SSIS Exploration Document

Inventory and Sales Management System
Project business intelligence

Avishay Aknin

CONTENT

1   Document Management

1.1   Project Scope & Objective

The purpose of the project is to provide Data Warehouse Solution Creation from AdventureWorks2019 Database for AvishayDB, A company that supplies products and conducts business transactions.

This project aims to establish a comprehensive BI solution transferring data from the AdventureWorks2019 Database for SSIS_DB. The solution will contain summarized data tables, with a focus on sales data, employee records, customer information, product details.
Moreover, the project will solve the following issues:

- Providing a unified approach to organized and well-suited data for significant decision-making scenarios.

- Integrating data from different sources to be further used by reporting activities. The data is updated in real time, which makes it conveniently used for automated actions.

1.2   Project Content

In this project, we will build a Data Mart that will include information about many aspects of a real-world business order detailing from end-to-end perspective.

1. Data Cleaning and Preparation: Before the analysis starts, we filter our data via data cleaning method and preparation to verify their quality and consistency.

2. Main summary tables that will be built for the company's demands:

   o **Sales_Fact_table**– Contains information about all the orders, including dates, pricing and quantity per customer from the transactions of the transactional database.

   o **Stores_Dim**– Information about the company's Stores**.**

   o **Customer_Dim**– Information about the company's customers.

   o **Products_dim**- Information about the products that are sold by the company.

   o **Calendar_dim** - Information about date of transaction .

3. The project will contain measures that will lead into action to the achievement of the project's goal:

✔ **Sales Transaction orders Department:**

The Orders Department will focus on the sales-related data, when the order took place, identified by the customer that made the order, the store that activated the order. This department will contain a full description of the purchase including the productID, its price and the total amount per product in the order.
The department gathers information every time that a new order enters to the system and help the user to follow over the new transactions.

✔ **Store Department:**

Within this department, stores are tasked with initiating orders for specific customers, with each order being documented in the Sales_Fact_table table. Relevant stores details are housed in the Stores_Dim table. Each store is distinguished by a unique identifier (StoreId), establishing a direct connection to the Sales_Fact_table table for efficient order management.

✔ **Customer Department:**

A customer can make a new order, all the data about the customer will be stored inside the dimCustomers table – including an identifier (CustomerID) that is connected to the Sales_Fact_table table.

✔ **Production Department:**

A table that contains all the manufactured product in the company, every product has a description about it. It includes an identifier (ProductID) which is connected to Sales_Fact_table Table and will take part inside the transaction process.

2 Technical Specifications

2.1 Prerequisites – the involved systems during the process

System / Process	Explanation
SQL Server	Operational DB – tables – data (SQL Files)
SSIS	ETL processes using SSIS in Visual Studio
Data Refreshing	Refreshing processes through an attribute of Employees in SSMS

2.2 Source to Target and ERD models

2.3 Detailed description of ETL Process

### 1. Master Package-

Stage One: Data Deletion
The initial phase involves clearing existing data from dimension and FACT tables to establish a clean environment for subsequent operations. This is achieved by connecting to the database, executing SQL tasks to truncate or delete data from the tables, and ensuring seamless interaction between the SSIS package and the database.

Stage Two: Data Manipulation
In this stage, data manipulation occurs, focusing on loading data into dimension tables. A container is introduced for organizational clarity, within which "Data Flow Tasks" are created for each dimension. These tasks extract data from source views and load it into corresponding dimension tables, with field mappings ensuring data alignment.

