UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT REPORT

DATA WAREHOUSE AND INTEGRATION

Topic:

BUILD A DATA WAREHOUSE

ON THE ADVENTUREWORKS 2019 DATABASE

MODULES: SALES AND PURCHASING

Ho Chi Minh City, December 23, 2022

UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT REPORT

DATA WAREHOUSE AND INTEGRATION

Topic:

BUILD A DATA WAREHOUSE

ON THE ADVENTUREWORKS 2019 DATABASE

MODULES: SALES AND PURCHASING

Team UP:

NO	Name	ID	Faculty	Role	Phone	Email
1	Vu Van Son	K20410669	Khoa HTTT	Member	0944244857	sonvv20416c @st.uel.edu.vn
2	Chau Nguyen Huong Thao	K204162000	Khoa HTTT	Member	0919772802	thaocnh20416c @st.uel.edu.vn
3	Huynh Hong Uyen	K204162003	Khoa HTTT	Leader	0934190565	uyenhh20416c @st.uel.edu.vn
4	Nguyen Tuong Vy	K204162009	Khoa HTTT	Member	0764187037	vynt20416c @st.uel.edu.vn

Lecturer: **MS. Nguyen Van Ho**

Ho Chi Minh City, December 23, 2022

ACKNOWLEDGEMENTS

During the project, we got a lot of devoted help and advice from teachers. Besides, we also referenced a number of authors. Thanks to your distribution, we can fully complete our final project.

Despite our best effort, mistakes are inevitable. Therefore, we are glad to receive your judges and comments in order to improve our research. Those will be our enormous motivation to develop our project on the horizon.

COMMITMENT

We commit that our final Data Warehouse and Integration project is unique due to the whole team's research. There are still some documents we referenced, which we have listed and cited particularly in the report.

If all of the above is wrong, we will take all responsibilities from professors.

Ho Chi Minh City, December 23, 2022.

Committed person

TEAM UP

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACRONYMS	FULL NAME
AWC	Adventure Works Cycles
BI	Business Intelligence
DHW	Data Warehouse
DBMS	Database Management System
ETL	Extract, transform and load
OLTP	Online Transaction Processing
SCD	Slowly Changing Dimension

PROJECT EVALUATION

MEMBER	CONTRIBUTION	SIGNATURE
Vu Van Son	100%	
Chau Nguyen Huong Thao	100%	
Huynh Hong Uyen	100%	
Nguyen Tuong Vy	100%	

1. INTRODUCTION

1.1. Business case for the project

The Microsoft product sample for online transaction processing (OLTP) databases is called AdventureWorks 2019 Database. The AdventureWorks 2019 database is used by Adventure Works Cycles, an established international manufacturing company.

The enterprise system (AdventureWorks 2019 Database) on which Adventure Works Cycles relies too heavily on all reports. When the search takes almost half an hour or more and the demand for the data is huge, these two factors combined will slow down the system and can lead to crashes. Recent historical reporting efforts take too long, deadlines are missed, and current reporting takes up too much of the operation enterprise staff's time.

The project seeks approval to create a data warehouse architecture to improve information availability and support better and more informed decision making. Data warehouse professionals will use ETL tools, a key component of data warehouse architectures, to model and build target warehouses, extract data from a variety of sources, transform data so that OE architects and developers can access and load multiple target database.

The data warehouse supports business operations including sales, purchasing, and inventory management.

1.2. Objectives of the project

Once a data warehouse is fully deployed, using the Online Analytical Processing approach, optimized routing (OLAP) to enable implementation of granular advanced features. Provide decision-makers with a fact-based assessment of 2 main department (Sale and Purchasing) of the company's previous performance, and data can be shared between the mentioned departments to get the most out of it.

1.3. General Objective

This project aims to build data warehouse and integrate so that data can be retrieved easily and quickly generate reports for the purpose of supporting business operations in sales and purchasing departments.

1.4. Specific Objectives

Study about the basic knowledge related to Data warehouse to be able to apply to build a good Data warehouse model.

Learn about the AdventureWorks 2019 database and set business requirements to create dimensions needed for sales and purchases department.

Perform the ETL process that will extract all the data from the database. So these are the flows from "source" to "destination". During this process, the conversion engine will process all data changes.

Implements construction of a Multi-Dimensional Cube that acts as a database and is suitable for OLAP applications and data warehouses for online analytical processing (OLAP cubes). And used to generate reports and enable more efficient analysis, using appropriate data storage in a multi-dimensional format to make the data appear more logical.

1.5. Research Objects

The research objective is to gain meaningful insights into the operation of your sales and purchasing department's organization, a data warehouse aims to generate a treasure trove of historical data that can be retrieved and examined.

1.6. Scope of the project

The project aims to build a data warehouse from the AdventureWorks sample database to support the main audience, which are lecturers, students, and those who intend to study in this field. In addition, the project is within the scope of the Data Warehouse subject, so the knowledge learned from that subject will be mainly used in this project.

The project has some scope limitations. Using Microsoft's AdventureWorks sample data - although this is a very formal sample database - it is difficult to avoid the fact that the data is not very realistic and the construction of a data warehouse to solve business problems based solely on the hypotheses we studied based on other businesses. But the analysis based on sample data still provides some basic knowledge and background, the closest experience to reality, before having the opportunity to use data from real businesses.

1.7. Value and desired outcome of the project

The value that the project wants to bring is including knowledge about building a data warehouse, data warehouse applications to solve problems that businesses face. In addition, having the opportunity to interact with supporting software in building a data warehouse is also a valuable opportunity to experience more. Building a data warehouse is very meaningful for the development of the business. When the regular database is not structured for reporting, difficult to access, Captured in a silo and not integrated with all the other data to give a complete picture. That's where data warehousing makes sense with being able to extract data from your organizational systems, load it into a centralized location, and transform and integrate the data into an optimized format. for analysis. And these things are always necessary in business, so building a data warehouse always makes sense.

Within the scope of the project, building a data warehouse with two main areas, Sales and Purchasing, will support businesses to make decisions about which products to maintain, which products to develop, and to promote strategies. translate to increase sales, evaluate productivity suppliers, ..

1.8. Structure of projec

We will first explore AdventureWorks data to understand business processes and pose research questions. Next we will split up into small teams to jointly build a data warehouse to solve the problem together. The data warehouse will be built on Kimball's method

Parts covered in the project include:

- First, give some general theory in chapter 1.
- Then, analyze business processes of the business in chapter 2.
- Next, analysis of research questions in chapter 3.
- And, Build a suitable data warehouse to solve the above research questions in chapter 4.
- Finally, conclusion in chapter 5.

The project will be started on a small scale, designed in accordance with the business needs of the enterprise and the construction time of the data warehouse is less than 1 year, and then it will develop gradually and be applied more widely. than. At the project in this subject, it will only be on a small scale to solve some of the problems of the Sales and Purchasing departments.

2. THEORETICAL BASIS

2.1. Overview of Data Warehouse (DWH)

How can a business store all its data for historical analysis, business performance planning, and present and future strategies in the most optimal and effective way? That answer is Data Warehouse. The data warehouse is the solution to help businesses store information in the most optimal way.

2.1.1. What is DWH?

A data warehouse is a collection of decision support technologies, which aim to enable knowledge workers (executives, managers, analysts) to make better and faster decisions (Chaudhuri, S., & Dayal, U, 1997) it can be understood as a central location where centralized data from multiple locations. Data warehouse technology has been successfully deployed in a variety of industries: manufacturing (for Shipment and customer support), retail (to manage user records and inventory), financial services (for requirements analysis, etc.), risk analysis, credit card analysis, and fraud detection), shipping (for fleet management), telecommunications (for call analysis and fraud detection), utilities (for analytics power usage) and healthcare (for results analysis),...

2.1.2. DWH Architecture

The DWH architecture (Moody, D. L., & Kortink, M. A. , 2000) consists of the following components:

- **Operational systems**

These are systems that record details of business transactions. This is where most of the data needed to support the decision is generated.

- **External sources**

Data warehouses often combine data from external sources (e.g. census data, economic data) to support analysis.

- **Extract processes**

These processes "load" data into the data warehouse on a regular basis (daily, weekly, monthly). Data is extracted from different sources, merged and matched, and stored in a consistent format. This corresponds to a shopping function.

- **Central data warehouse**

This serves as the central source of decision support data across the enterprise. This forms the “wholesale level” of the data warehouse environment and is used to power data marts. Central data warehouses are typically implemented using a traditional relational DBMS.

- **Load processes**

These processes feed data from the central data warehouse to the data mart. This corresponds to a distribution function.

- **Data marts:**

They represent the “retail stores” of the data warehouse that provide data in a usable form for analysis by end users. Data marts are often tailored to the needs of a particular set of users or decision-making tasks. They can be “real” (stored as fact tables populated from a central data store) or virtual (defined as views on a central data store). Data marts can be deployed using traditional relational DBMS or OLAP tools.

- **End users:**

Write queries and analysis against data stored in a data warehouse using “user-friendly” query tools.,...

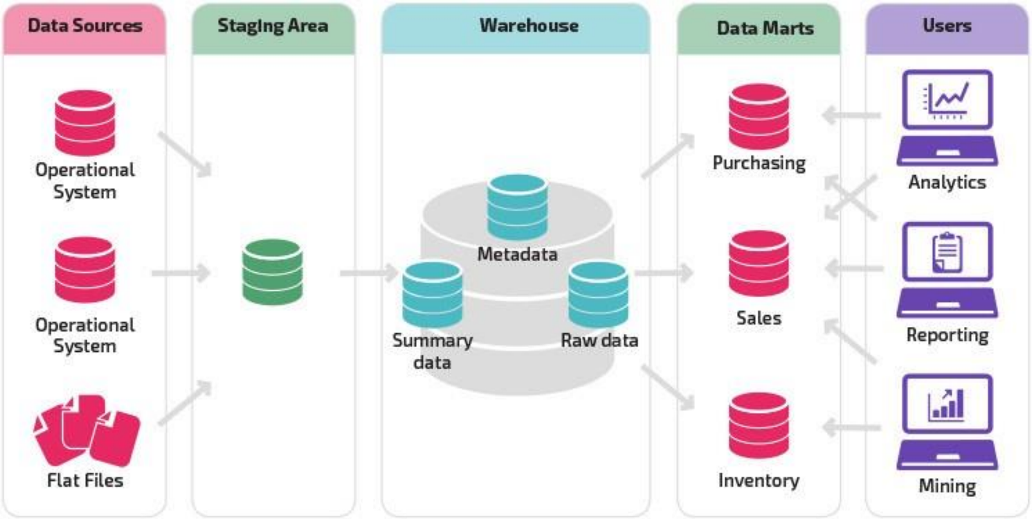


Figure 1: Data Warehouse Architecture
(Source: https://www.astera.com/type/blog/data-warehouse-architecture)

2.1.3. Advantage of BI in enterprises

Business intelligence(BI) (Tableau, 2022) is a set of tools and techniques that collects and analyzes large volumes of data to provide organizations with a clear and complete picture of their data. The objective of a BI system is to the enterprises with a clear, individualized view of their data to enable them to make data-driven decisions. Many BI systems are intuitive to use, highly customizable, and self-service. BI includes processes such as Data Mining, Infrastructure, Visualization, Analytics, and more.

Advantages of BI in the enterprise are Faster partitioning, intuitive control panel

- Increase organizational efficiency
- Data-driven business decisions
- Improved customer experience
- Improve employee satisfaction
- Trusted and Managed Data
- Increase competitive advantage

2.2. Data warehouse and Data mart

2.2.1. What are Data warehouse and Data mart?

A data warehouse is a place where multi-division organizations can store all data sets of each department into a single database. Some of the main features are:

- Old data is retained when new data is uploaded, providing access to large amounts of historical data.
- Data is usually sorted by subject.
- Enables complex retrieval processes.

Data Mart can be understood as a simplified version of Data Warehouse, which provides users with data about one of the organization's parts or an enterprise edge. The main functions of a Data mart are usually to provide relevant information that is needed to make important decisions in a particular department of the company.

The need to build Data mart:

- Demand for input data from the business area
- Data Separation

Top-Down Approach (Bill Inmon). He introduced the concept of a data warehouse that will contain all the information of a business. Data Marts serving different business needs can only be created after having a complete Data Warehouse.

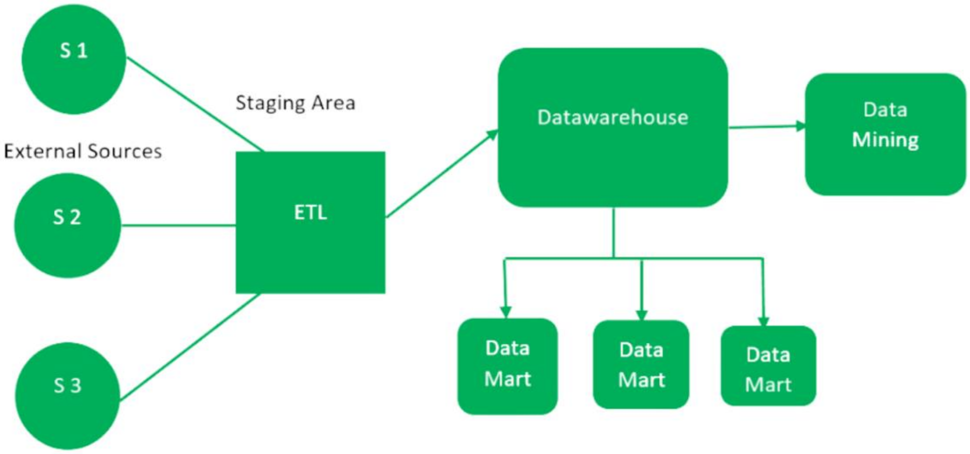


Figure 2: Top-down approaches for building data warehouse.
(Source: https://www.ques10.com/p/43662/top-down-and-bottom-up-approaches-for-building-dat)

While the Bottom-Up approach (Ralph Kimball) suggests that Data Marts are created first to serve the reporting needs of the business. A Data Mart will correspond to a single business field such as sales, finance. The Data Mart is then integrated to build a complete data warehouse.

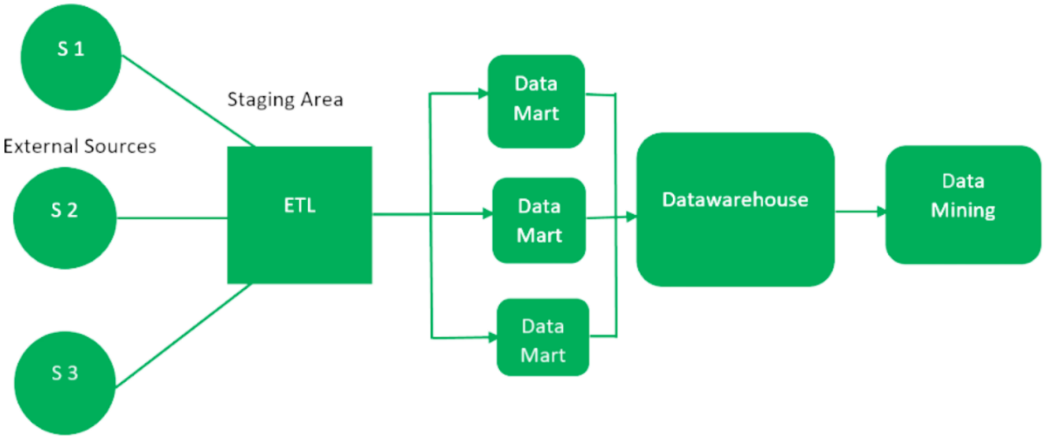


Figure 3: Bottom-up approaches for building data warehouse
(Source: https://www.ques10.com/p/43662/top-down-and-bottom-up-approaches-for-building-dat)

2.2.2. Who needs Data warehouse and Data mart?

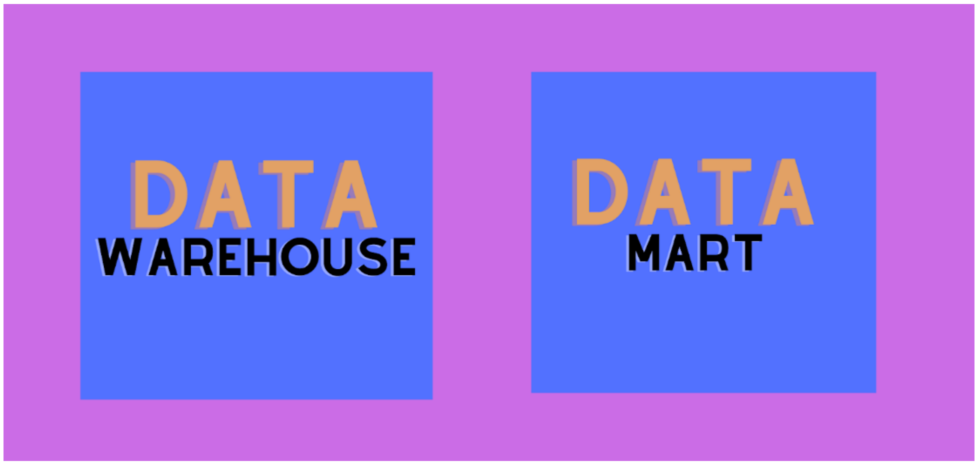


Figure 4: Data Warehouse and Data Mart

A data mart is limited to a single focus for a certain line of business or a certain part of an enterprise. While a data warehouse, with more data, lasts longer, more granular will be used across the enterprise and span many different areas.

