Software Requirements Specification

**Apple DWH Solution**

Table of Content

[**Versions Management** 3](#_Toc111991298)

[**TBD Management** 3](#_Toc111991299)

[**Overview** 3](#_Toc111991300)

[Business Background 3](#_Toc111991301)

[Benefits 4](#_Toc111991302)

[**Requirements** 4](#_Toc111991303)

[Business Requirements 4](#_Toc111991304)

[Functional Requirements 5](#_Toc111991305)

[Non - Functional Requirements 5](#_Toc111991306)

[**Solution Sketch** 6](#_Toc111991307)

[DWH Solution Sketch 6](#_Toc111991308)

[Source Tables structure 6](#_Toc111991309)

[**Scheme Structure** 7](#_Toc111991310)

[Star Scheme 7](#_Toc111991311)

[Logical Diagram 7](#_Toc111991312)

[Snowflake Scheme 7](#_Toc111991313)

[Logical Diagram 7](#_Toc111991314)

[**Aggregate Facts** 8](#_Toc111991315)

[Fact Table 8](#_Toc111991316)

[**User Access and Rights Matrix** 8](#_Toc111991317)

[**Data Warehouse Architecture** 9](#_Toc111991318)

[**Partitioning Fact Table** 10](#_Toc111991319)

[**Parallel Execution** 10](#_Toc111991320)

[**Extraction Description** 11](#_Toc111991321)

[Offline Extraction 11](#_Toc111991322)

[Full Extraction 11](#_Toc111991323)

[**Transformation Description** 11](#_Toc111991324)

[SQL Transformation 11](#_Toc111991325)

[PL/SQL Transforming 11](#_Toc111991326)

[Transforming Data Using Table Functions 11](#_Toc111991327)

[Business solution concept Transformation Strategy 11](#_Toc111991328)

[**Transportation Description** 12](#_Toc111991329)

[Transportation Using Flat Files 12](#_Toc111991330)

[Transportation Using Transportable Tablespaces 12](#_Toc111991331)

[**Glossary** 14](#_Toc111991332)

# **Versions Management**

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | 07/24/2022 | Document created, (Added Version Management, TBD Management, Overview, Requirements, and Solution Sketch chapters). |
| 2.0 | 07/25/2022 | Added Scheme Structure chapter (Logical Diagram) |
| 2.1 | 08/03/2022 | Minor fixes, added Parallel Execution chapter, edited Glossary |
| 2.2 | 08/17/2022 | Added Chapter Extraction Description |
| 2.3 | 08/21/2022 | Added Chapters Transformation Description, Transportation Description |

# **TBD Management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Short description | Date of creating | Status | Date of determination | Determination Version Number |
| TBD – 01 | Which reports? | 07/24/22 | Non – determined |  |  |
| TBD – 02 | Oher business requirements | Non – determined |  |  |
| TBD – 03 | Oher functional requirements | Determined | 08/21/22 | 2.3 |
| TBD – 04 | Oher non – functional requirements | Determined | 08/21/22 | 2.3 |
| TBD – 05 | Source tables structure, e.g. add more tables | Determined | 08/21/22 | 2.3 |
| TBD – 06 | Determined | 08/21/22 | 2.3 |
| TBD – 07 | Determined | 08/21/22 | 2.3 |
| TBD – 08 | Determined | 08/21/22 | 2.3 |
| TBD – 09 | Overview | Determined | 08/04/22 | 2.1 |

# **Overview**

As an example, I wanted to demonstrate Apple sales to show specifics of building DWH in this sphere. (TBD - 09) Nowadays apple sells so many products every second, minute, hour that it would be hindsight enough not to have a picture expressed in data about them. One of the most reasonable solution that can be used to store all these data is a DWH.

## Business Background

Apple Inc. is an American multinational technology company that specializes in consumer electronics, software and online services. Apple is the largest technology company by revenue and it is the world's second biggest company by market capitalization, the fourth-largest personal computer vendor by unit sales and second-largest mobile phone manufacturer. It is one of the Big Five American information technology companies, alongside Alphabet, Amazon, Meta, and Microsoft. In fact, Apple Has More Cash than the US Government.