Stage Three: Updating the FACT Table
The final stage revolves around updating the FACT table with processed data from the dimensions. A "Data Flow Task" is employed to transfer data from source views containing all dimensions and the FACT table. Field mappings are established to integrate data seamlessly into the FACT table, maintaining data integrity.
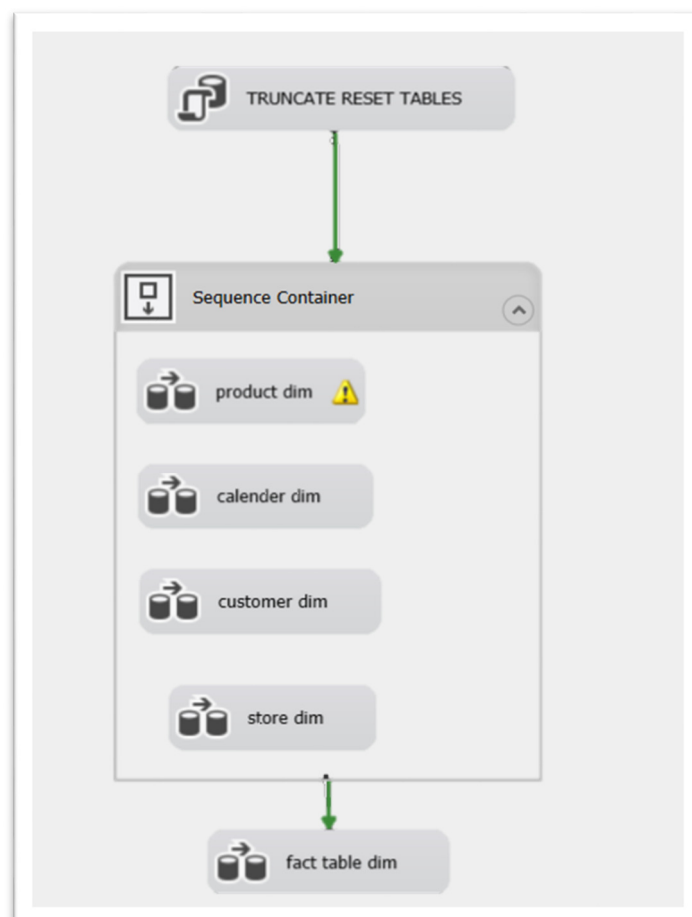
2. CSV TO DB -
Product table – insert just the new products

Loading the csv files: Retailers products(such as spring , tnuva) into Products tables.
The load action will load ONLY new records that did not appear in the 'products' table
before to avoid a case that we load duplicated data. All the new records will be counted.
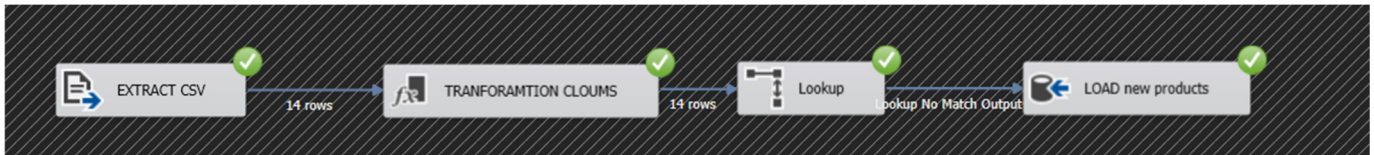
In this loop there is an exchange of the files that contain items of products in Excel files that
belong to each company - such as Castro, Tnuva, etc.
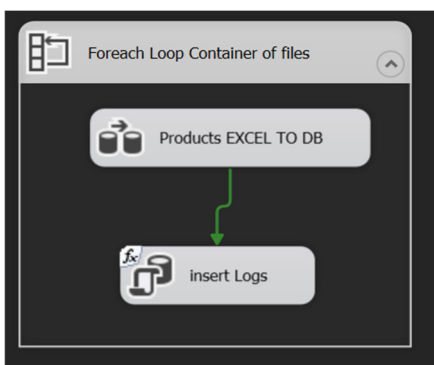
In addition, I built documentation of the logs of all the files, if there are new details there will be
documentation in the database "LogTable"
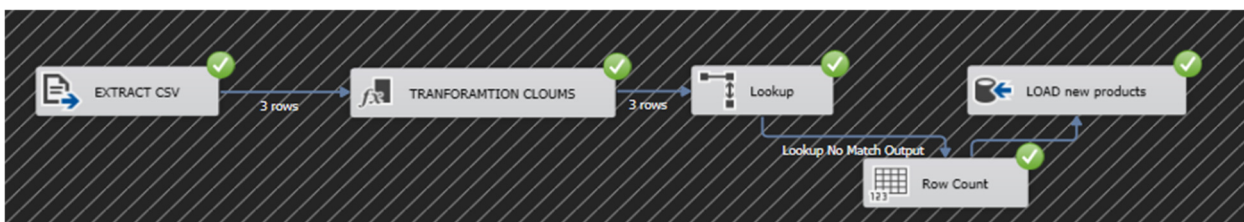
	id	status	file_name	records_insert	log_date
1	1	File loaded	C:\Users\מיכאל\Desktop\ssmsssispowerbi\databases...	3	2024-03-18 22:01:23.903
2	2	File loaded	C:\Users\מיכאל\Desktop\ssmsssispowerbi\databases...	6	2024-03-18 22:01:24.097
3	3	File loaded	C:\Users\מיכאל\Desktop\ssmsssispowerbi\databases...	1	2024-03-18 22:01:24.290