2.2.3. Advantages and disadvantages of Data warehouse

All data in Data Warehouse is placed in a centralized repository, while Data Mart does not store data centrally.

A Data Warehouse has concise and curated data, while a Data Warehouse has detailed data.

Data Warehouses are typically longer-lived, more information-oriented, and easier to manipulate while Data marts are typically shorter-lived, project-oriented, and constrained.

The basis for comparison	**Data Warehouse**	**Data Mart**
System	Decentralized	Concentration
Data	Detail	Not detail

Time for using	Flexible, data-oriented and long life.	Limited, project oriented and short life.
Build	Difficult	Simple

Table 1: Data Warehouse and Data Mart

2.3. Snowflake and Star schemas

2.3.1. The Star schema

The Star schema of a data warehouse, where a fact table and several related dimension tables can be found in the center of the star. Because of its star-like structure, it is called a star schema. The most basic type of Data Warehouse schema is the Star Schema data model. It is specialized for searching huge data sets and is also known as Star Join Schema. SQL queries are efficient because there are fewer joins. Data redundancy is high and requires more storage space.

2.3.2. The Snowflake schema

A snowflake ER diagram is created by logically arranging the tables in a multidimensional database, known as a snowflake schema. The star schema is extended by the snowflake schema, adding new dimensions. The data is divided into additional tables due to the normalization of the dimension tables. SQL queries perform slightly worse than star schema because there are more joins. Compared to star scheme, data redundancy is low and uses less disk space.

2.4. Integration and ETL Process

2.4.1. What is ETL?

ETL known as Extract-Transform-Load is the process of how data is loaded from the source system into the data warehouse. Data is extracted from many sources such as OLTP database (operational system data), flat file, and external data,... transformed to fit the data warehouse schema, and loaded into the data warehouse. (Kimball, R., & Caserta, J, 2004)

2.4.2. Why do we need ETL?

Enterprises own many data sources and they are saved in many different formats (Excel, CSV...), email, web pages,… or managed on many different database management systems (SQL Server, Oracle, NoSQL servers,...). How can all data from different sources be converted into one homogeneous? ETL will help enterprises to solve this problem.

2.4.3. ETL Process

The ETL process includes steps: Extract, Transform and Load. (IBM, 2022)

- **Extract**

The first step, raw data is extracted or from source locations to a staging area. Data can be structured or unstructured and be extracted from multiple data sources such as flat files, transaction data, operational data, web pages, …

- **Transform**

In the staging area, raw data goes through data processing like filtering, cleaning, deduplication, translations, summarizations, performing calculations, renaming row and column headers for consistency, and other tasks. Here, the data is transformed and consolidated.

- **Load**

In the loading step, the converted data is moved from the staging area into the destination data warehouse. This step is typically set up to be automated, well-defined, continuous, and batched. ETL is conducted outside business hours when traffic on the source system and data warehouse is at its lowest.

3. REQUIREMENTS ANALYTICS

3.1. Business processes

3.1.1. Department

- **Sale department**

AWC sales include Two sales channels, such as resellers and internet/online. Bicycles, clothing and accessories & parts are the three main product categories for AWC's online sales. Bicycles for sale under three major brands, including mountain bikes, road bikes and touring bikes.

Multiple sales territories were given to sales representatives from regional sales offices. A team manager and multiple sales agents work at each regional office.

- **Purchasing department**

The department in charge of managing the sourcing of raw materials and various ingredients from several suppliers. In order to send important information and documents such as purchase orders, invoices, payments, and product specifications to Adventure Works Cycles, the purchasing department identified a key supplier interested in the design of the product. establish data exchange with the company.

- **Sale & Purchasing department: Inventory**

Organizations can better meet customer needs and prevent overstocking or out of stock by tracking all incoming and outgoing products using an inventory management department. Every critical business function—including accounting, manufacturing, inventory management, purchasing, selling, and customer service—involves inventory management.

3.1.2. The purpose of Production, Sales and Inventory

- **The purpose of Purchasing**

The organization's business purpose of the purchasing department plans high-level procurement activities. The mission of the strategic purchasing department is to facilitate the strategic sourcing of goods (e-sourcing) at reasonable cost and top quality. At the strategic purchasing stage, choices are made between in-house production and purchasing from external suppliers.

The second purpose of the purchasing department is mainly related to operational management and business operations. The production line is kept running at full capacity by maintaining repeat orders, replenishing inventory, and paying bills. Procurement mostly meets the long-term requirements of the business.

- **The purpose of Sales**

The sales department of a company is responsible for marketing its goods and services. This division includes a salesperson who collaborates to close deals, increase profits, and establishes and maintain connections with customers to drive repeat business and brand loyalty. Look for business opportunities to sell goods and services to both current and potential consumers. Prepare a compelling presentation to the customer about the goods or services they are looking for. Maintain regular contact with customers to learn about their needs and needs. Define budget goals for each year and track quarterly budget execution rates. In the target budget or forecast list, the business opportunity of each client should be considered. Establish relationships with all current and potential consumers to learn about their future plans and coordinate with marketing to meet strategic goals

3.1.3. Business processes

- **Sale processes**

Adventure Works Cycles has two main sales processes: sales through distribution channels and direct sales via the Internet.

For sales channels through distribution channels, the company's customers are agents (Resellers). The sales process will go through 4 steps:

+ Processing orders
+ Packing, delivery of goods
+ Make invoices, record debts
+ Payment

For the Internet sales channel, the company sells directly to customers through the company's website. The process will go through 2 steps:

+ Order payment
+ Delivery, invoice

- **Purchasing Process**

At Adventure Works Cycles, the purchasing department buys raw materials and parts used to manufacture Adventure Works Cycles bikes.

Adventure Works Cycles also buys products for resale, such as bicycle clothing, and bicycle parts such as water bottles and pumps. Information about these products and the supplier they are obtained from is stored in the AdventureWorks sample database.

3.2. Data source and challenges

- **Data source:** *Source: AdventureWorks sample databases - SQL Server | Microsoft Learn*

Microsoft provides the AdventureWorks database. The Adventure Works Illustrated Database is set on the data of a hypothetical company called Adventure Works Cycles. This is a large-scale multinational manufacturing company. The company manufactures and sells bicycles made of metal and synthetic substances. The company's headquarters is located in BoyHell, Washington with 290 employees. The company's market spans from North America, Europe to Asia.

The schemas used in the AdventureWorks database:

Schema	**Description**	**Number of tables**
Human Resources	Employees of the company Adventure Works Cycles.	6
Person	Names and addresses of customers: individual customers, suppliers and employees.	13
Production	Products manufactured and sold by Adventure Works Cycles.	25
Purchasing	Suppliers of products that the company buys.	5
Sales	Customers and data related to the purchase.	18

Table 2: Overview of AdventureWork database

Here, we use tables mainly from Purchasing and Sales schema, in addition, we also use some tables from other Schemas to help build a data warehouse.

- **Challenges**

Adventure Works 2019 can't move and transfer data periodically, it's hard to generate regular reports because a multi-table database is very time consuming to query.

3.3. Business Requirements Analysis (Purchasing, Production, Sales, or HR)

3.3.1. Business Requirements Sales

- Who are the top customers across all product categories?
- Who should salespeople focus their efforts on to develop a long-term relationship with?
- Which demographic issue (geographical area, revenue history, etc.) best describes consumer problems?
- What goods are consumers buying and at what rate?

3.3.2. Business Requirements Purchasing

- Identify needs for products, materials, and services.
- Locating reliable suppliers.
- Control spending and payments.
- On-time delivery rate of goods.
- How many providers were used and how many things they provided.
- Supplier accessibility and Supplier availability.

3.3.3. Business Requirements Inventory

- How to track the status of products and materials in the supply chain?
- How to permanently track inventory to help automate manual tasks?
- How to predict the trend of inventory?

4. BUILDING DATA WAREHOUSE AND INTEGRATING DATA

4.1. Designing Data Warehouse

Since your warehouse is not as powerful as the data it holds, aligning the Department's needs and objectives with the overall project is critical to business performance. Therefore, determining specific business requirements before proceeding to build a data warehouse (Bottom-Up) is the preferred option when designing a data warehouse for businesses and our priority in this project.

By choosing two schemas from Adventure Works 2019 database (Sales and Purchasing), we expect to have a deeper understanding of enterprise in the product life cycle management and generate practical data warehouse structure handling business requirements.

Specifically, sales and purchasing are closely related to each other based on product elements to ensure cash flow in the business. Therefore, the factors required in both modules that need to be included in the analysis are price, shipping, and the person responsible for performing the transaction.

4.1.1. Bus Matrix

As mentioned above, our project concentrate on Sales, Purchasing, and Inventory management. The table below shows general business requirements:

	COMMON DIMENSIONS								
BUSINESS PROCESSES	Date	Products	Warehouses	Customers	Employees	Vendors	Shipping	Location	Promotion
Issues Purchase Orders	x	x	x		x	x			
Inventory Plan	x	x	x	x			x	x	
Supply management	x	x	x			x			
Sales Quota management	x				x				
Vendors Evaluation	x	x				x	x	x	

Price Determination	x	x	x	x	x		x	x	
Promotion Tracking	x	x		x					x
Logistics	x		x	x		x	x	x	
Sales Forecasting	x	x	x	x	x	x	x	x	x
Customer Behaviors Understanding	x	x		x	x			x	x

Table 3: Bus Matrix

In general, there will be 9 dimensions that allow dealing with some following issues in present or future.

4.1.2. Master Data

For business flows, most of the purchasing and sales processes will be taken by employees. Moreover, customers and vendors are involved in the product cycle life.

4.1.3. Transaction Data

For the sales and purchasing schemas, our team concentrated on computing business performance and determining customer insight through revenue and product genres. Moreover, product inventory plays a vital role in ensuring these two processes occur smoothly.

4.1.4. Fact and dimension tables

4.1.4.1. Dimension tables

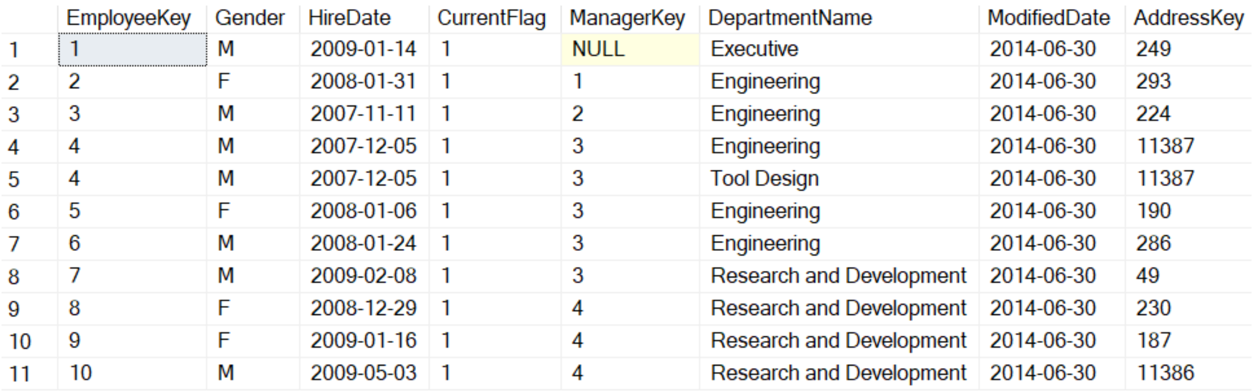
- **DIM Employee Table**

No.	Field Name	Description	Type	Key
1	EmployeeKey	Employee ID, selected as the primary key for employee records	int	PK
2	Gender	Employee's gender Rule: M = Male, F = Female	nchar(1)	

3	HireDate	The date the employee was hired	date	
4	CurrentFlag	Rule: 0 = Inactive, 1 = Active	bit	
5	ManagerKey	The depth of the employee in the corporate hierarchy	smallint	
6	DepartmentName	The name of the department the employee is working in	nvarchar(50)	
7	AddressKey	Employee address code	int	FK
8	ModifiedDate	Date and time the record was last updated	date	

Table 4: DIM Employee Table

```
SELECT [HUMANRESOURCES].[EMPLOYEE].BUSINESSENTITYID AS EMPLOYEEKEY
       ,[HUMANRESOURCES].[EMPLOYEE].GENDER
       ,[HUMANRESOURCES].[EMPLOYEE].HIREDATE
       ,[HUMANRESOURCES].[EMPLOYEE].CURRENTFLAG
       ,[HUMANRESOURCES].[EMPLOYEE].ORGANIZATIONLEVEL AS MANAGERKEY
       ,[HUMANRESOURCES].[DEPARTMENT].NAME AS DEPARTMENTNAME
       ,CONVERT(DATE, [HUMANRESOURCES].[EMPLOYEE].MODIFIEDDATE) AS MODIFIEDDATE
       ,PEBEA.ADDRESSID AS ADDRESSKEY
FROM [HUMANRESOURCES].[EMPLOYEE]
       JOIN [HUMANRESOURCES].[EMPLOYEEDEPARTMENTHISTORY]
ON [HUMANRESOURCES].[EMPLOYEE].BUSINESSENTITYID =
[HUMANRESOURCES].[EMPLOYEEDEPARTMENTHISTORY].BUSINESSENTITYID
       JOIN [HUMANRESOURCES].[DEPARTMENT]
ON [HUMANRESOURCES].[DEPARTMENT].DEPARTMENTID =
[HUMANRESOURCES].[EMPLOYEEDEPARTMENTHISTORY].DEPARTMENTID
       JOIN [PERSON].[BUSINESSENTITYADDRESS] AS PEBEA
ON [HUMANRESOURCES].[EMPLOYEE].BUSINESSENTITYID = PEBEA.BUSINESSENTITYID
ORDER BY [HUMANRESOURCES].[EMPLOYEE].BUSINESSENTITYID
```



	EmployeeKey	Gender	HireDate	CurrentFlag	ManagerKey	DepartmentName	ModifiedDate	AddressKey
1	1	M	2009-01-14	1	NULL	Executive	2014-06-30	249
2	2	F	2008-01-31	1	1	Engineering	2014-06-30	293
3	3	M	2007-11-11	1	2	Engineering	2014-06-30	224
4	4	M	2007-12-05	1	3	Engineering	2014-06-30	11387
5	4	M	2007-12-05	1	3	Tool Design	2014-06-30	11387
6	5	F	2008-01-06	1	3	Engineering	2014-06-30	190
7	6	M	2008-01-24	1	3	Engineering	2014-06-30	286
8	7	M	2009-02-08	1	3	Research and Development	2014-06-30	49
9	8	F	2008-12-29	1	4	Research and Development	2014-06-30	230
10	9	F	2009-01-16	1	4	Research and Development	2014-06-30	187
11	10	M	2009-05-03	1	4	Research and Development	2014-06-30	11386

Figure 5: Details of DIM Employee Table

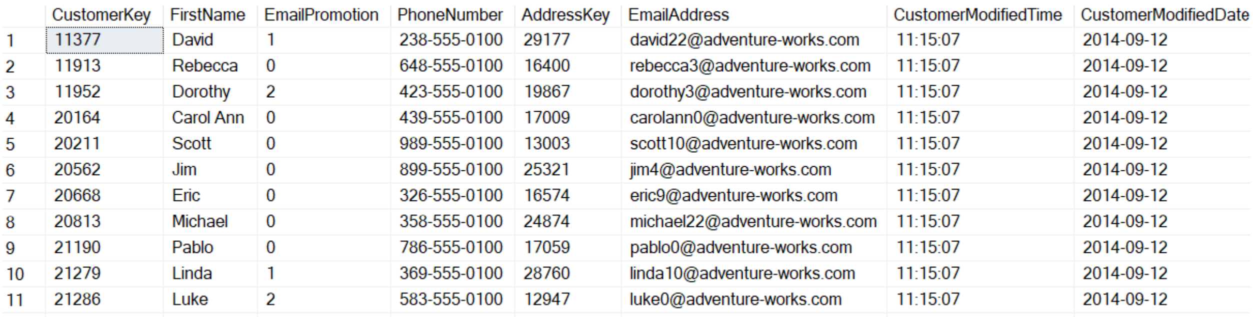
- **DIM Customer Table**

No.	Field Name	Description	Type	Key
1	Customerkey	Primary key of customer	int AUTOINCREMENT	PK
2	FirstName	First name of the person.	Varchar	
3	EmailPromotion	0 = Contact does not wish to receive e-mail promotions, 1 = Contact does wish to receive e-mail promotions from AdventureWorks, 2 = Contact does wish to receive e-mail promotions from AdventureWorks and selected partners.	int DEFAULT 0	
4	PhoneNumber	Telephone number identification number.	int	
5	AddressKey	ID of the address in which the customer is located	int	FK
6	EmailAdress	E-mail address for the person.	nvarchar	
7	CustomerModifiedDate	Date and time the record was last updated in customer table	date	

Table 5: DIM Customer Table

```
SELECT [SALES].[SALESORDERDETAIL].SALESORDERDETAILID AS ORDERKEY
       , [SALES].[SALESORDERDETAIL].PRODUCTID AS PRODUCTKEY
       , [SALES].[SALESORDERDETAIL].ORDERQTY AS ORDERQTY
```

```
        , [SALES].[SALESORDERDETAIL].UNITPRICE AS UNITPRICE
        , [SALES].[SALESORDERDETAIL].UNITPRICEDISCOUNT AS UNITPRICEDISCOUNT
        , [SALES].[SALESORDERDETAIL].LINETOTAL AS LINETOTAL
        , CONVERT(DATE, SALES.SALESORDERDETAIL.MODIFIEDDATE) AS MODIFIEDDATE
FROM [SALES].[SALESORDERDETAIL]
JOIN [SALES].[SALESORDERHEADER]
ON [SALES].[SALESORDERDETAIL].SALESORDERID = [SALES].[SALESORDERHEADER].SALESORDERID
```