Most popular products:

1. IPhone
2. Services
3. Mac
4. IPad
5. Wearables, Home and Accessories

## Benefits

1. **Time saving.** In the modern fast-paced world of cutthroat competition, your capacity as a business is to swiftly make refined decisions is essential to outpace your opponents. A DWH provides access to all required data in a matter of minutes, so you and your employees do not have to dread an approaching deadline. All you need to do is deploy your data model to acquire data within seconds. Our DWH solution allow you to do that without using a complex query or machine learning. With data warehousing, your business will not have to rely on the 24/7 availability of a technical expert to troubleshoot problems associated with retrieving information. This way, we can save plenty of time.
2. **Improves Data Quality.** The refined quality of data helps guarantee that Apple Company’s policies are based on precise information about your corporate exertions. By understanding the data warehousing meaning, we can transform data from multiple sources into a shared arrangement. Consequently, we can ensure the reliability and quality of corporate data. This way, we can identify and remove replicated data, poorly recorded data, and any other errors.
3. **Increases BI – operations performance.** We can use a data warehouse to gather, assimilate, and derive data from any source and set up a process to leverage business analytics. As a result, BI will improve rapidly, owing to the capability of effortlessly integrating data from distinct sources.

With a data warehouse in place, every team member can have an integrated understanding of all the relevant information in a timely manner.

1. **Storing Historical Data.** Data Warehouse allows storing large volumes of historical data from operational databases. It is easy to investigate different time phases and inclinations that can be innovative for your company. Thus, with the right and real-time data in your hands, you can make superior corporate decisions concerning business strategies. Moreover, predicting the results of business processes is a significant aspect. It can be challenging to forecast the future without a tangible understanding of our historical achievements and letdowns of Apple Inc. Thus, using EDW gives us an advantage in business procedures.

# **Requirements**

## Business Requirements

|  |  |
| --- | --- |
| Business Requirements | |
| ID | Requirement Description |
| BR – 01 | Get an actual information about different separate product sales |
| BR – 02 | Get reports structured by different time dimensions to decrease amount of time needed to create right report. |
| BR – 03 | Improve data storing performance to have an ability to target different offers |
| BR – 04 | Improve data access performance |
| BR – 05 | Improve security of data storages |
| BR – 06 | Improve the quality of stored data |
| BR – 07 | Ensure long - lasting and effective storage of historical data |

## Functional Requirements

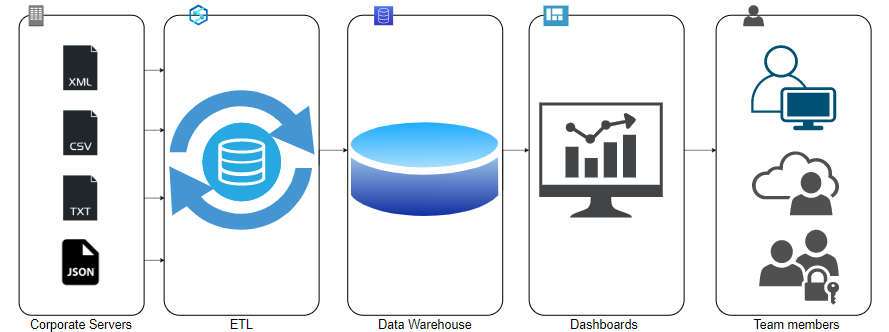
|  |  |
| --- | --- |
| Functional Requirements | |
| ID | Requirement Description |
| FR – 01 | System should give an opportunity to see sales statistics divided by different time dimensions |
| FR – 02 | System should give an opportunity to see sales statistics divided by different geo dimensions |
| FR – 03 | System should give an opportunity to choose geo and time dimensions that user interested in to see in report |
| FR – 04 | System should store time and geo dimension packages on company servers at least 5 years from the date of commissioning |
| FR – 05 | System should generate daily/weekly/monthly reports automatically and store them on company servers 5 years |
| FR – 06 | System should give an opportunity to increase or decrease the amount of time the data will be stored on servers |
| FR – 07 | System should give an opportunity to make reports manually (not only at the end of day/week/month) |
| FR – 08 | System should give an opportunity to configure reports in Data Mart Level (Except of FR – 01 / FR - 02) (… TBD – 01) |
| FR – 09 | System should automatically update an information about actual sales data. |
| FR – 10 | DWH Architecture should consist of Storage, Cleansing, Data Warehouse, Data Warehouse Prepare Star Cleansing Level, STAR – Cleansing and STAR Levels ([Data Warehouse Architecture](#_Data_Warehouse_Architecture) Chapter) |
| FR – 11 | System should have an opportunity to be built using either [STAR](#_Star_Scheme) or [SNOWFLACKE](#_Snowflake_Scheme) Schema |
| TBD – 03 | … |