3.

DB TO DWH -
Product table – insert, delete, update just the relevant products rows

Packeage aim:
Ensure synchronization of data between a database (db) table and a data warehouse
(dwh) table efficiently and accurately using SQL Server Integration Services (SSIS).

Levels Explained:

- Database and Data Warehouse Setup:

- Create tables in both the database (db) and data warehouse (dwh).
- Data Comparison and Sorting: Employ a Merge Join tool to compare data between
  the db and dwh tables. Utilize a Conditional Split tool to categorize changes such as
  new data, updates, or deletions of data.
- Detailed Data Analysis: Conduct in-depth analysis of existing data to identify any
  discrepancies between the db and dwh tables.
- Verify if the data in both tables aligns accurately.
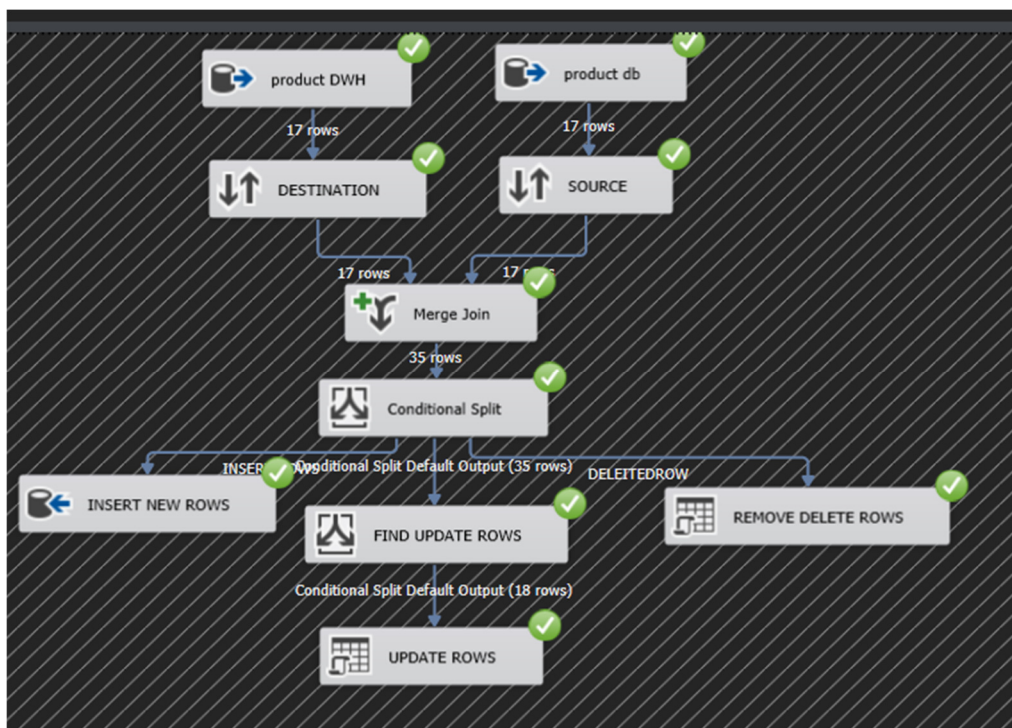
Furthermore, I have implemented a feature whereby all updated products are logged into a text file. This log file contains a comprehensive list of products that underwent updates on any given day.
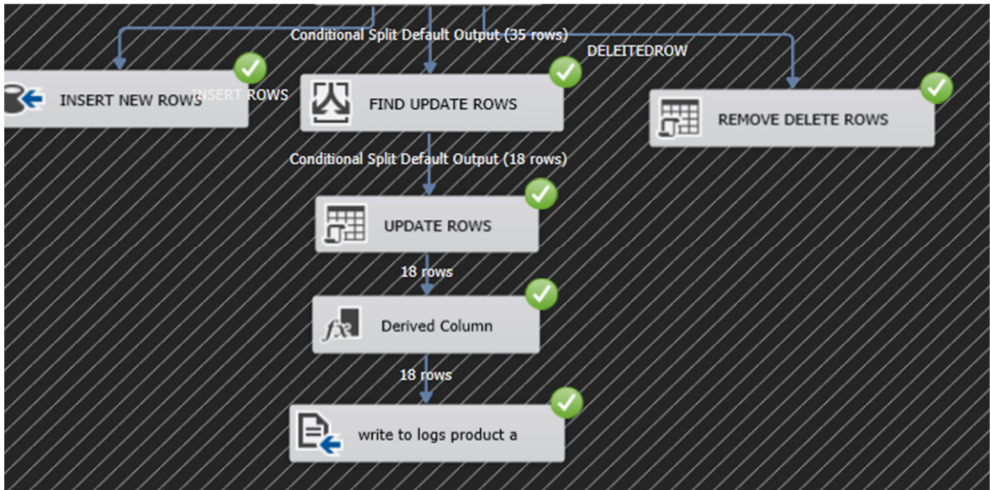
This it the logs products.txt file

```
1X1,Milk           ,11,111    ,0      ,Update row ,19/03/2024 17:39:25
1X1,Milk           ,11,111    ,0      ,Update row ,19/03/2024 17:39:25
2X2,yogurt         ,7,222     ,0      ,Update row ,19/03/2024 17:39:25
2X2,yogurt         ,7,222     ,0      ,Update row ,19/03/2024 17:39:25