	CustomerKey	FirstName	EmailPromotion	PhoneNumber	AddressKey	EmailAddress	CustomerModifiedTime	CustomerModifiedDate
1	11377	David	1	238-555-0100	29177	david22@adventure-works.com	11:15:07	2014-09-12
2	11913	Rebecca	0	648-555-0100	16400	rebecca3@adventure-works.com	11:15:07	2014-09-12
3	11952	Dorothy	2	423-555-0100	19867	dorothy3@adventure-works.com	11:15:07	2014-09-12
4	20164	Carol Ann	0	439-555-0100	17009	carolann0@adventure-works.com	11:15:07	2014-09-12
5	20211	Scott	0	989-555-0100	13003	scott10@adventure-works.com	11:15:07	2014-09-12
6	20562	Jim	0	899-555-0100	25321	jim4@adventure-works.com	11:15:07	2014-09-12
7	20668	Eric	0	326-555-0100	16574	eric9@adventure-works.com	11:15:07	2014-09-12
8	20813	Michael	0	358-555-0100	24874	michael22@adventure-works.com	11:15:07	2014-09-12
9	21190	Pablo	0	786-555-0100	17059	pablo0@adventure-works.com	11:15:07	2014-09-12
10	21279	Linda	1	369-555-0100	28760	linda10@adventure-works.com	11:15:07	2014-09-12
11	21286	Luke	2	583-555-0100	12947	luke0@adventure-works.com	11:15:07	2014-09-12

Figure 6: Details of DIM Customer Table

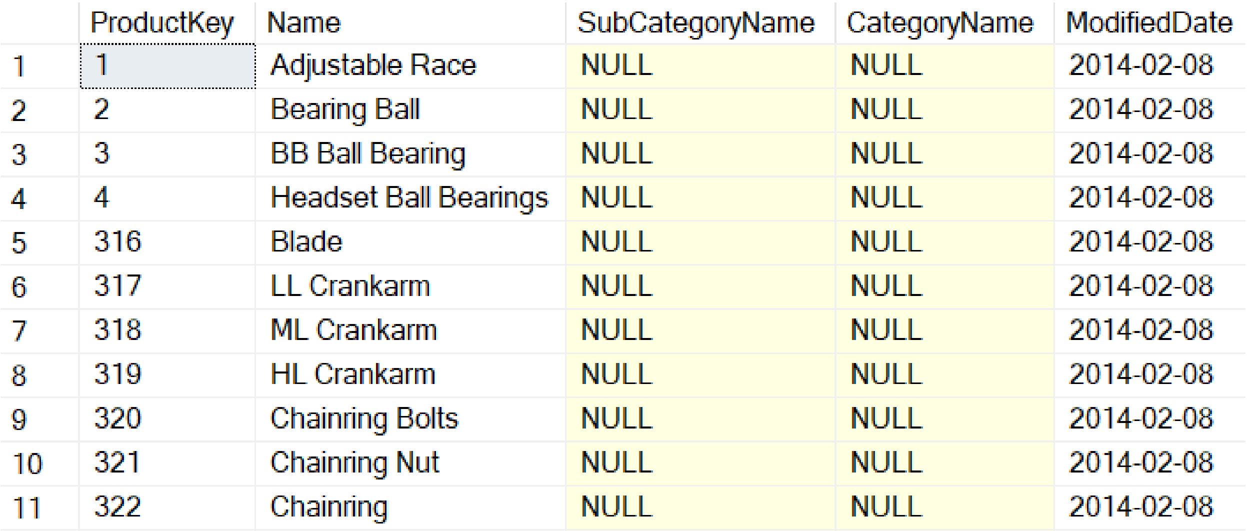
- **DIM Products Table**

No.	Field Name	Description	Type	Key
1	ProductKey	Product code, selected as the primary key	int	PK
2	Name	Name of the product	nvarchar	
3	SubCategoryName	Product is a member of this product subcategory	int	
4	CategoryName	Product category identification number.	int	
5	ModifiedDate	Date and time the record was last updated	date	

Table 6: DIM Products Table

```
SELECT PROPRODUCT.PRODUCTID AS PRODUCTKEY
        , PROPRODUCT.NAME
        , PROCASUB.NAME AS SUBCATEGORYNAME
        , PROCA.NAME AS CATEGORYNAME
        , CONVERT(DATE, PROPRODUCT.MODIFIEDDATE) AS MODIFIEDDATE
FROM [PRODUCTION].[PRODUCT] AS PROPRODUCT
        LEFT JOIN [PRODUCTION].[PRODUCTSUBCATEGORY] AS PROCASUB
ON PROCASUB.PRODUCTSUBCATEGORYID = PROPRODUCT.PRODUCTSUBCATEGORYID
        LEFT JOIN [PRODUCTION].[PRODUCTCATEGORY] AS PROCA
```

```
ON PROCA.PRODUCTCATEGORYID = PROCASUB.PRODUCTCATEGORYID
ORDER BY PRODUCTKEY
```



	ProductKey	Name	SubCategoryName	CategoryName	ModifiedDate
1	1	Adjustable Race	NULL	NULL	2014-02-08
2	2	Bearing Ball	NULL	NULL	2014-02-08
3	3	BB Ball Bearing	NULL	NULL	2014-02-08
4	4	Headset Ball Bearings	NULL	NULL	2014-02-08
5	316	Blade	NULL	NULL	2014-02-08
6	317	LL Crankarm	NULL	NULL	2014-02-08
7	318	ML Crankarm	NULL	NULL	2014-02-08
8	319	HL Crankarm	NULL	NULL	2014-02-08
9	320	Chainring Bolts	NULL	NULL	2014-02-08
10	321	Chainring Nut	NULL	NULL	2014-02-08
11	322	Chainring	NULL	NULL	2014-02-08

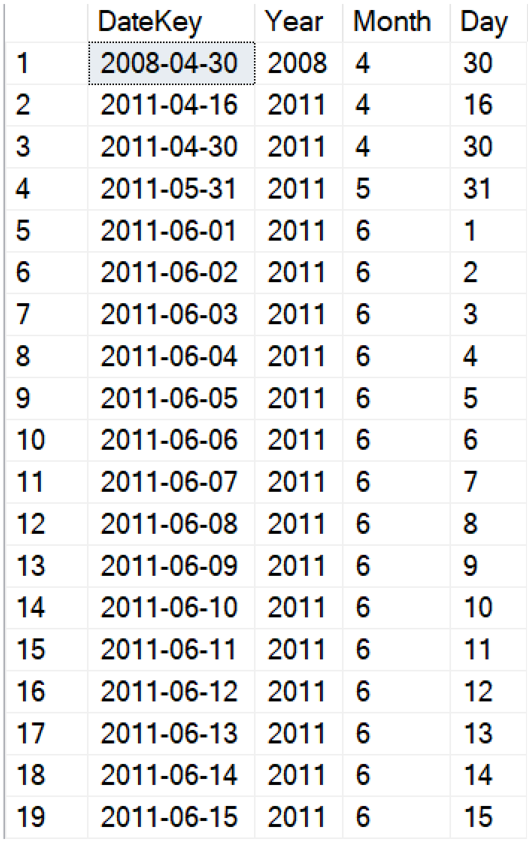
Figure 7: Details of DIM Products Table

- **DIM Time Table**

No.	Field Name	Description	Type	Key
1	DateKey	Order time	date	PK
2	Year	Year of order	int	
3	Month	Month of order	int	
4	Day	Day of order	int	

Table 7: DIM Time Table

```
SELECT CONVERT(DATE,PUPOH.ORDERDATE) AS 'DATEKEY',
  DATEPART(YEAR, PUPOH.ORDERDATE ) AS'YEAR',
  DATEPART(MONTH,PUPOH.ORDERDATE) AS 'MONTH',
  DATEPART(DAY,PUPOH.ORDERDATE) AS 'DAY'
FROM [PURCHASING].[PURCHASEORDERHEADER] AS PUPOH
GROUP BY PUPOH.ORDERDATE

UNION
SELECT CONVERT(DATE,SALESOH.ORDERDATE ) AS 'DATEKEY',
  DATEPART(YEAR, SALESOH.ORDERDATE )  AS 'YEAR',
  DATEPART(MONTH,SALESOH.ORDERDATE ) AS 'MONTH',
  DATEPART(DAY,SALESOH.ORDERDATE ) AS 'DAY'
FROM [SALES].[SALESORDERHEADER] AS SALESOH
GROUP BY SALESOH.ORDERDATE

UNION
SELECT CONVERT(DATE,QUOTADATE) AS 'DATEKEY',
DATEPART(YEAR, QUOTADATE )  AS 'YEAR',
  DATEPART(MONTH,QUOTADATE ) AS 'MONTH',
  DATEPART(DAY,QUOTADATE ) AS 'DAY'
FROM [SALES].[SALESPERSONQUOTAHISTORY]
GROUP BY QUOTADATE

UNION
SELECT CONVERT(DATE,MODIFIEDDATE ) AS 'DATEKEY',
  DATEPART(YEAR, MODIFIEDDATE)  AS 'YEAR',
  DATEPART(MONTH,MODIFIEDDATE) AS 'MONTH',
  DATEPART(DAY,MODIFIEDDATE) AS 'DAY'
FROM [PRODUCTION].[PRODUCTINVENTORY]
GROUP BY MODIFIEDDATE
```



	DateKey	Year	Month	Day
1	2008-04-30	2008	4	30
2	2011-04-16	2011	4	16
3	2011-04-30	2011	4	30
4	2011-05-31	2011	5	31
5	2011-06-01	2011	6	1
6	2011-06-02	2011	6	2
7	2011-06-03	2011	6	3
8	2011-06-04	2011	6	4
9	2011-06-05	2011	6	5
10	2011-06-06	2011	6	6
11	2011-06-07	2011	6	7
12	2011-06-08	2011	6	8
13	2011-06-09	2011	6	9
14	2011-06-10	2011	6	10
15	2011-06-11	2011	6	11
16	2011-06-12	2011	6	12
17	2011-06-13	2011	6	13
18	2011-06-14	2011	6	14
19	2011-06-15	2011	6	15

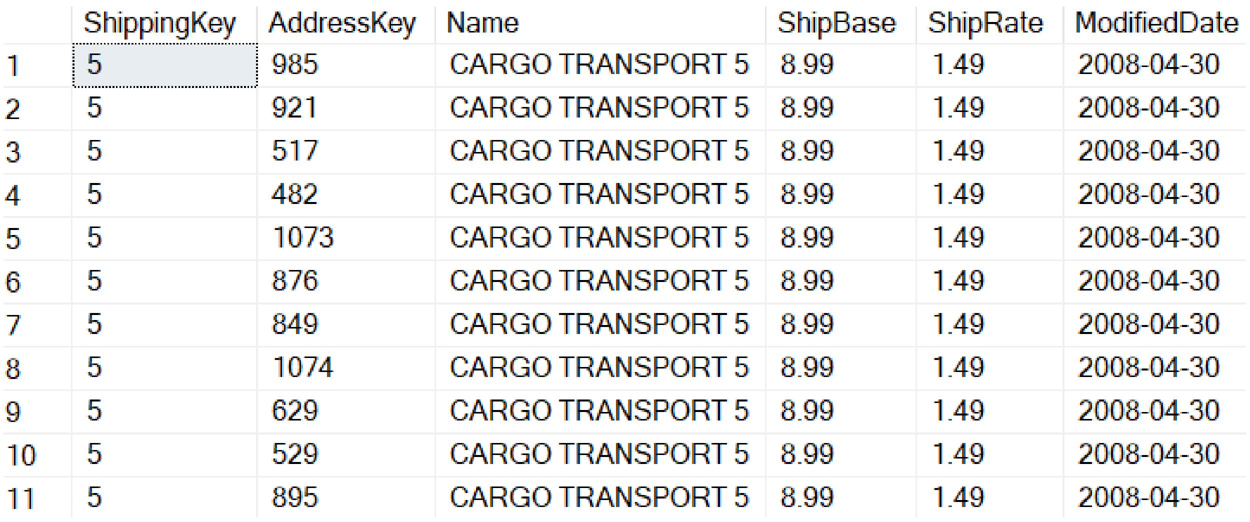
Figure 8: Details of DIM Time Table

- **DIM Shipping Table**

No.	Field Name	Description	Type	Key
1	ShippingKey	Primary key for ShipMethod records	int	PK
2	AddressKey	Customer shipping address	int	FK
3	Name	Shipping company name	nvarchar(50)	
4	ShipBase	Minimum shipping charge	money	
5	ShipRate	Shipping charge per pound	money	
6	ModifiedDate	Date and time the record was last updated	date	

Table 8: DIM Shipping Table

```sql
SELECT PUSHIP.SHIPMETHODID AS SHIPPINGKEY
       , PEADDRESS.ADDRESSID AS ADDRESSKEY
       , PUSHIP.NAME
       , PUSHIP.SHIPBASE
       , PUSHIP.SHIPRATE
       , CONVERT(DATE, PUSHIP.MODIFIEDDATE ) AS MODIFIEDDATE
FROM [PURCHASING].[SHIPMETHOD] AS PUSHIP
       JOIN [SALES].[SALESORDERHEADER] AS SALESSOH
ON PUSHIP.SHIPMETHODID = SALESSOH.SHIPMETHODID
       JOIN [PERSON].[ADDRESS] AS PEADDRESS
ON SALESSOH.SHIPTOADDRESSID = PEADDRESS.ADDRESSID
```



	ShippingKey	AddressKey	Name	ShipBase	ShipRate	ModifiedDate
1	5	985	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
2	5	921	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
3	5	517	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
4	5	482	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
5	5	1073	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
6	5	876	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
7	5	849	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
8	5	1074	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
9	5	629	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
10	5	529	CARGO TRANSPORT 5	8.99	1.49	2008-04-30
11	5	895	CARGO TRANSPORT 5	8.99	1.49	2008-04-30

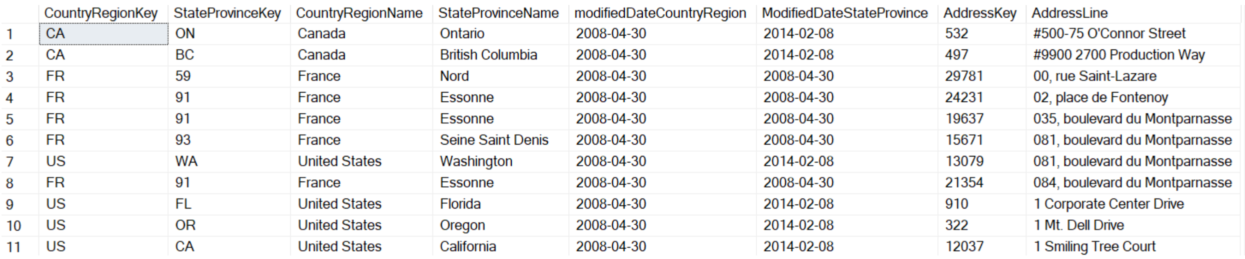
Figure 9: Details of DIM Shipping Table

- **DIM Location Table**

No.	Field Name	Description	Type	Key
1	CountryRegionKey	ISO standard code for countries and regions	nvarchar(3)	PK,FK1
2	StateProvinceKey	ISO standard state or province code	nchar(3)	PK,FK2
3	AddressKey	Address code	int	PK,FK3
4	CountryRegionName	Country name	nvarchar(50)	
5	StateProvinceName	State or province description	nvarchar(50)	
6	AddressLine	Street address line	nvarchar(60)	
7	modifiedDateCountryRegion	Date and time the record was last updated	date	
8	ModifiedDateStateProvince	Date and time the record was last updated	date	

Table 9: DIM Location Table

```
SELECT PECOUNTRY.COUNTRYREGIONCODE AS COUNTRYREGIONKEY
       , PESTATE.STATEPROVINCECODE AS STATEPROVINCEKEY
       , PECOUNTRY.NAME AS COUNTRYREGIONNAME
       , PESTATE.NAME AS STATEPROVINCENAME
       , CONVERT(DATE,PECOUNTRY.MODIFIEDDATE) AS MODIFIEDDATECOUNTRYREGION
       , CONVERT(DATE,PESTATE.MODIFIEDDATE) AS MODIFIEDDATESTATEPROVINCE
       , PEADDRESS.ADDRESSID AS ADDRESSKEY
       , PEADDRESS.ADDRESSLINE1 AS ADDRESSLINE
FROM [PERSON].[COUNTRYREGION] AS PECOUNTRY
       JOIN [PERSON].[STATEPROVINCE] AS PESTATE
ON PECOUNTRY.COUNTRYREGIONCODE = PESTATE.COUNTRYREGIONCODE
       JOIN [PERSON].[ADDRESS] AS PEADDRESS
ON PEADDRESS.STATEPROVINCEID = PESTATE.STATEPROVINCEID
```