## Non - Functional Requirements

|  |  |
| --- | --- |
| Functional Requirements | |
| ID | Requirement Description |
| NFR – 01 | System should give an opportunity to access data in any time (24 hours a day) |
| NFR – 02 | System should be integrated with Banking API |
| NFR – 03 | System should update data no longer than 5 seconds (except of complex queries) |
| NFR – 04 | System should have an actual back up version of current region sales on master server and current region server |
| NFR – 05 | System should have an opportunity to store an information about hew opened stores (extensible) |
| NFR – 06 | System should have an opportunity to use it (or its separate parts) in other company’s products(other DWH’s e.g.) |
| NFR – 07 | System should have built according to ETL philosophy |
| TBD - 04 | … |

# **Solution Sketch**

## DWH Solution Sketch

At the diagram above you can see Conceptual Sketch Diagram that shows stages of data transforming: 

Picture 1 - [DWH Data Transformation sketch](https://drive.google.com/file/d/10h7tpl3ZA6PTKaClzNf1Xgne3ohdrE5x/view?usp=sharing) (Link)

## Source Tables structure

|  |  |
| --- | --- |
| Name | T\_products |
| Description | Table contains data about different products |
| Data Source | Flat files or operational databases |
| Frequency of updating | Automatically every week / manually |
| TBD – 05 | … |

|  |  |
| --- | --- |
| Name | T\_stores |
| Description | Table contains data about different stores and their locations |
| Data Source | Flat files or operational databases |
| Frequency of updating | Automatically every month / manually |
| TBD – 06 | … |

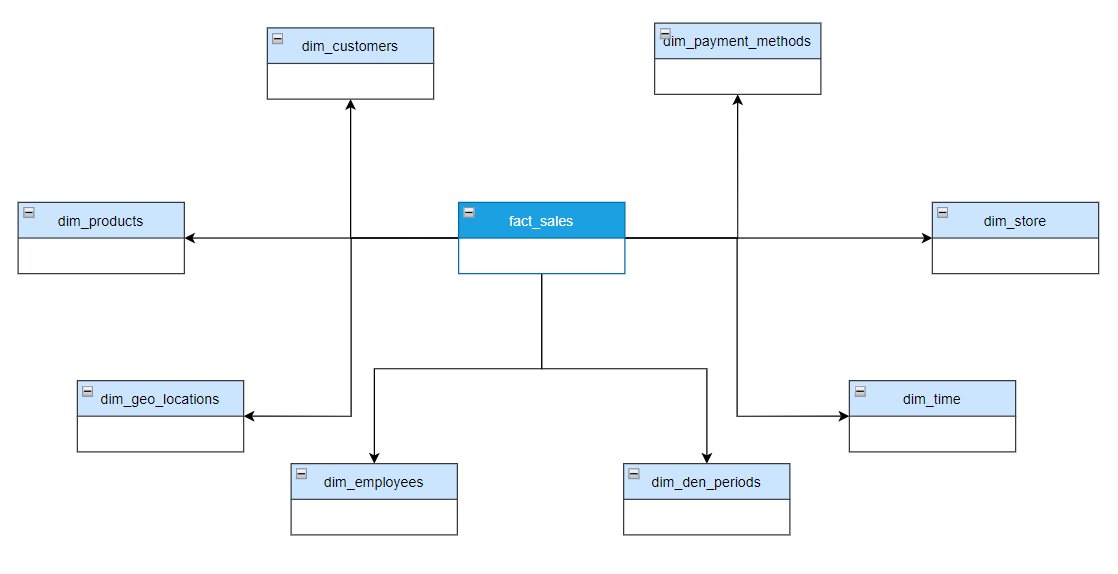
|  |  |
| --- | --- |
| Name | T\_customers |
| Description | Table contains data about all customers |
| Data Source | Flat files or operational databases |
| Frequency of updating | Automatically every customer creates new account on web site / manually |
| TBD – 08 | … |