P044,coffe         ,12,111    ,0      ,Update row ,19/03/2024 17:39:25
P044,coffe         ,12,111    ,0      ,Update row ,19/03/2024 17:39:25
p342,underwear          ,17,V002   ,3      ,Update row ,19/03/2024 17:39:25
p342,underwear          ,17,V002   ,3      ,Update row ,19/03/2024 17:39:25
p342,underwear          ,17,V002   ,3      ,Update row ,19/03/2024 17:39:25
p999,socks         ,10,V002   ,3      ,Update row ,19/03/2024 17:39:25
p999,socks         ,10,V002   ,3      ,Update row ,19/03/2024 17:39:25
p999,socks         ,10,V002   ,3      ,Update row ,19/03/2024 17:39:25
P002,T-shirt       ,20,V002   ,3      ,Update row ,19/03/2024 17:39:25
P002,T-shirt       ,20,V002   ,3      ,Update row ,19/03/2024 17:39:25
P002,T-shirt       ,20,V002   ,3      ,Update row ,19/03/2024 17:39:25
P777,Pants         ,50,V002   ,3      ,Update row ,19/03/2024 17:39:25
P777,Pants         ,50,V002   ,3      ,Update row ,19/03/2024 17:39:25
P777,Pants         ,50,V002   ,3      ,Update row ,19/03/2024 17:39:25
```

4.

The next step is to deploy the Project

The deployed path : /SSISDB/BI33/PRODPROJ_SSIS_DB Into SSMS.
After that I create a job called MrrLoadingJob which will call the next job in line until all the jobs will be done.
The jobs are separated into small parts because if an error will occur, we want to detect it in the specific job and be more efficient instead of loading massive amount of data each time.