	CountryRegionKey	StateProvinceKey	CountryRegionName	StateProvinceName	modifiedDateCountryRegion	ModifiedDateStateProvince	AddressKey	AddressLine
1	CA	ON	Canada	Ontario	2008-04-30	2014-02-08	532	#500-75 O'Connor Street
2	CA	BC	Canada	British Columbia	2008-04-30	2014-02-08	497	#9900 2700 Production Way
3	FR	59	France	Nord	2008-04-30	2008-04-30	29781	00, rue Saint-Lazare
4	FR	91	France	Essonne	2008-04-30	2008-04-30	24231	02, place de Fontenoy
5	FR	91	France	Essonne	2008-04-30	2008-04-30	19637	035, boulevard du Montparnasse
6	FR	93	France	Seine Saint Denis	2008-04-30	2008-04-30	15671	081, boulevard du Montparnasse
7	US	WA	United States	Washington	2008-04-30	2014-02-08	13079	081, boulevard du Montparnasse
8	FR	91	France	Essonne	2008-04-30	2008-04-30	21354	084, boulevard du Montparnasse
9	US	FL	United States	Florida	2008-04-30	2014-02-08	910	1 Corporate Center Drive
10	US	OR	United States	Oregon	2008-04-30	2014-02-08	322	1 Mt. Dell Drive
11	US	CA	United States	California	2008-04-30	2014-02-08	12037	1 Smiling Tree Court

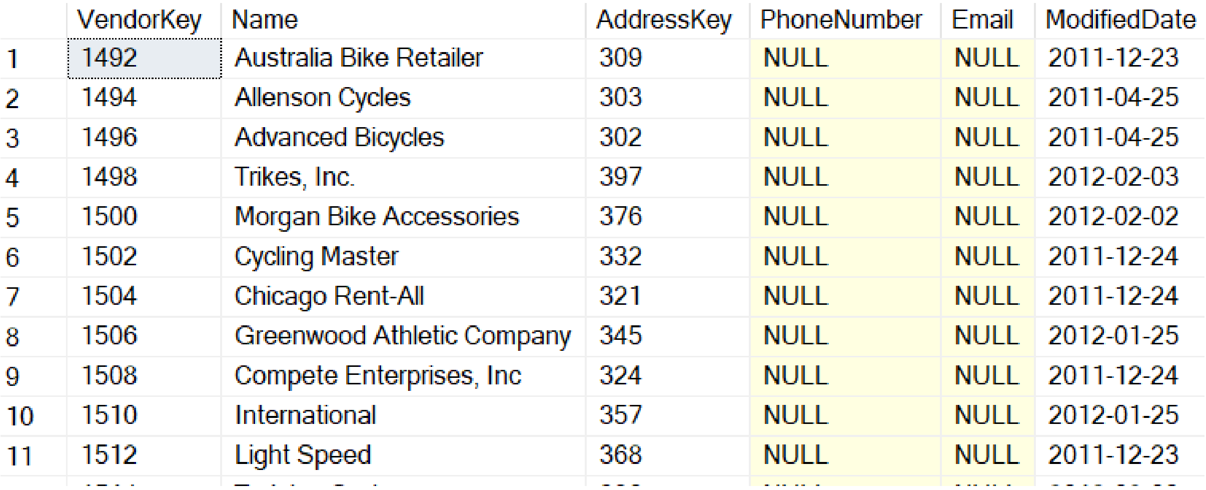
Figure 10: Details of DIM Location Table

- **DIM Vendor Table**

No.	Field Name	Description	Type	Key
1	VendorKey	Vendor ID, selected as the primary key for vendor records	int	PK
2	AddressKey	Vendor address code	int	FK
3	Name	Company name	nvarchar(50)	
4	PhoneNumber	Company’s telephone number	nvarchar(25)	
5	Email	Company’s email	nvarchar(50)	
6	ModifiedDate	Date and time the record was last updated	date	

Table 10: DIM Vendor Table

```
SELECT PUVENDOR.BUSINESSENTITYID AS VENDORKEY
       , PUVENDOR.NAME AS NAME
       , PEBEA.ADDRESSID AS ADDRESSKEY
       , PEPHONE.PHONENUMBER AS PHONENUMBER
       , PEEMAIL.EMAILADDRESS AS EMAIL
       , CONVERT(DATE, PUVENDOR.MODIFIEDDATE) AS MODIFIEDDATE
FROM [PURCHASING].[VENDOR] AS PUVENDOR
       JOIN [PERSON].[BUSINESSENTITYADDRESS] AS PEBEA
ON PUVENDOR.BUSINESSENTITYID = PEBEA.BUSINESSENTITYID
       LEFT JOIN [PERSON].[EMAILADDRESS] AS PEEMAIL
ON PEEMAIL.BUSINESSENTITYID = PUVENDOR.BUSINESSENTITYID
       LEFT JOIN [PERSON].[PERSONPHONE] AS PEPHONE
ON PEPHONE.BUSINESSENTITYID = PUVENDOR.BUSINESSENTITYID
ORDER BY VENDORKEY
```



	VendorKey	Name	AddressKey	PhoneNumber	Email	ModifiedDate
1	1492	Australia Bike Retailer	309	NULL	NULL	2011-12-23
2	1494	Allenson Cycles	303	NULL	NULL	2011-04-25
3	1496	Advanced Bicycles	302	NULL	NULL	2011-04-25
4	1498	Trikes, Inc.	397	NULL	NULL	2012-02-03
5	1500	Morgan Bike Accessories	376	NULL	NULL	2012-02-02
6	1502	Cycling Master	332	NULL	NULL	2011-12-24
7	1504	Chicago Rent-All	321	NULL	NULL	2011-12-24
8	1506	Greenwood Athletic Company	345	NULL	NULL	2012-01-25
9	1508	Compete Enterprises, Inc	324	NULL	NULL	2011-12-24
10	1510	International	357	NULL	NULL	2012-01-25
11	1512	Light Speed	368	NULL	NULL	2011-12-23

Figure 11: Details of DIM Vendor Table

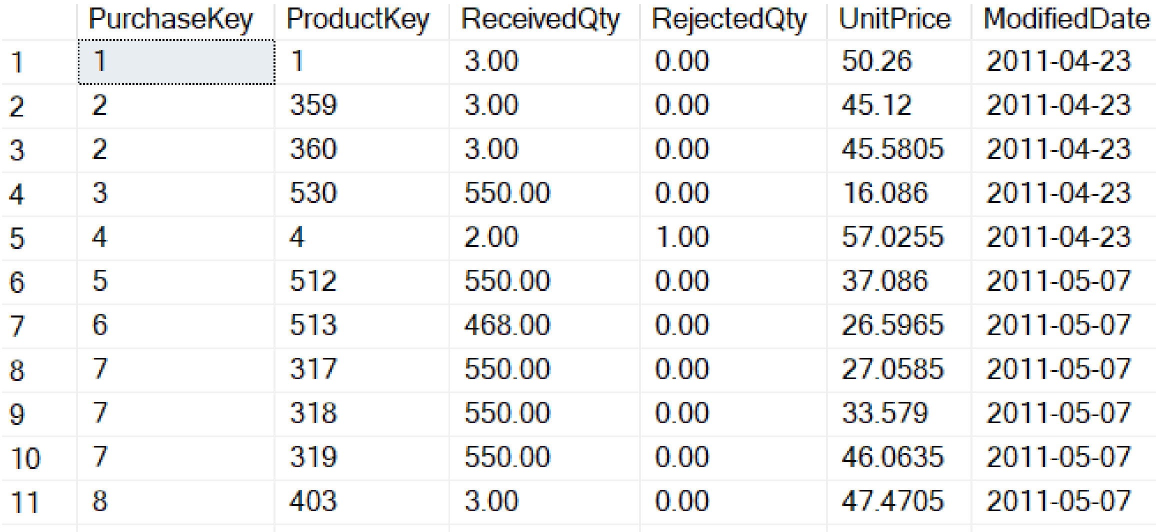
- **DIM PurchaseProduct Table**

No.	Field Name	Description	Type	Key
1	PurchaseKey	Purchase order number, selected as the primary key	int	PK, FK1
2	ProductKey	Product code	int	PK, FK2
3	ReceivedQty	Quantity actually received from the vendor.	decimal(8, 2)	
4	RejectedQty	Quantity rejected during inspection.	decimal(8, 2)	
5	UnitPrice	Price of a single product	money	
6	ModifiedDate	Date and time the record was last updated	date	

Table 11: DIM PurchaseProduct Table

```
SELECT PURCHASEORDERID AS PURCHASEKEY
        , PRODUCTID AS PRODUCTKEY
        , RECEIVEDQTY
        , REJECTEDQTY
```

```
       , UNITPRICE
       , CONVERT(DATE, MODIFIEDDATE ) AS MODIFIEDDATE
FROM [PURCHASING].[PURCHASEORDERDETAIL]
```



	PurchaseKey	ProductKey	ReceivedQty	RejectedQty	UnitPrice	ModifiedDate
1	1	1	3.00	0.00	50.26	2011-04-23
2	2	359	3.00	0.00	45.12	2011-04-23
3	2	360	3.00	0.00	45.5805	2011-04-23
4	3	530	550.00	0.00	16.086	2011-04-23
5	4	4	2.00	1.00	57.0255	2011-04-23
6	5	512	550.00	0.00	37.086	2011-05-07
7	6	513	468.00	0.00	26.5965	2011-05-07
8	7	317	550.00	0.00	27.0585	2011-05-07
9	7	318	550.00	0.00	33.579	2011-05-07
10	7	319	550.00	0.00	46.0635	2011-05-07
11	8	403	3.00	0.00	47.4705	2011-05-07

Figure 12: Details of DIM PurchaseProduct Table

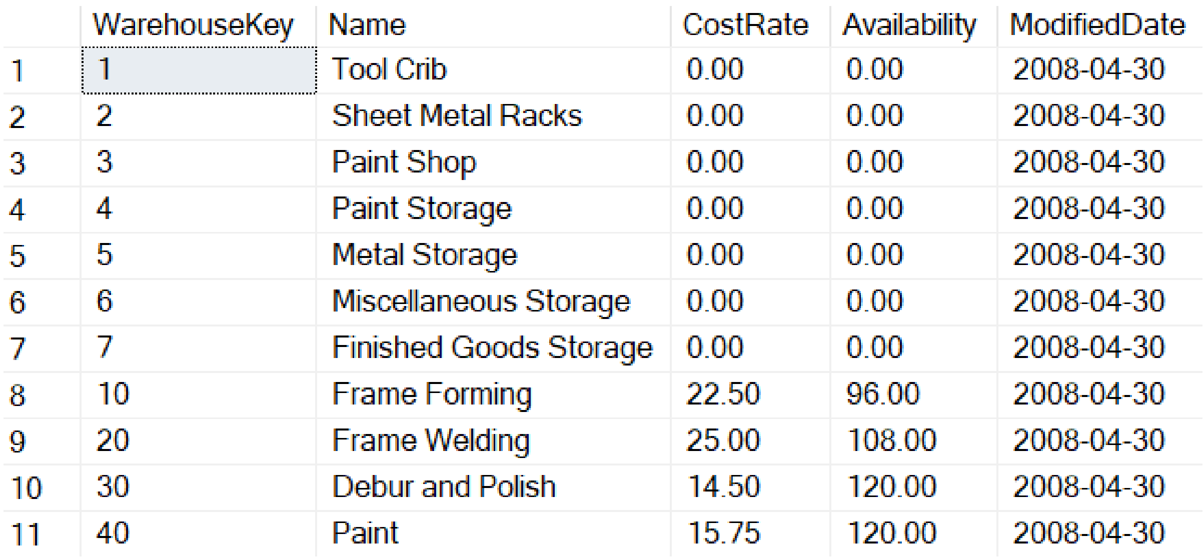
- **DIM Warehouse Table**

No.	Field Name	Description	Type	Key
1	WarehouseKey	Warehouse location, selected as primary key.	smallint	PK
2	Name	Location description	nvarchar(50)	
3	CostRate	Standard hourly cost of the manufacturing location	smallmoney	
4	Availability	Work capacity (in hours) of the manufacturing location	decimal(8,2)	
5	ModifiedDate	Date and time the record was last updated	date	

Table 12: DIM Warehouse Table

```
SELECT LOCATIONID AS WAREHOUSEKEY
       , NAME
```

```
        , COSTRATE
        , AVAILABILITY
        , CONVERT(DATE, MODIFIEDDATE ) AS MODIFIEDDATE
FROM [PRODUCTION].[LOCATION]
```



	WarehouseKey	Name	CostRate	Availability	ModifiedDate
1	1	Tool Crib	0.00	0.00	2008-04-30
2	2	Sheet Metal Racks	0.00	0.00	2008-04-30
3	3	Paint Shop	0.00	0.00	2008-04-30
4	4	Paint Storage	0.00	0.00	2008-04-30
5	5	Metal Storage	0.00	0.00	2008-04-30
6	6	Miscellaneous Storage	0.00	0.00	2008-04-30
7	7	Finished Goods Storage	0.00	0.00	2008-04-30
8	10	Frame Forming	22.50	96.00	2008-04-30
9	20	Frame Welding	25.00	108.00	2008-04-30
10	30	Debur and Polish	14.50	120.00	2008-04-30
11	40	Paint	15.75	120.00	2008-04-30

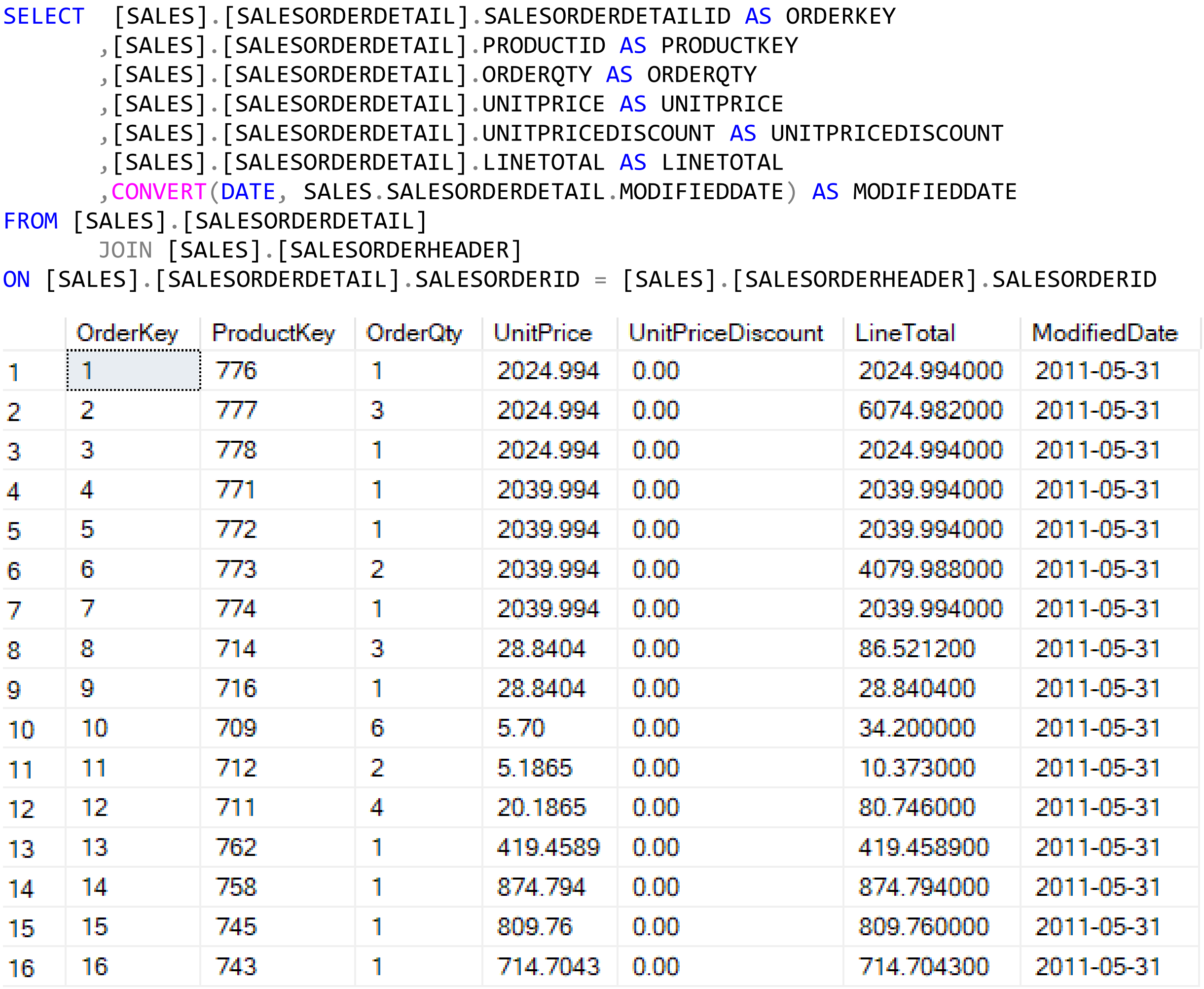
Figure 13: Details of DIM Warehouse Table

- **DIM OrderProduct Table**

No.	**Field Name**	**Description**	**Type**	**Key**
1	Orderkey	Primary key of Sales.SalesOrderHeader	int	PK
2	ProductKey	Product identification number	int	PK
3	OrderQty	Quantity ordered per product.	Varchar	
4	UnitPrice	Selling price of a single product.	int	
5	UnitPriceDiscount	Discount amount	int	

6	Amount	product subtotal. Computed as UnitPrice * (1 - UnitPriceDiscount) * OrderQty.	int	
7	ModifiedDate	Date and time the record was last updated.	date	