|  |  |
| --- | --- |
| Name | T\_countries |
| Description | Table contains data about different countries and regions |
| Data Source | Flat files or operational databases |
| Frequency of updating | Automatically with opening new stores or sales markets / manually |
| TBD – 07 | … |

# **Scheme Structure**

## Star Scheme

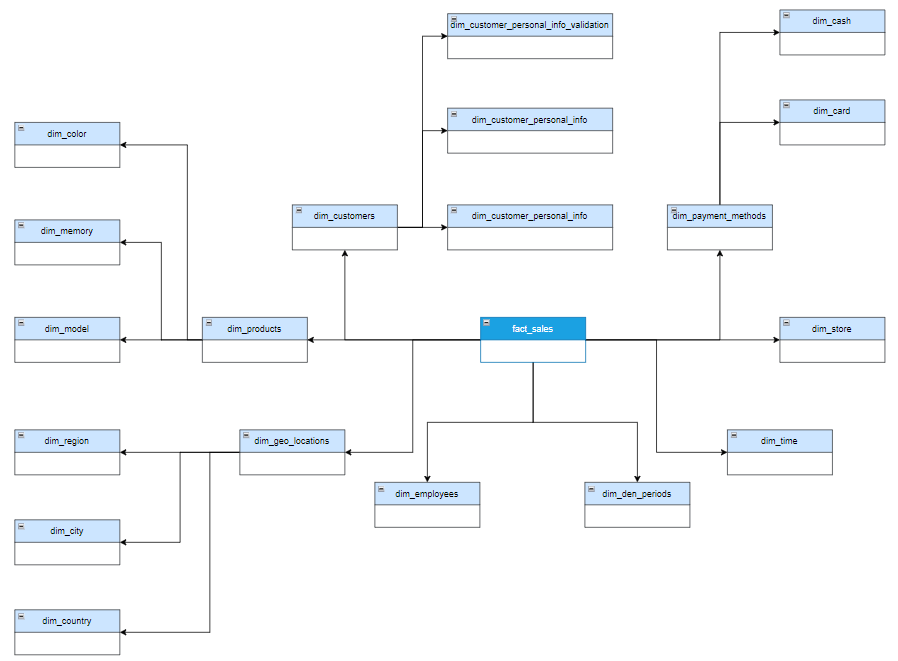
## Logical Diagram



Picture 3 - [Logical Diagram](https://drive.google.com/file/d/1KqppAhMj9edn3CljhLuN5aDDsmBu0UIH/view?usp=sharing) (Link)

## Snowflake Scheme

## Logical Diagram



Picture 5 - [Logical Diagram](https://drive.google.com/file/d/1a9XCwER_AamCMXMwhrDoYYRC90YwF9nf/view?usp=sharing) (Link)

# **Aggregate Facts**

## Fact Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Table Name | Additive | Descriptions |
| Total products sold | Sales\_Amount | FACT\_SALES | + | Shows the amount of sold products according to range of time, product etc. |
| Total employeed | Employees\_Amount | FACT\_SALES | + | Shows the amount hired employees according to selected period e.g. |

# **User Access and Rights Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User  Function | Admin | Super users (Chiefs, CO – founders etc.) | Manager | BI – Analyst | Other Systems (TBD) |
| DML | CRUD | CRUD | CRUD | CRD | CRUD |
| DDL | CRUD | CRUD | CRUD | CRUD | CRUD\* |
| DCL | L | CRUD | L | L | L |
| DQL | CRUL | CRUD | CRUL | CRUD | CRUD |

