedicated disk. [↑](#footnote-ref-2)
3. Additional information about performance counters is under the link [BI performance counters (sqlshack.com)](https://www.sqlshack.com/bi-performance-counters/). [↑](#footnote-ref-3)
4. This task can be automated using the [maintenance plan](#_Subsystems_maintenance_plan) shown below. [↑](#footnote-ref-4)
5. Table 11. NorthwindETL parameters. [↑](#footnote-ref-5)