Table 13: DIM OrderProduct Table



```
SELECT  [SALES].[SALESORDERDETAIL].SALESORDERDETAILID AS ORDERKEY
       ,[SALES].[SALESORDERDETAIL].PRODUCTID AS PRODUCTKEY
       ,[SALES].[SALESORDERDETAIL].ORDERQTY AS ORDERQTY
       ,[SALES].[SALESORDERDETAIL].UNITPRICE AS UNITPRICE
       ,[SALES].[SALESORDERDETAIL].UNITPRICEDISCOUNT AS UNITPRICEDISCOUNT
       ,[SALES].[SALESORDERDETAIL].LINETOTAL AS LINETOTAL
       ,CONVERT(DATE, SALES.SALESORDERDETAIL.MODIFIEDDATE) AS MODIFIEDDATE
FROM [SALES].[SALESORDERDETAIL]
       JOIN [SALES].[SALESORDERHEADER]
ON [SALES].[SALESORDERDETAIL].SALESORDERID = [SALES].[SALESORDERHEADER].SALESORDERID
```

	OrderKey	ProductKey	OrderQty	UnitPrice	UnitPriceDiscount	LineTotal	ModifiedDate
1	1	776	1	2024.994	0.00	2024.994000	2011-05-31
2	2	777	3	2024.994	0.00	6074.982000	2011-05-31
3	3	778	1	2024.994	0.00	2024.994000	2011-05-31
4	4	771	1	2039.994	0.00	2039.994000	2011-05-31
5	5	772	1	2039.994	0.00	2039.994000	2011-05-31
6	6	773	2	2039.994	0.00	4079.988000	2011-05-31
7	7	774	1	2039.994	0.00	2039.994000	2011-05-31
8	8	714	3	28.8404	0.00	86.521200	2011-05-31
9	9	716	1	28.8404	0.00	28.840400	2011-05-31
10	10	709	6	5.70	0.00	34.200000	2011-05-31
11	11	712	2	5.1865	0.00	10.373000	2011-05-31
12	12	711	4	20.1865	0.00	80.746000	2011-05-31
13	13	762	1	419.4589	0.00	419.458900	2011-05-31
14	14	758	1	874.794	0.00	874.794000	2011-05-31
15	15	745	1	809.76	0.00	809.760000	2011-05-31
16	16	743	1	714.7043	0.00	714.704300	2011-05-31

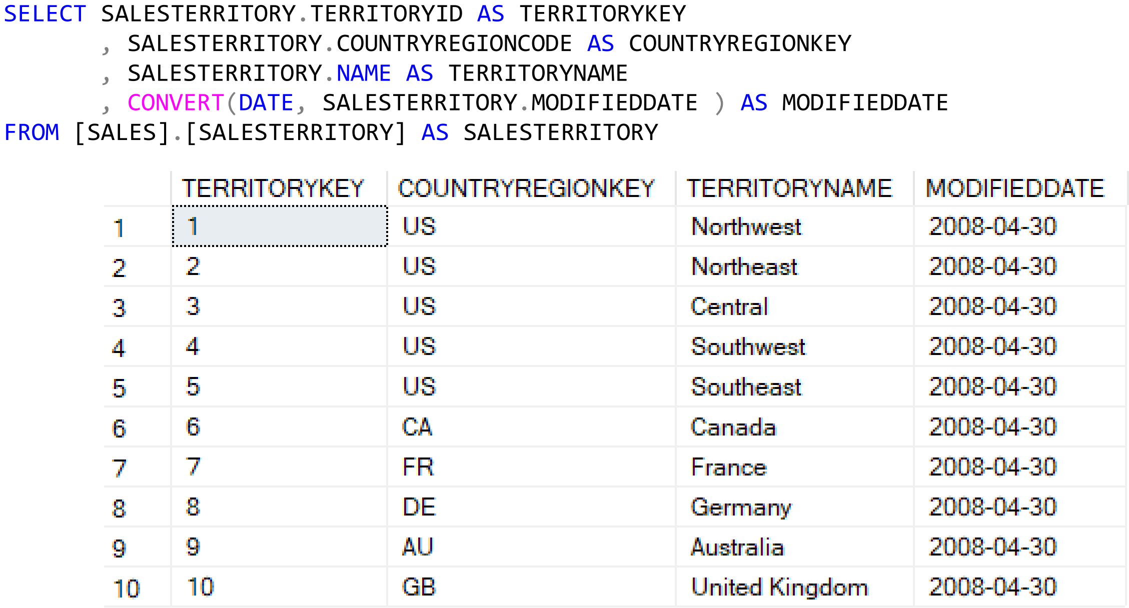
Figure 14: Details of DIM OrderProduct Table

- **DIM SalesTerritory Table**

No.	Field Name	Description	Type	Key

1	Territorykey	Primary key of Territory record	int	PK
2	CountryRegionKey	ISO standard code for countries and regions.	Varchar	FK
3	TerritoryName	Name of sale Territory	Varchar	
4	ModifiedDate	Date and time the record was last updated of **SalesTerritory table**	date	

Table 14: DIM SalesTerritory Table



```sql
SELECT SALESTERRITORY.TERRITORYID AS TERRITORYKEY
        , SALESTERRITORY.COUNTRYREGIONCODE AS COUNTRYREGIONKEY
        , SALESTERRITORY.NAME AS TERRITORYNAME
        , CONVERT(DATE, SALESTERRITORY.MODIFIEDDATE ) AS MODIFIEDDATE
FROM [SALES].[SALESTERRITORY] AS SALESTERRITORY
```

	TERRITORYKEY	COUNTRYREGIONKEY	TERRITORYNAME	MODIFIEDDATE
1	1	US	Northwest	2008-04-30
2	2	US	Northeast	2008-04-30
3	3	US	Central	2008-04-30
4	4	US	Southwest	2008-04-30
5	5	US	Southeast	2008-04-30
6	6	CA	Canada	2008-04-30
7	7	FR	France	2008-04-30
8	8	DE	Germany	2008-04-30
9	9	AU	Australia	2008-04-30
10	10	GB	United Kingdom	2008-04-30

Figure 15: Details of DIM SalesTerritory Table

4.1.4.2. Fact Tables

- **FACT Purchasing Table**

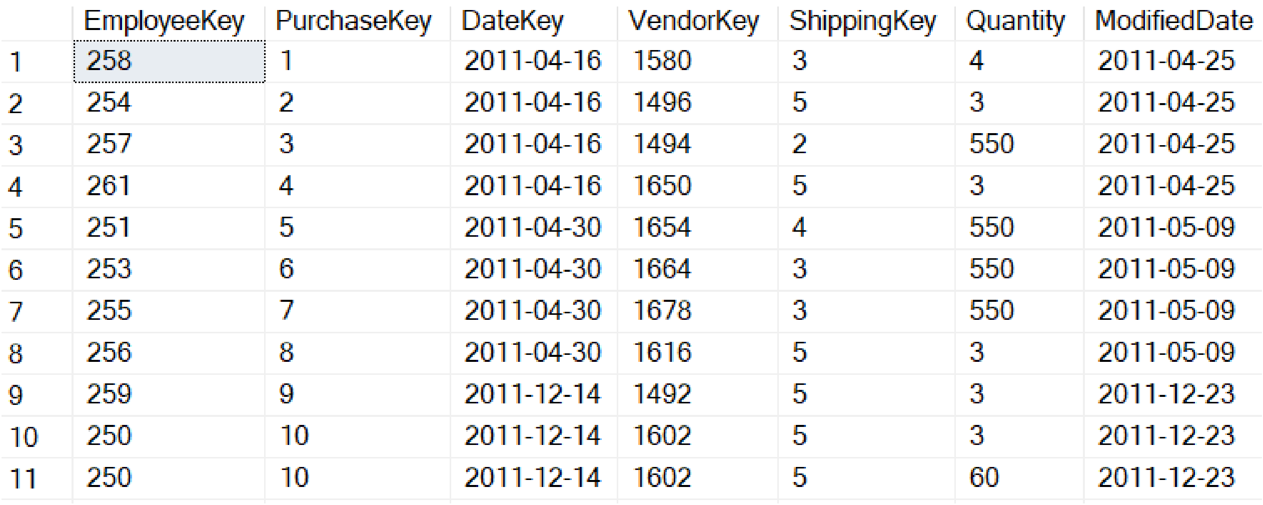
Purpose: to synthesize information on purchase and sale invoices with partners supplying spare parts.

No.	Field Name	Description	Type	Key
1	PurchaseKey	Purchase order number, selected as the primary key	int	PK

2	EmployeeKey	Employee ID	int	FK1
3	DateKey	Order date	date	FK2
4	VendorKey	Vendor ID	int	FK3
5	ShippingKey	Ship Method code	int	FK4
6	Quantity	Quantity ordered	smallint	
7	ModifiedDate	Date and time the record was last updated	date	

Table 15: FACT Purchasing Table

```
SELECT DISTINCT PUPOH.EMPLOYEEID AS EMPLOYEEKEY
       , PUPOD.PURCHASEORDERID AS PURCHASEKEY
       , CAST(ORDERDATE AS DATE) AS DATEKEY
       , PUPOH.VENDORID AS VENDORKEY
       , PURSHIP.SHIPMETHODID AS SHIPPINGKEY
       , PUPOD.ORDERQTY AS QUANTITY
       , CONVERT(DATE, PUPOH.MODIFIEDDATE ) AS MODIFIEDDATE
FROM [PURCHASING].[PURCHASEORDERHEADER] AS PUPOH
       JOIN [PURCHASING].[PURCHASEORDERDETAIL] AS PUPOD
ON PUPOD.PURCHASEORDERID = PUPOH.PURCHASEORDERID
       JOIN [PURCHASING].[SHIPMETHOD] AS PURSHIP
ON PURSHIP.SHIPMETHODID = PUPOH.SHIPMETHODID
ORDER BY DATEKEY
```



	EmployeeKey	PurchaseKey	DateKey	VendorKey	ShippingKey	Quantity	ModifiedDate
1	258	1	2011-04-16	1580	3	4	2011-04-25
2	254	2	2011-04-16	1496	5	3	2011-04-25
3	257	3	2011-04-16	1494	2	550	2011-04-25
4	261	4	2011-04-16	1650	5	3	2011-04-25
5	251	5	2011-04-30	1654	4	550	2011-05-09
6	253	6	2011-04-30	1664	3	550	2011-05-09
7	255	7	2011-04-30	1678	3	550	2011-05-09
8	256	8	2011-04-30	1616	5	3	2011-05-09
9	259	9	2011-12-14	1492	5	3	2011-12-23
10	250	10	2011-12-14	1602	5	3	2011-12-23
11	250	10	2011-12-14	1602	5	60	2011-12-23

Figure 16: Details of FACT Purchasing Table

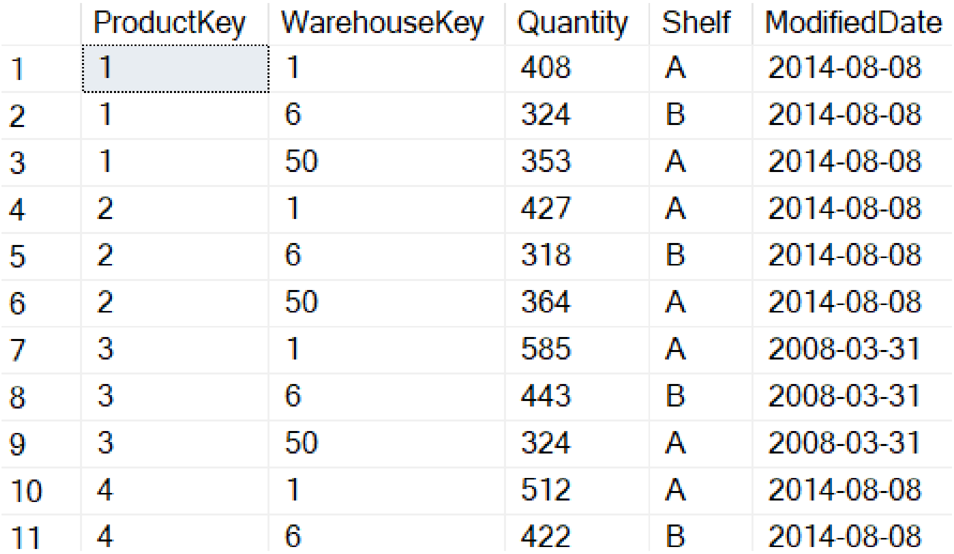
- **FACT Inventory Table**

Purpose: to aggregate information about existing products in the warehouse including warehouse location, storage place, quantity, ..

No.	Field Name	Description	Type	Key
1	ProductKey	Product identification number	int	PK,FK1
2	WarehouseKey	Inventory location identification number	smallint	PK,FK2
3	Quantity	Quantity of products in the inventory location	smallint	
4	Shelf	Storage compartment within an inventory location	nvarchar(10)	
5	ModifiedDate	Date and time the record was last updated	date	

Table 16: FACT Inventory Table

```sql
SELECT PRODUCTID AS PRODUCTKEY
       , LOCATIONID AS WAREHOUSEKEY
       , QUANTITY
       , SHELF
       , CONVERT(DATE, MODIFIEDDATE ) AS MODIFIEDDATE
FROM [PRODUCTION].[PRODUCTINVENTORY]
```



	ProductKey	WarehouseKey	Quantity	Shelf	ModifiedDate
1	1	1	408	A	2014-08-08
2	1	6	324	B	2014-08-08
3	1	50	353	A	2014-08-08
4	2	1	427	A	2014-08-08
5	2	6	318	B	2014-08-08
6	2	50	364	A	2014-08-08
7	3	1	585	A	2008-03-31
8	3	6	443	B	2008-03-31
9	3	50	324	A	2008-03-31
10	4	1	512	A	2014-08-08
11	4	6	422	B	2014-08-08

Figure 17: Details of FACT Inventory Table

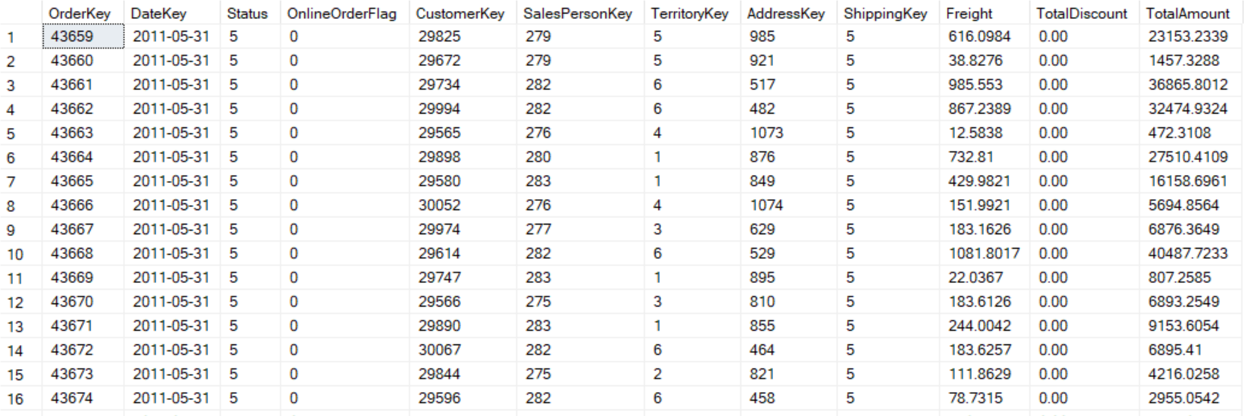
- **FACT SalesOrder Table**

No.	Field Name	Description	Type	Key
1	Orderkey	Primary key of Sales.SalesOrderHeader	int	PK
2	DateKey	Dates the sales order was created.	date	FK
f	Status	Name of State or Province	Varchar	
4	OnlineOrderFlag	0 = Order placed by sales person. 1 = Order placed online by customer.	flag DEFAULT 1	
5	CustomerKey	Customer ID	int	FK
6	SalesPersonKey	ID of Sale representative	int	FK
7	TerritoryKey	Territory ID	int	FK
8	AddressKey	Address ID	int	FK

9	ShippingKey	Shipping method ID	int	FK
10	Freight	Shipping cost	int	
11	TotalDiscount	Total discount percentage	int	
12	TotalAmount	Total product subtotal. Computed as UnitPrice * (1 - UnitPriceDiscount) * OrderQty.	numeric(38, 6)	

Table 17: FACT SalesOrder Table

```
SELECT [SALES].[SALESORDERHEADER].SALESORDERID AS ORDERKEY
        , CONVERT(DATE,ORDERDATE) AS DATEKEY
        , STATUS , ONLINEORDERFLAG
        , CUSTOMERID AS CUSTOMERKEY, SALESPERSONID AS SALESPERSONKEY
        , TERRITORYID AS TERRITORYKEY
        , SHIPTOADDRESSID AS ADDRESSKEY
        , SHIPMETHODID AS SHIPPINGKEY, FREIGHT
        , SUM([SALES].[SALESORDERDETAIL].[UNITPRICEDISCOUNT]) AS TOTALDISCOUNT
        , TOTALDUE AS TOTALAMOUNT
FROM [SALES].[SALESORDERHEADER]
        JOIN [SALES].[SALESORDERDETAIL]
ON [SALES].[SALESORDERDETAIL].SALESORDERID = [SALES].[SALESORDERHEADER].SALESORDERID
GROUP BY [SALES].[SALESORDERHEADER].SALESORDERID , ORDERDATE
        , STATUS , ONLINEORDERFLAG, CUSTOMERID, SALESPERSONID
        , TERRITORYID, SHIPTOADDRESSID, SHIPMETHODID, FREIGHT, TOTALDUE
```



	OrderKey	DateKey	Status	OnlineOrderFlag	CustomerKey	SalesPersonKey	TerritoryKey	AddressKey	ShippingKey	Freight	TotalDiscount	TotalAmount
1	43659	2011-05-31	5	0	29825	279	5	985	5	616.0984	0.00	23153.2339
2	43660	2011-05-31	5	0	29672	279	5	921	5	38.8276	0.00	1457.3288
3	43661	2011-05-31	5	0	29734	282	6	517	5	985.553	0.00	36865.8012
4	43662	2011-05-31	5	0	29994	282	6	482	5	867.2389	0.00	32474.9324
5	43663	2011-05-31	5	0	29565	276	4	1073	5	12.5838	0.00	472.3108
6	43664	2011-05-31	5	0	29898	280	1	876	5	732.81	0.00	27510.4109
7	43665	2011-05-31	5	0	29580	283	1	849	5	429.9821	0.00	16158.6961
8	43666	2011-05-31	5	0	30052	276	4	1074	5	151.9921	0.00	5694.8564
9	43667	2011-05-31	5	0	29974	277	3	629	5	183.1626	0.00	6876.3649
10	43668	2011-05-31	5	0	29614	282	6	529	5	1081.8017	0.00	40487.7233
11	43669	2011-05-31	5	0	29747	283	1	895	5	22.0367	0.00	807.2585
12	43670	2011-05-31	5	0	29566	275	3	810	5	183.6126	0.00	6893.2549
13	43671	2011-05-31	5	0	29890	283	1	855	5	244.0042	0.00	9153.6054
14	43672	2011-05-31	5	0	30067	282	6	464	5	183.6257	0.00	6895.41
15	43673	2011-05-31	5	0	29844	275	2	821	5	111.8629	0.00	4216.0258
16	43674	2011-05-31	5	0	29596	282	6	458	5	78.7315	0.00	2955.0542

Figure 18: Details of FACT SalesOrder Table

4.1.5. Data Warehouse model (Snowflake or Star)

4.2. ETL processes

4.2.1. Dimension Table’s ETL Process

Our Data Warehouse includes a 12-dimensional table and we would like to take Dim_SalesTerritory as an example.