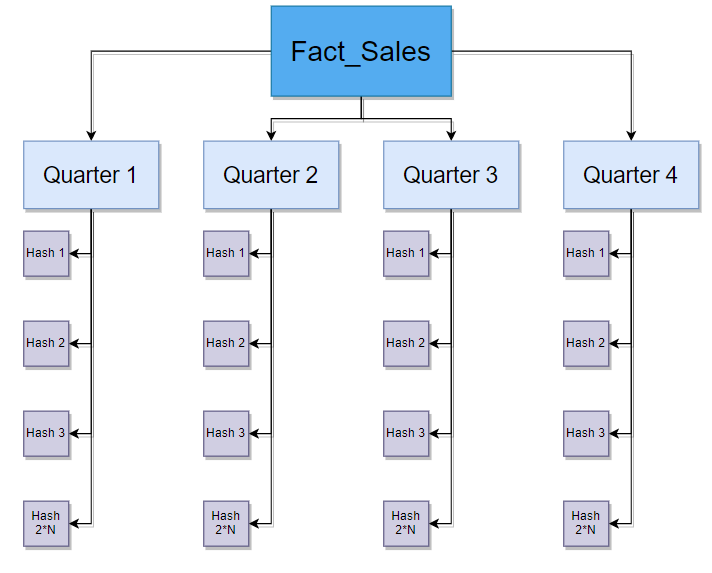
* C – create
* R – read (COST > 50)
* U – update
* D – delete
* L – list (COST < 50)

\*(exp. can call creating entities, queries)

# **Data Warehouse Architecture**

|  |  |  |  |
| --- | --- | --- | --- |
| Level Type | Object Name | Tablespace | Description |
| Storage level | SA\_CUSTOMERS | ts\_SA\_CUSTOMERS\_DATA\_01 | Loads data from flat files, stores data about customers/products |
| SA\_PRODUCTS | ts\_SA\_PRODUCTS\_DATA\_01 |
| DW Cleansing Level | DW\_CLEANSING | ts\_DW\_CLEANSING | Loads from stage level |
| SA Date | SA\_DATE | ts\_sa\_date | Contains all info about dates/locations |
| DW Level | DW\_DATA | ts\_DW\_DATA\_01 | Loads data from cleansing level and preparing it for to the 3 normal form |
| DW Prepare Star Cleansing Level | SAL\_DW\_CL | ts\_SA\_DW\_CL\_01 | Loads data from DW system. Contains views and objects from DW - Level |
| STAR Cleansing | SAL\_CL | ts\_SA\_CL\_001 | Loads data from DW\_CL, but clean some redundancy if are some |
| STAR Level | DM\_FCT\_SALES | ts\_SA\_FCT\_SALES\_01 | INFO |
| DM\_CUSTOMERS | ts\_SA\_DIM\_CUSTOMERS\_01 |
| DM\_PRODUCTS | ts\_SA\_DIM\_PRODUCTS\_01 |
| DM\_GEO | ts\_SA\_DIM\_GEO\_01 |

# **Partitioning Fact Table**

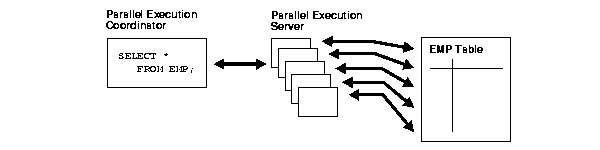
****

Picture 6 - [Partitioning Fact Table](https://drive.google.com/file/d/1zI7lp4bh-fcQjHRa-U6Rf8kwjx2uSpq8/view?usp=sharing) (Link)

# **Parallel Execution**

Parallel execution is the ability to apply multiple CPU and I/O resources to the execution of a single database operation. It dramatically reduces response time for data-intensive operations on large databases typically associated with decision support systems (DSS) and data warehouses.

Now it is clear that we can use parallel execution to increase query performance know that we are going to operate with enormous amount of data.



Picture 7 - Parallel Execution ex.