At first, our database source is available in Microsoft SQL Server, therefore we chose "OLE DB Source".

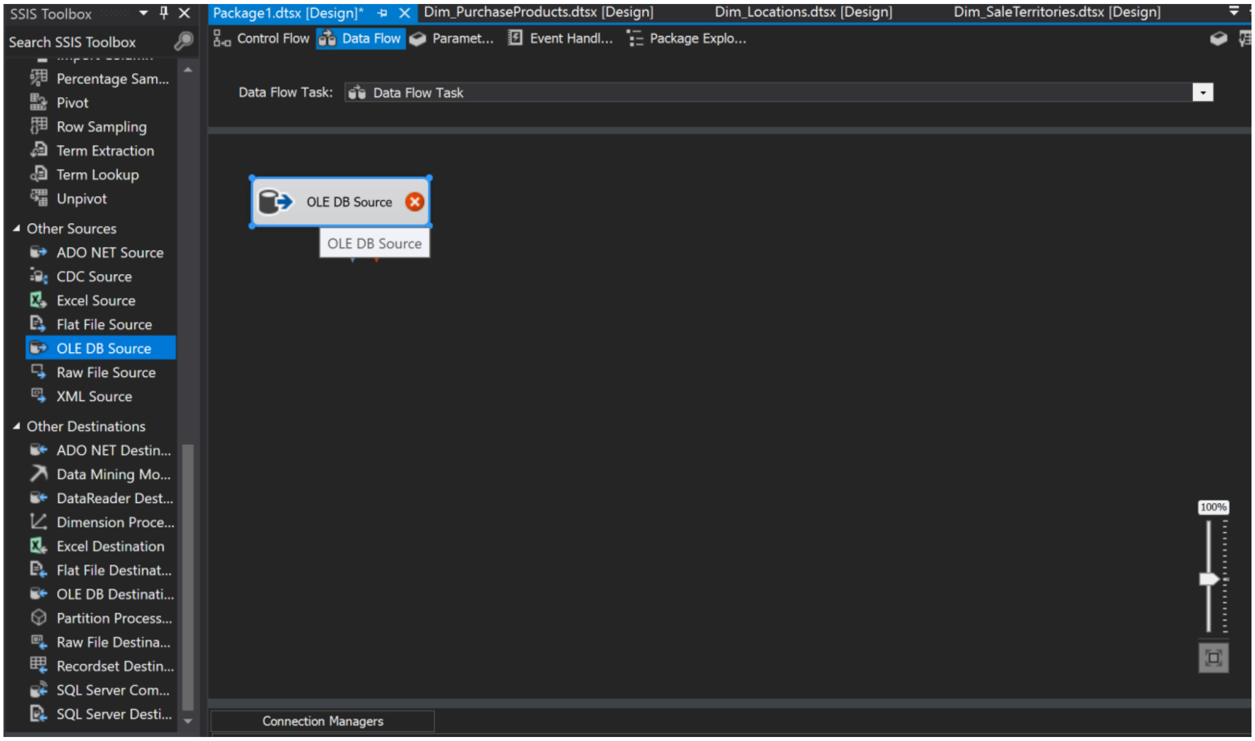
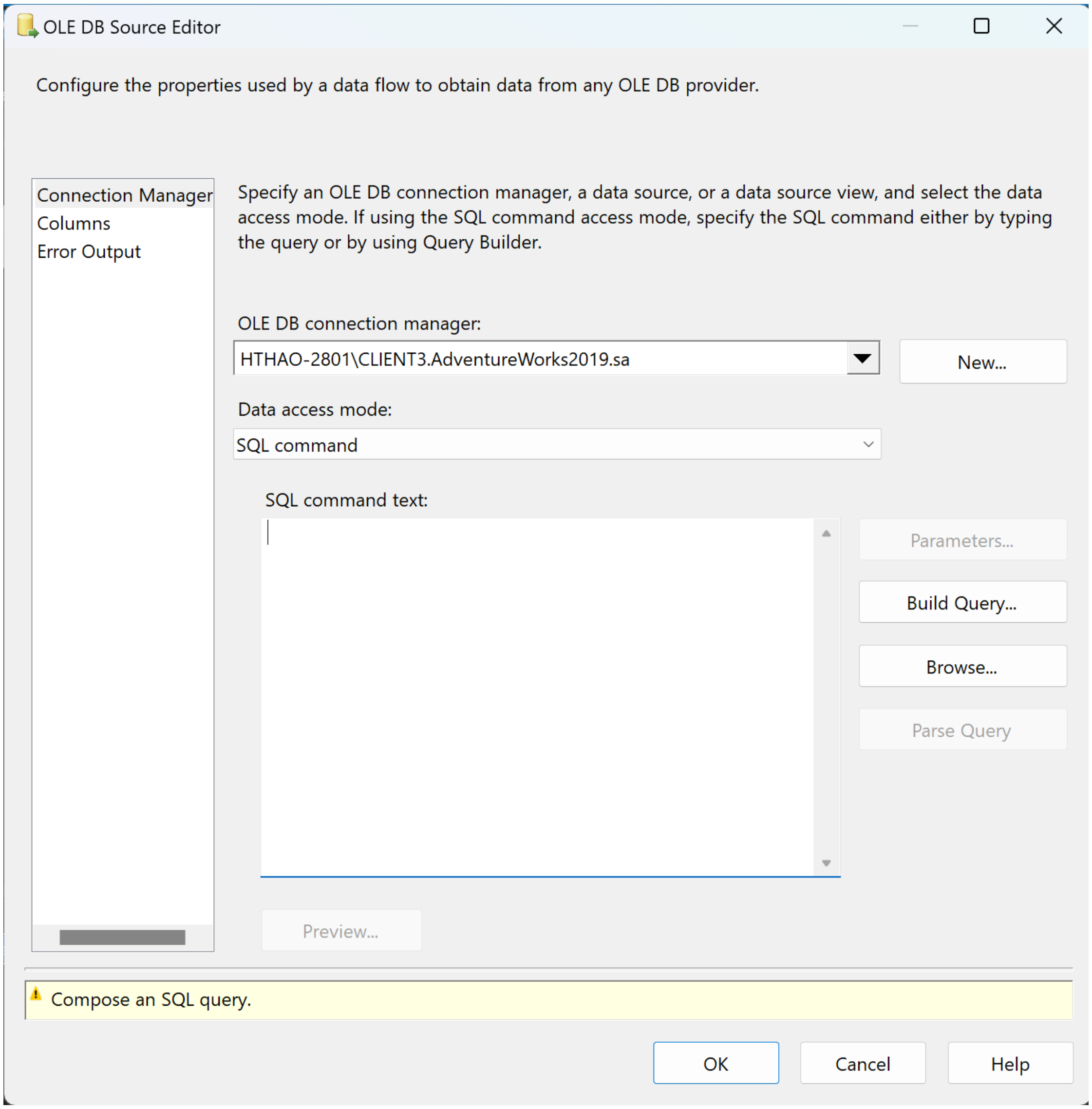


Figure 19. ETL process for Dim_SalesTerritory



OLE DB Source Editor

Configure the properties used by a data flow to obtain data from any OLE DB provider.

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.

OLE DB connection manager: HTHAO-2801\CLIENT3.AdventureWorks2019.sa

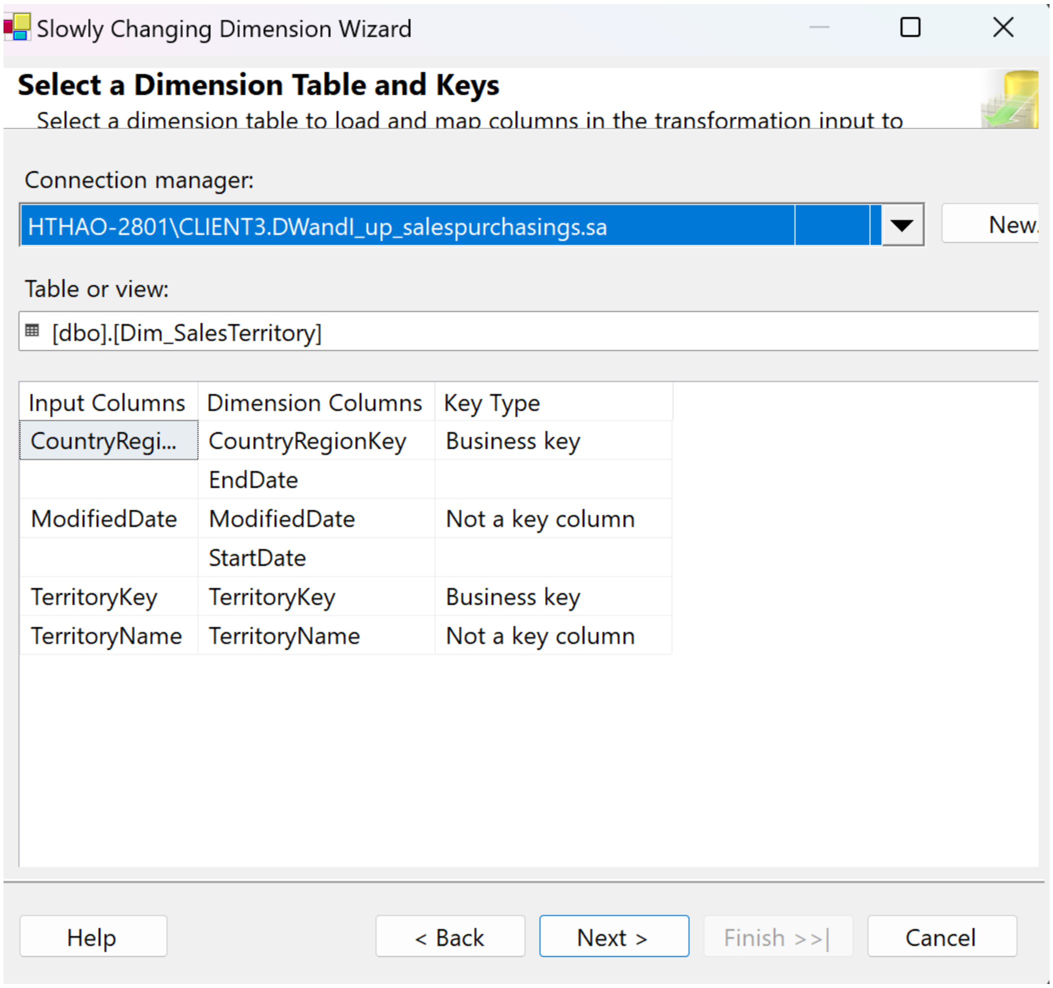
Data access mode: SQL command

SQL command text:

Compose an SQL query.

Figure 20. SQL command for Extraction data

After connecting Database source and using SQL query above, we can paste our SQL comment in the text box there. Next, we will use Slowly Changing Dimension (SCD) in Microsoft Visual Studio for loading data.



Slowly Changing Dimension Wizard

Select a Dimension Table and Keys

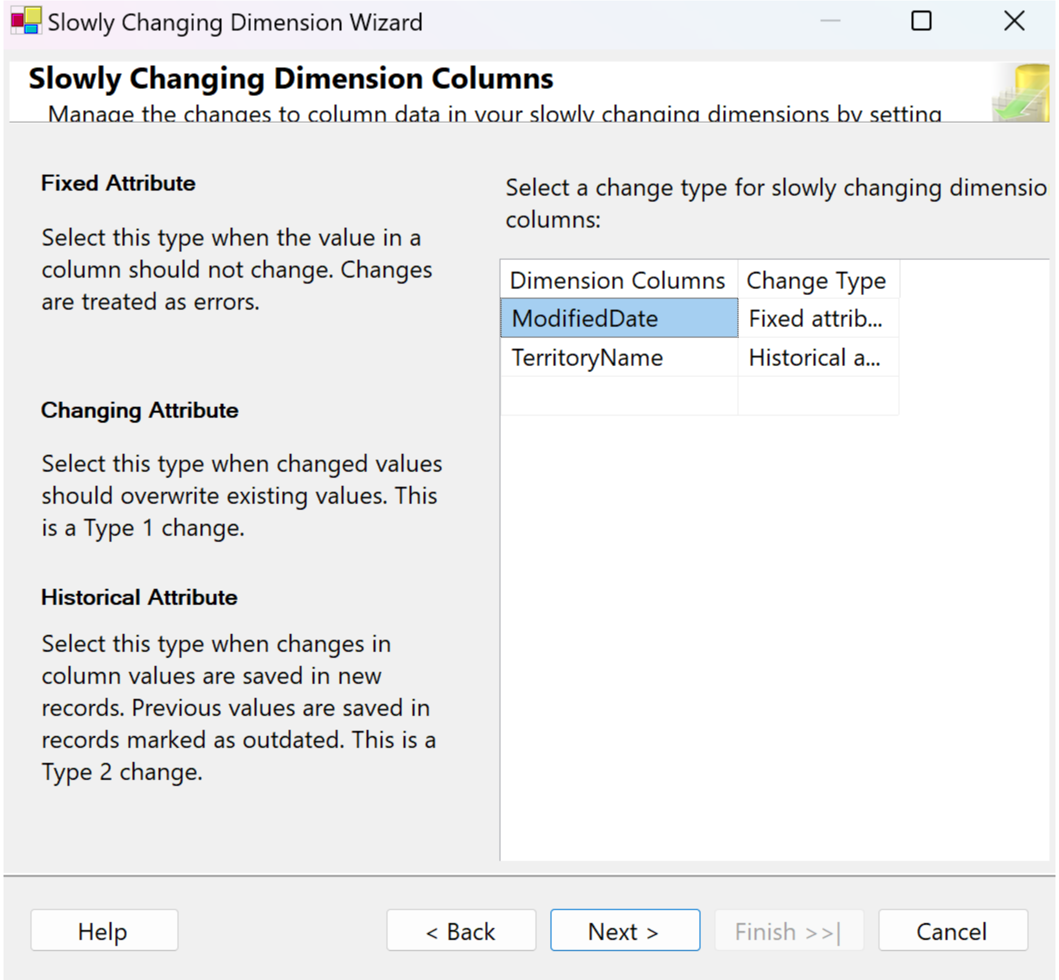
Select a dimension table to load and map columns in the transformation input to

Connection manager: HTHAO-2801\CLIENT3.DWandI_up_salespurchasings.sa

Table or view: [dbo].[Dim_SalesTerritory]

Input Columns	Dimension Columns	Key Type
CountryRegi...	CountryRegionKey	Business key
	EndDate	
ModifiedDate	ModifiedDate	Not a key column
	StartDate	
TerritoryKey	TerritoryKey	Business key
TerritoryName	TerritoryName	Not a key column

Figure 21. Set Business key for dimension tables



Slowly Changing Dimension Columns

Manage the changes to column data in your slowly changing dimensions by setting

Fixed Attribute

Select this type when the value in a column should not change. Changes are treated as errors.

Changing Attribute

Select this type when changed values should overwrite existing values. This is a Type 1 change.

Historical Attribute

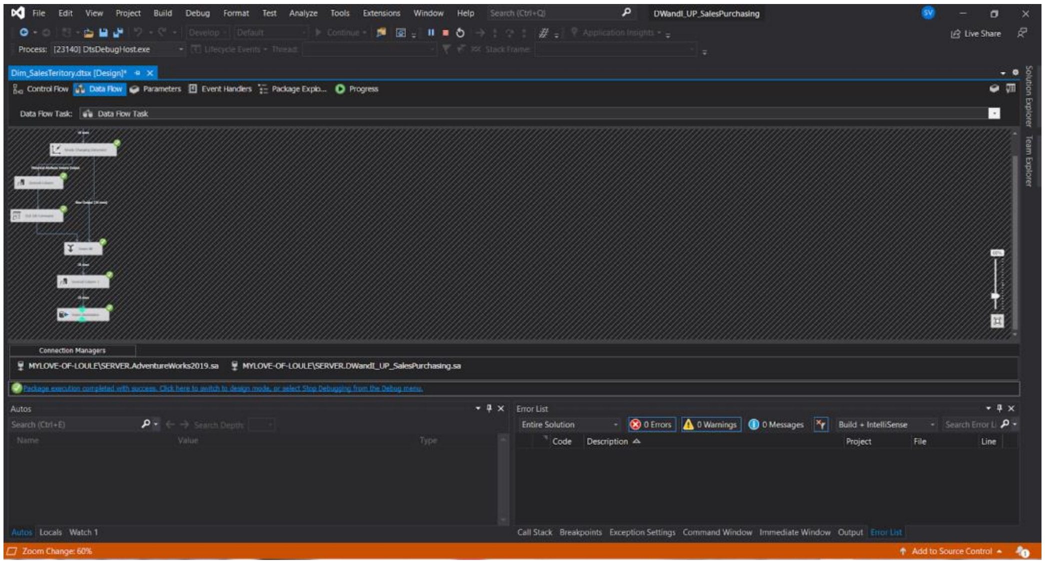
Select this type when changes in column values are saved in new records. Previous values are saved in records marked as outdated. This is a Type 2 change.

Select a change type for slowly changing dimensio columns:

Dimension Columns	Change Type
ModifiedDate	Fixed attrib...
TerritoryName	Historical a...

Figure 22. Select type change

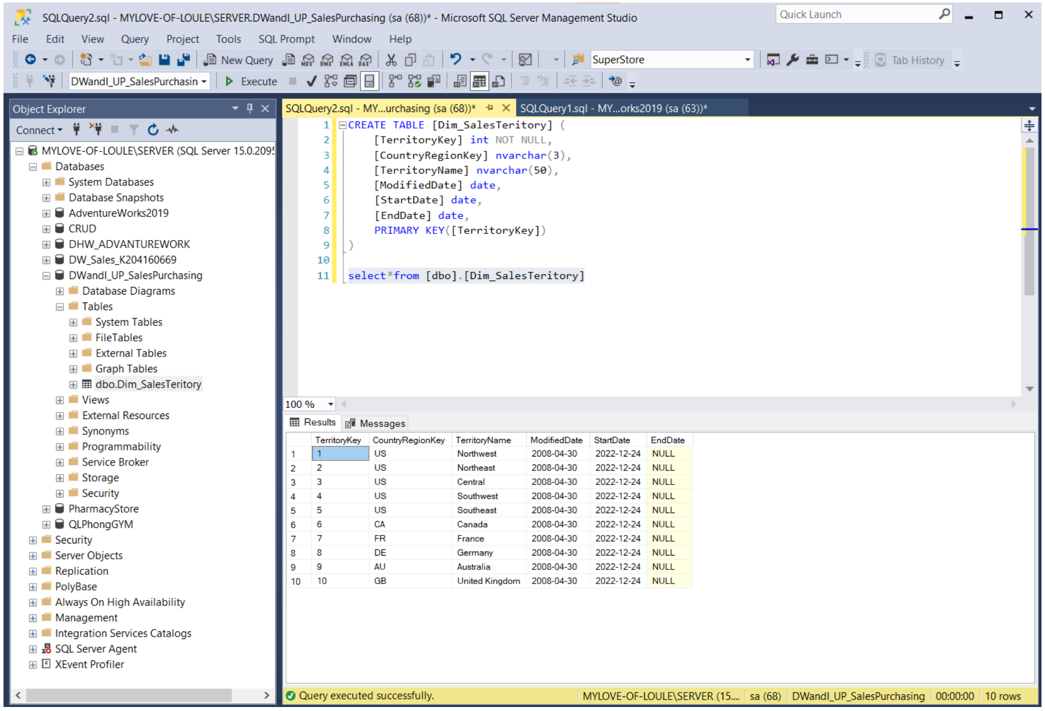
In this step, we selected type 0 for ModifiedDate attribute because it is hard to be changed and type 2 for TerritoryName, which must create Start and End values to update any changes.



MYLOVE-OF-LOULE\SERVER.AdventureWorks2019.sa   MYLOVE-OF-LOULE\SERVER.DWandI_UP_SalesPurchasing.sa

Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Figure 23. ETL data succeeded



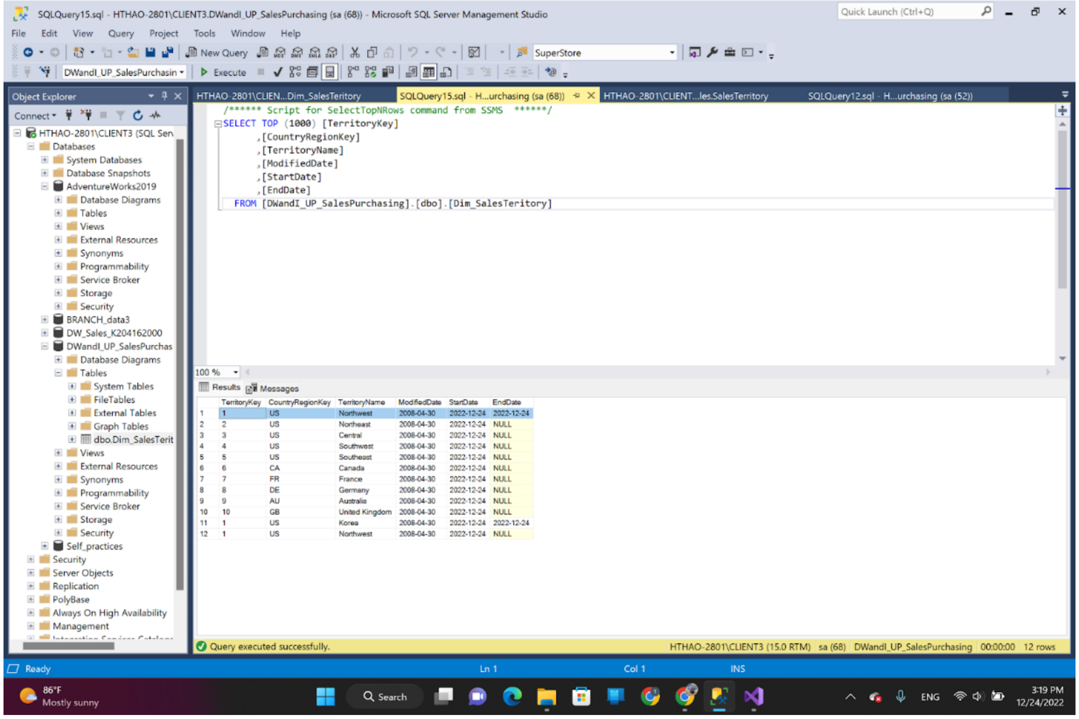
```sql
CREATE TABLE [Dim_SalesTeritory] (
    [TerritoryKey] int NOT NULL,
    [CountryRegionKey] nvarchar(3),
    [TerritoryName] nvarchar(50),
    [ModifiedDate] date,
    [StartDate] date,
    [EndDate] date,
    PRIMARY KEY([TerritoryKey])
)

select*from [dbo].[Dim_SalesTeritory]
```

	TerritoryKey	CountryRegionKey	TerritoryName	ModifiedDate	StartDate	EndDate
1	1	US	Northwest	2008-04-30	2022-12-24	NULL
2	2	US	Northeast	2008-04-30	2022-12-24	NULL
3	3	US	Central	2008-04-30	2022-12-24	NULL
4	4	US	Southwest	2008-04-30	2022-12-24	NULL
5	5	US	Southeast	2008-04-30	2022-12-24	NULL
6	6	CA	Canada	2008-04-30	2022-12-24	NULL
7	7	FR	France	2008-04-30	2022-12-24	NULL
8	8	DE	Germany	2008-04-30	2022-12-24	NULL
9	9	AU	Australia	2008-04-30	2022-12-24	NULL
10	10	GB	United Kingdom	2008-04-30	2022-12-24	NULL

Query executed successfully. | MYLOVE-OF-LOULE\SERVER (15.... | sa (68) | DWandI_UP_SalesPurchasing | 00:00:00 | 10 rows

Figure 24. Review result in Data Warehouse

To understand how SCD occurs, we would change value in database ([Sales].[SalesTerritory]).



```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [TerritoryKey]
      ,[CountryRegionKey]
      ,[TerritoryName]
      ,[ModifiedDate]
      ,[StartDate]
      ,[EndDate]
  FROM [DWandI_UP_SalesPurchasing].[dbo].[Dim_SalesTeritory]
```

	TerritoryKey	CountryRegionKey	TerritoryName	ModifiedDate	StartDate	EndDate
1	1	US	Northwest	2008-04-30	2022-12-24	2022-12-24
2	2	US	Northeast	2008-04-30	2022-12-24	NULL
3	3	US	Central	2008-04-30	2022-12-24	NULL
4	4	US	Southwest	2008-04-30	2022-12-24	NULL
5	5	US	Southeast	2008-04-30	2022-12-24	NULL
6	6	CA	Canada	2008-04-30	2022-12-24	NULL
7	7	FR	France	2008-04-30	2022-12-24	NULL
8	8	DE	Germany	2008-04-30	2022-12-24	NULL
9	9	AU	Australia	2008-04-30	2022-12-24	NULL
10	10	GB	United Kingdom	2008-04-30	2022-12-24	NULL
11	1	US	Korea	2008-04-30	2022-12-24	2022-12-24
12	1	US	Northwest	2008-04-30	2022-12-24	NULL

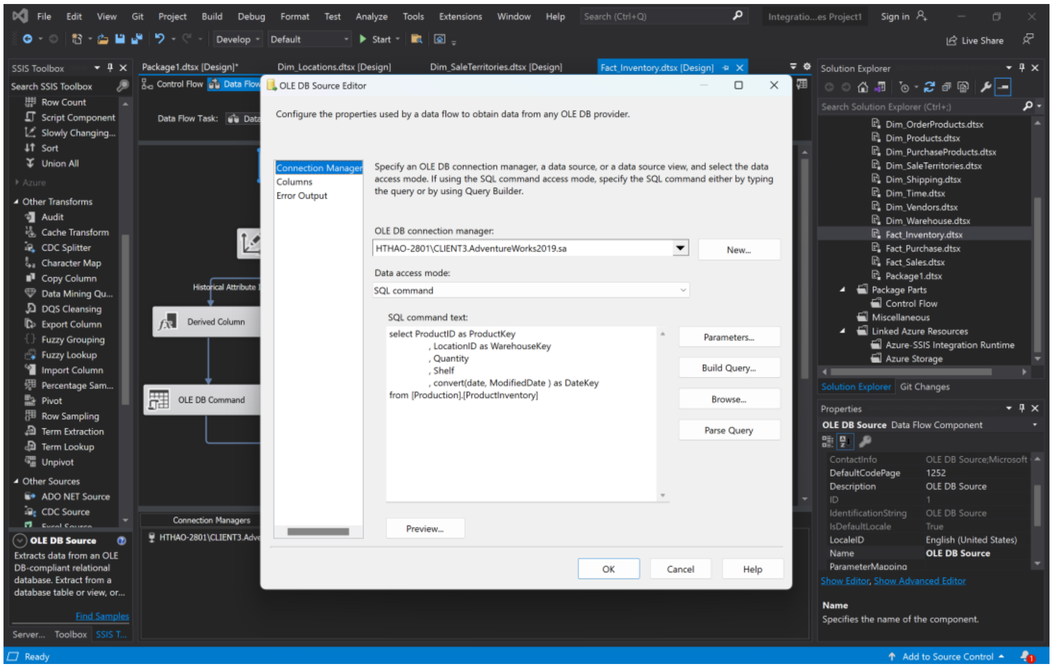
Query executed successfully.

Figure 25. Test for SCD

At first, we changed the territory name of the first row in AdventureWorks2019 from "Northwest" to Korea, then refreshed the table in our data warehouse. The data warehouse table has been updated with one additional row for each change in the database. And "Korea" (line 11) was added with the specific modification time written on StartDate.

4.2.2. Fact Table’s ETL Process

Similar to dimension table, we have three Fact tables in this project and we would like to take Fact_Sales for example.



Configure the properties used by a data flow to obtain data from any OLE DB provider.

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.

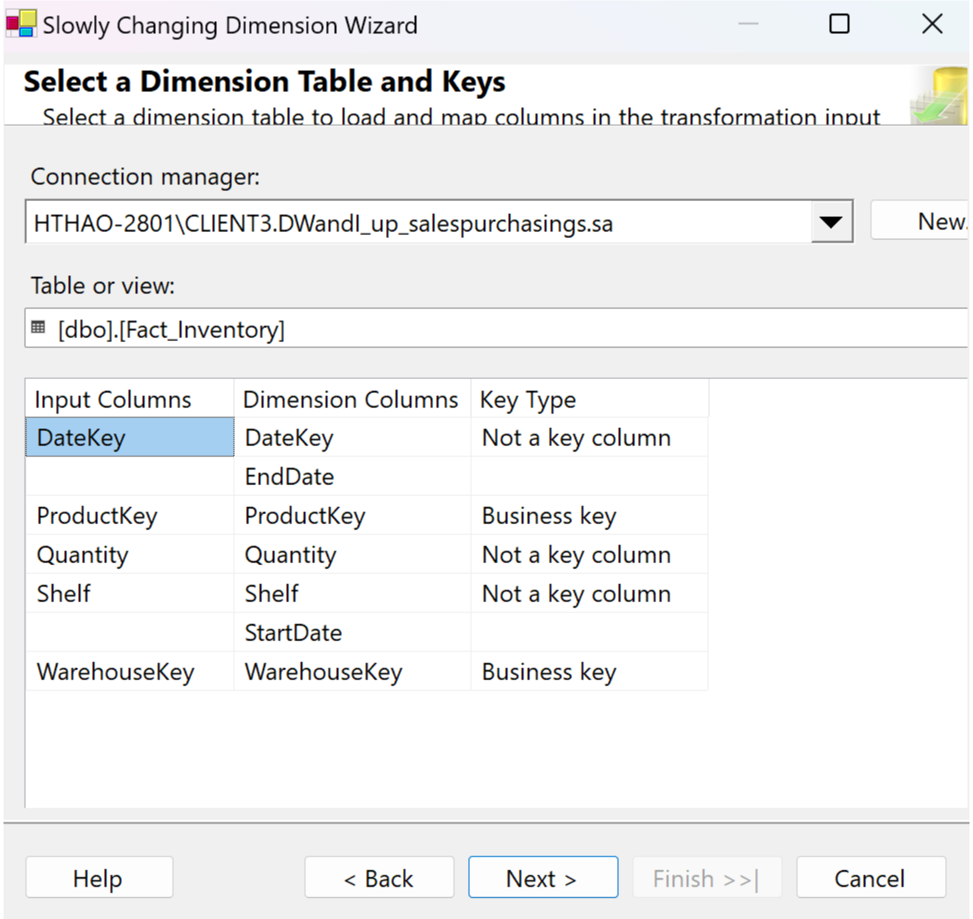
OLE DB connection manager: HTHAO-2801\CLIENT3.AdventureWorks2019.sa

Data access mode: SQL command

SQL command text:

```
select ProductID as ProductKey
          , LocationID as WarehouseKey
          , Quantity
          , Shelf
          , convert(date, ModifiedDate ) as DateKey
from [Production].[ProductInventory]
```

Figure 26. SQL command for Extraction data



Slowly Changing Dimension Wizard

Select a Dimension Table and Keys

Select a dimension table to load and map columns in the transformation input

Connection manager:

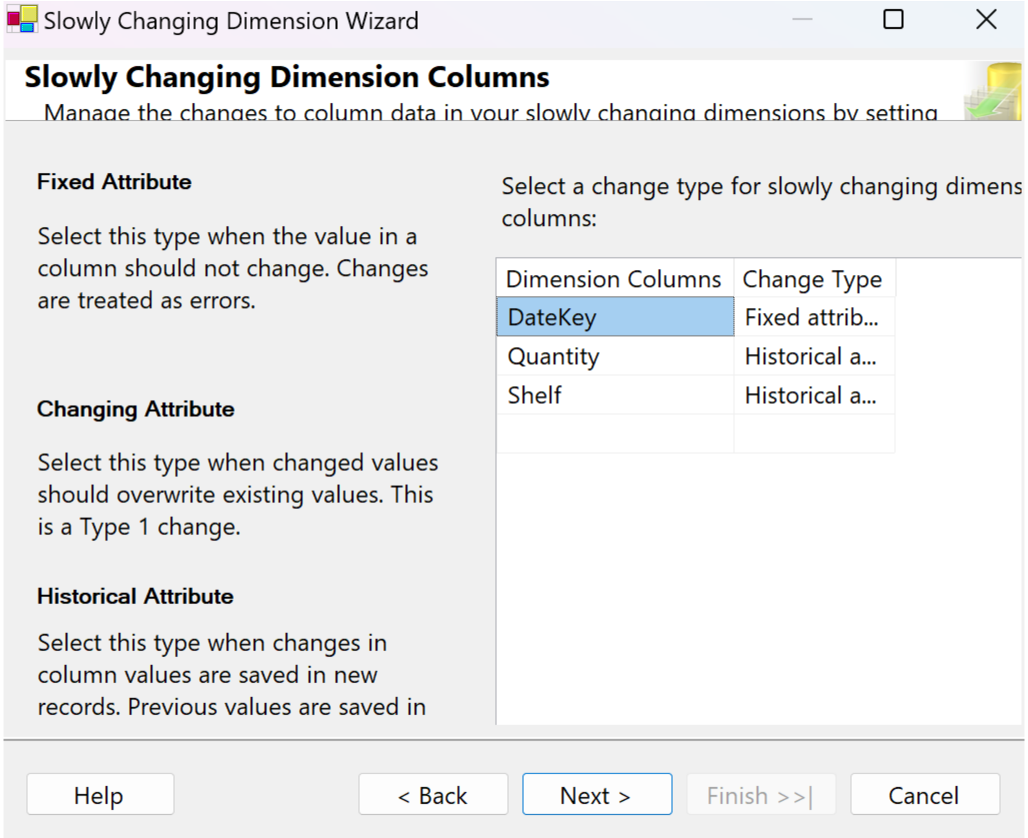
HTHAO-2801\CLIENT3.DWandI_up_salespurchasings.sa

Table or view:

[dbo].[Fact_Inventory]

Input Columns	Dimension Columns	Key Type
DateKey	DateKey	Not a key column
	EndDate	
ProductKey	ProductKey	Business key
Quantity	Quantity	Not a key column
Shelf	Shelf	Not a key column
	StartDate	
WarehouseKey	WarehouseKey	Business key

Figure 27. Business Key for Fact table



Slowly Changing Dimension Wizard

Slowly Changing Dimension Columns

Manage the changes to column data in your slowly changing dimensions by setting

Fixed Attribute

Select this type when the value in a column should not change. Changes are treated as errors.