# **Extraction Description**

## Offline Extraction

From our point of view the *offline extraction* can be assessed as followed: as data is not directly taken from the source, it already has an existing structure, namely it can be tablespaces and logs or in another case the structure may be a result of the extraction method.

## Full Extraction

Another type of extraction, namely, full extraction may be described as complete pulling of data directly from the source. This may be beneficial in a sense that there is no need to track the source system as the extraction process reflects all the available data in the system.

# **Transformation Description**

## SQL Transformation

The SQL transformation processes SQL queries midstream in a pipeline. The SQL transformation can be an active or passive transformation. You can insert, delete, update, and retrieve rows from a database. You can pass the database connection information to the SQL transformation as input data at run time. The transformation processes external SQL scripts or SQL queries that you create in an SQL editor. The SQL transformation processes the query and returns rows and database errors.

## PL/SQL Transforming

In a data warehouse environment, you can use procedural languages such as PL/SQL to implement complex transformations in the Oracle Database. Whereas CTAS operates on entire tables and emphasizes parallelism, PL/SQL provides a row-based approached and can accommodate very sophisticated transformation rules.

For example, a PL/SQL procedure could open multiple cursors and read data from multiple source tables, combine this data using complex business rules, and finally insert the transformed data into one or more target table. It would be difficult or impossible to express the same sequence of operations using standard SQL statements.

## Transforming Data Using Table Functions

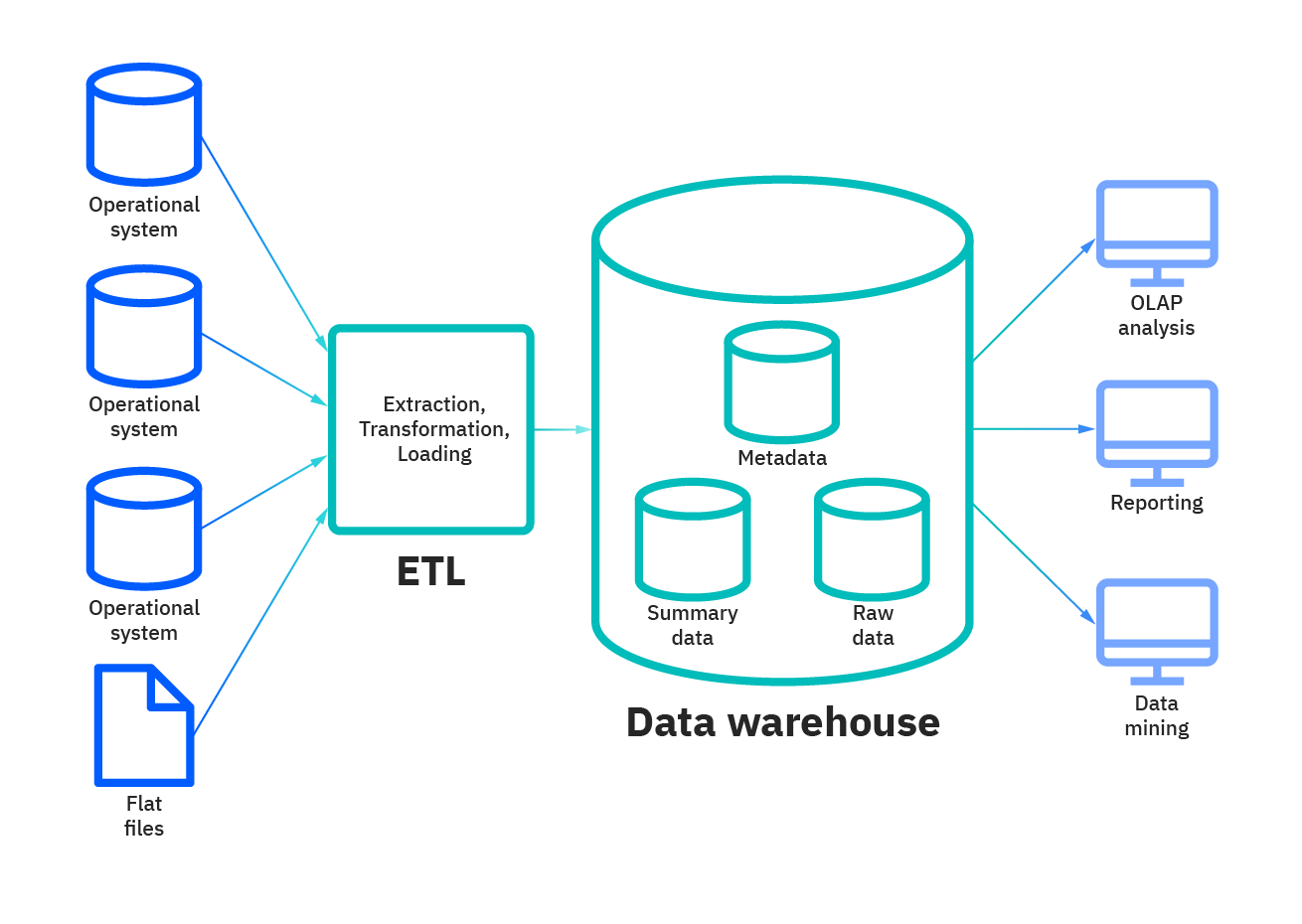
Table functions provide the support for pipelined and parallel execution of transformations implemented in PL/SQL, C, or Java.

Scenarios as mentioned earlier can be done without requiring the use of intermediate staging tables, which interrupt the data flow through various transformations steps.

## Business solution concept Transformation Strategy

The right way would be to use a combo of Transformation types. Thus different objects need to be transformed in different ways.

Reference tables are not needed to be transformed, for example, using PL/SQL Transformation Strategy as well as Customers and Employees Dimensions and Fact Sales table, for example are needed to be updated or loaded within new data using Multiple Cursors, Functions and Procedures to manipulate with several objects at the same time.



Picture 8 – Transformation Strategy

# **Transportation Description**

## Transportation Using Flat Files

The most common method for transporting data is by the transfer of flat files, using mechanisms such as FTP or other remote file system access protocols. Data is unloaded or exported from the source system into flat files, and is then transported to the target platform using FTP or similar mechanisms.

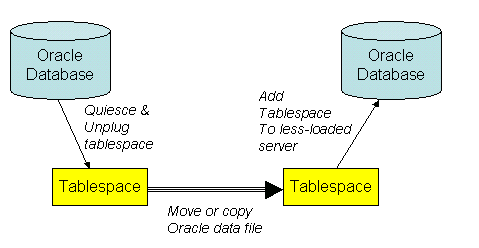
Source systems and DWH often use different operating systems and database systems, using flat files is often the simplest way to exchange data between heterogeneous systems with minimal transformations. However, even when transporting data between homogeneous systems, flat files are often the most efficient and most easy-to-manage mechanism for data transfer.

The best way to get the greatest performance is working with flat files because they are easy to generate, use quite a little memory, easy to parse and easy to generate from back – end algorithms.

## Transportation Using Transportable Tablespaces

Transportable tablespaces is a feature of the Oracle database, introduced with Oracle 8i, that allows DBAs to copy or move tablespaces between databases. It is especially helpful when tablespace size is in huge, in Terabytes and tradition ways are not as quicker as scp..ing the datafiles. In other words, Transportable Tablespaces allow Oracle data files to be unplugged from a database and copied to another Oracle instance (on another server) and then added into that Oracle instance (refer to Figure below).

The most common applications of transportable tablespaces in data warehouses are in moving data from a staging database to a data warehouse, or in moving data from a data warehouse to a data mart.

****

Picture 9 – Transportation Strategy

# **Glossary**

* Customer – a person (or business) who bought at least 1 product.
* Cost - an estimate of the run-time for a given operation.
* Dashboard - a dashboard is a visual display of data.
* Logical Scheme - a data model of a specific problem domain expressed independently of a particular database management product or storage technology.
* Physical Scheme - a representation of a data design as implemented, or intended to be implemented, in a database management system.
* TBD – To be determined
* DWH – Data WareHouse
* EDW – Enterprise Data WareHouse
* ETL – Extract, Transform and Load
* API – Application Programming Interface