Changing Attribute

Select this type when changed values should overwrite existing values. This is a Type 1 change.

Historical Attribute

Select this type when changes in column values are saved in new records. Previous values are saved in

Select a change type for slowly changing dimens columns:

Dimension Columns	Change Type
DateKey	Fixed attrib...
Quantity	Historical a...
Shelf	Historical a...

Figure 28. Select types of changing

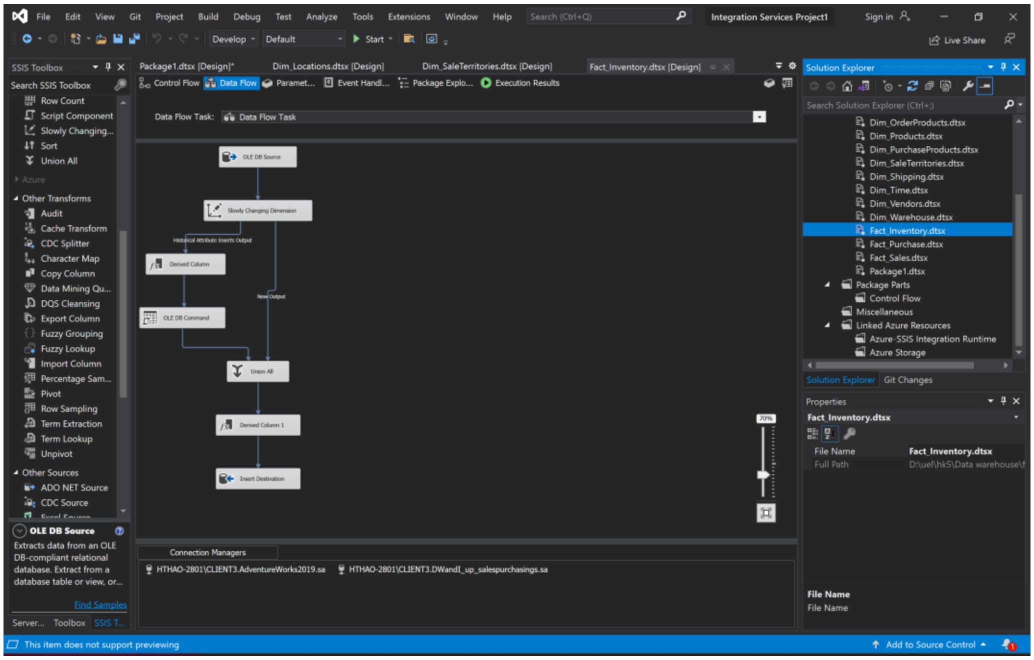
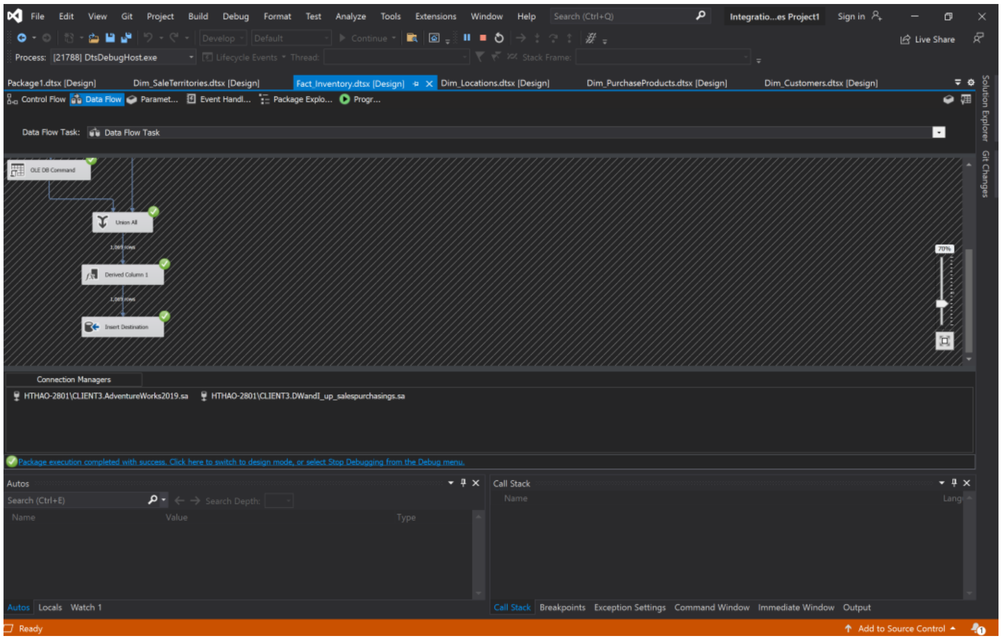


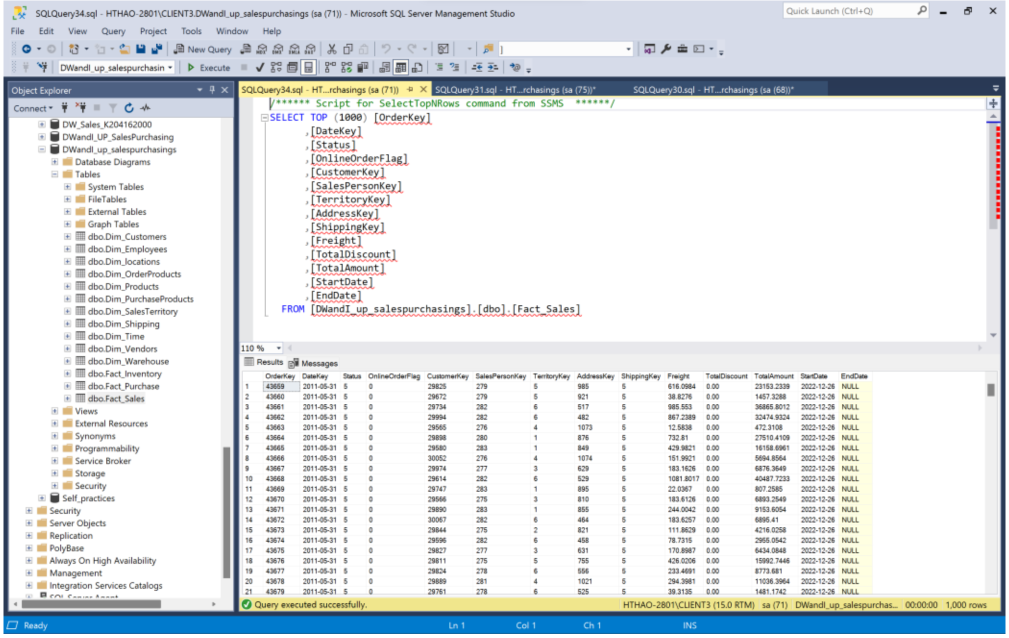
Figure 29. SCD flows



HTHAO-2801\CLIENT3.AdventureWorks2019.sa   HTHAO-2801\CLIENT3.DWandI_up_salespurchasings.sa

Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Figure 30. Loading succeeded


SQLQuery34.sql - HTHAO-2801\CLIENT3.DWandI_up_salespurchasings (sa (71)) - Microsoft SQL Server Management Studio
Object Explorer
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [OrderKey]
      ,[DateKey]
      ,[Status]
      ,[OnlineOrderFlag]
      ,[CustomerKey]
      ,[SalesPersonKey]
      ,[TerritoryKey]
      ,[AddressKey]
      ,[ShippingKey]
      ,[Freight]
      ,[TotalDiscount]
      ,[TotalAmount]
      ,[StartDate]
      ,[EndDate]
  FROM [DWandI_up_salespurchasings].[dbo].[Fact_Sales]
```

	OrderKey	DateKey	Status	OnlineOrderFlag	CustomerKey	SalesPersonKey	TerritoryKey	AddressKey	ShippingKey	Freight	TotalDiscount	TotalAmount	StartDate	EndDate
1	43659	2011-05-31	5	0	29825	279	5	985	5	616.0984	0.00	23153.2339	2022-12-26	NULL
2	43660	2011-05-31	5	0	29672	279	5	921	5	38.8276	0.00	1457.3288	2022-12-26	NULL
3	43661	2011-05-31	5	0	29734	282	6	517	5	985.553	0.00	36865.8012	2022-12-26	NULL
4	43662	2011-05-31	5	0	29994	282	6	482	5	867.2389	0.00	32474.9324	2022-12-26	NULL
5	43663	2011-05-31	5	0	29565	276	4	1073	5	12.5838	0.00	472.3108	2022-12-26	NULL
6	43664	2011-05-31	5	0	29898	280	1	876	5	732.81	0.00	27510.4109	2022-12-26	NULL
7	43665	2011-05-31	5	0	29580	283	1	849	5	429.9821	0.00	16158.6961	2022-12-26	NULL
8	43666	2011-05-31	5	0	30052	276	4	1074	5	151.9921	0.00	5694.8564	2022-12-26	NULL
9	43667	2011-05-31	5	0	29974	277	3	629	5	183.1626	0.00	6876.3649	2022-12-26	NULL
10	43668	2011-05-31	5	0	29614	282	6	529	5	1081.8017	0.00	40487.7233	2022-12-26	NULL
11	43669	2011-05-31	5	0	29747	283	1	895	5	22.0367	0.00	807.2585	2022-12-26	NULL
12	43670	2011-05-31	5	0	29566	275	3	810	5	183.6126	0.00	6893.2549	2022-12-26	NULL
13	43671	2011-05-31	5	0	29890	283	1	855	5	244.0042	0.00	9153.6054	2022-12-26	NULL
14	43672	2011-05-31	5	0	30067	282	6	464	5	183.6257	0.00	6895.41	2022-12-26	NULL
15	43673	2011-05-31	5	0	29844	275	2	821	5	111.8629	0.00	4216.0258	2022-12-26	NULL
16	43674	2011-05-31	5	0	29596	282	6	458	5	78.7315	0.00	2955.0542	2022-12-26	NULL
17	43675	2011-05-31	5	0	29827	277	3	631	5	170.8987	0.00	6434.0848	2022-12-26	NULL
18	43676	2011-05-31	5	0	29811	275	5	755	5	426.0206	0.00	15992.7446	2022-12-26	NULL
19	43677	2011-05-31	5	0	29824	278	6	556	5	233.4691	0.00	8773.681	2022-12-26	NULL
20	43678	2011-05-31	5	0	29889	281	4	1021	5	294.3981	0.00	11036.3964	2022-12-26	NULL
21	43679	2011-05-31	5	0	29761	278	6	525	5	39.3135	0.00	1481.1742	2022-12-26	NULL

Query executed successfully.   HTHAO-2801\CLIENT3 (15.0 RTM)   sa (71)   DWandI_up_salespurchas...   00:00:00   1,000 rows

Figure 31. Review results

5. CONCLUSION AND FUTURE WORKS

5.1. Results

As a result of our project, we built a data warehouse extracted from the AdventureWork2019 database. This data warehouse is structured from data marts with data belonging to 2 modules: sales and purchasing, from which it is possible to fulfill the requirements related to the sales and purchase process originally set out.

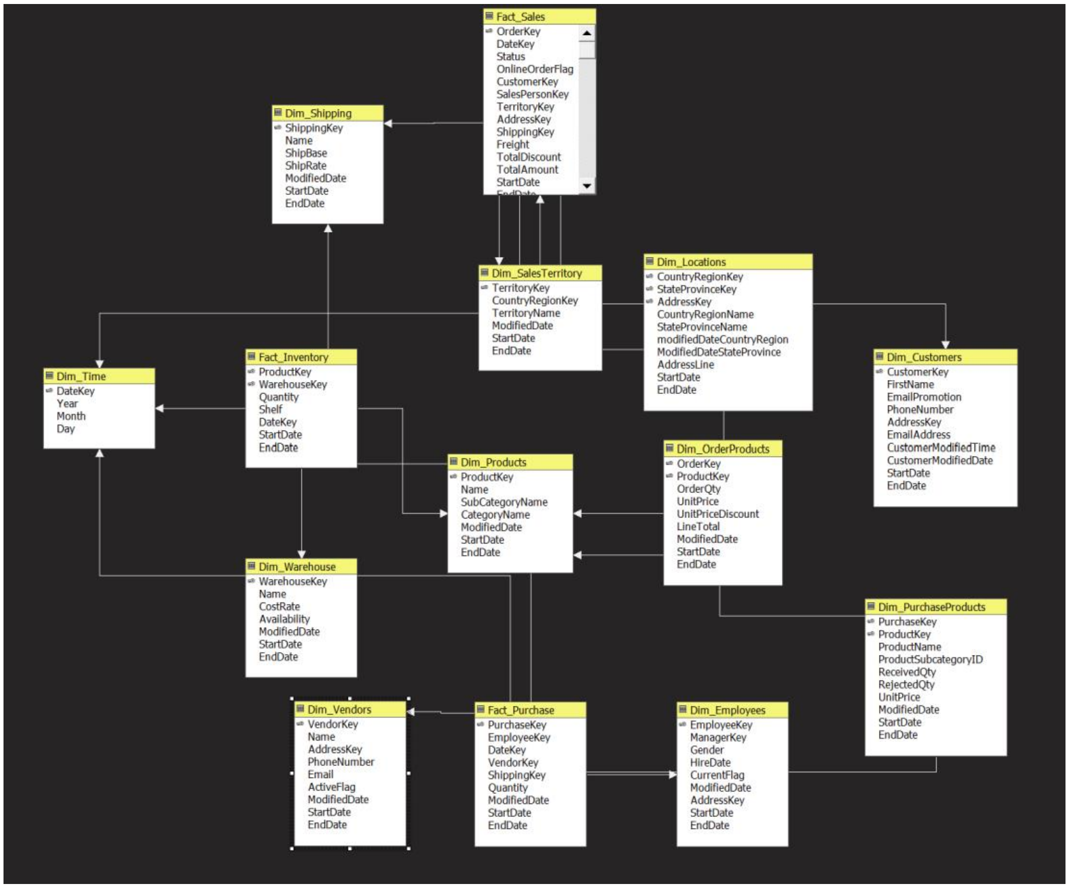


Figure 32. Cube view of Data Warehouse

5.2. Limitations

Because of the limited time and lack of experience in building a data warehouse before, the data warehouse that we have built is still quite small and still has a few non-optimal and inefficient points. Another limitation is that we do not really understand the AdventureWorks database because it is a rather large database that is combined with many different modules.

Although it has been thoroughly checked, it is certainly impossible to avoid shortcomings in this project, so we will try to improve. If there are any mistakes, we look forward to hearing from you and will correct them accordingly.

5.3. Future works

In the future, we will widen this data warehouse to not only sales and purchasing modules but also other modules such as Person, and Production modules. Besides that, we will try to improve the data warehouse more optimally and efficiently.

REFERENCES

(2022, 12 25). Retrieved from Tableau: https://www.tableau.com/learn/articles/business-intelligence/enterprise-business-intelligence/benefits

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM Sigmod record*, 26(1), 65-74.

Moody, D. L., & Kortink, M. A. . (2000). From enterprise models to dimensional models: a methodology for data warehouse and data mart design. *In DMDW (p. 5)*.

Teamques10. (2022, 12 25). *Quest10*. Retrieved from Quest10: https://www.ques10.com/p/43662/top-down-and-bottom-up-approaches-for-building-dat/

APPENDIX

https://www.guru99.com/star-snowflake-data-warehousing.html

https://www.javatpoint.com/data-warehouse-star-schema-vs-snowflake-schema