This is the chronological order of the jobs:

MrrLoadingJob → Cus_Stg2DimJob → Emp_Stg2DimJob → Prd_Stg2DimJob → Rtl_Stg2DimJob →

Str_Stg2DimJob → FactOrders_Stg2DWHJob → FactRetailOrders_Stg2DWHJob All the Extract-Transform-Load process will happen every Sunday at 04:00 AM.

MRRLoadingJob steps:

S...	Name	Type	On Su...	On Fail...
1	MrrFactOrders	SQL S...	Go to t...	Quit th...
2	mrrRetailOrders	SQL S...	Go to t...	Quit th...
3	mrrdimOrders	SQL S...	Go to t...	Quit th...
4	call2Cus_Stg2DimJob	Transa...	Quit th...	Quit th...

Cus_Stg2DimJob steps:

S...	Name	Type	On Su...	On Fail...
1	stgCustomers	SQL S...	Go to t...	Quit th...
2	dimCustomers	SQL S...	Go to t...	Quit th...
3	Call to Emp_Stg2DimJob	Transa...	Quit th...	Quit th...

Emp_Stg2DimJob steps:

S...	Name	Type	On Su...	On Fail...
1	stgEmployees	SQL S...	Go to t...	Quit th...
2	dimEmployees	SQL S...	Go to t...	Quit th...
3	Call Prd_Stg2DimJob	Transa...	Quit th...	Quit th...

Prd_Stg2DimJob steps:

S...	Name	Type	On Su...	On Fail...
1	StgProducts	SQL S...	Go to t...	Quit th...
2	DimProducts	SQL S...	Go to t...	Quit th...
3	Load dimProductHistoryDWH	SQL S...	Go to t...	Quit th...
4	Call Rtl_Stg2DimJob	Transa...	Quit th...	Quit th...

Rtl_Stg2DimJob steps:

S...	Name	Type	On Su...	On Fail...
1	DimRetailers	SQL S...	Go to t...	Quit th...
2	Call Str_Stg2DimJob	Transa...	Quit th...	Quit th...

Str_Stg2DimJob steps:

S...	Name	Type	On Su...	On Fail...
1	StgStores	SQL S...	Go to t...	Quit th...
2	dimStores	SQL S...	Go to t...	Quit th...
3	Call FactOrders_Stg2DWHJob	Transa...	Quit th...	Quit th...

FactOrders_Stg2DWHJOB steps:

S...	Name	Type	On Su...	On Fail...
1	StgOrders	SQL S...	Go to t...	Quit th...
2	FactOrders	SQL S...	Go to t...	Quit th...
3	Call FactRetOrders_Stg2DWHJob	Transa...	Quit th...	Quit th...

FactRetailOrders_Stg2DWHJOB steps:

S...	Name	Type	On Su...	On Fail...
1	StgRetailOrders	SQL S...	Go to t...	Quit th...
2	FactRetailOrders	SQL S...	Quit th...	Quit th...

2.4 in the DWH Tables + History tables

Tables in the DWH:

- FactOrders:

In the FactOrders table we have the following columns:

1. Order_ID: The ID of the order.
2. Order_Date: The date that the order was recorded.
3. Due_Date: The date that the order is due to the customer.
4. Ship_Date: The date that the order was shipped to the customer.
5. Customer_ID: The ID of the customer. Foreign Key to DimCustomers table.
6. SalesPersonID: The ID of the employee that created the order. Foreign Key to DimEmployees table.
7. TerritoryID: The ID of the place that the order took place at.
8. ProductID: The ID of the purchased product, Foreign Key to DimProducts Table.
9. Qty: The quantity of the selected product related to the specific order.
10. Price: The price of each unit of the purchased product.
11. Discount: The percentage of the discount in the given order for the given product.
12. Total: The total amount that the order was cost per product inside the order, including the unitprice, discount, quantity and an additional VAT payment in a rate of 17%.

	Order_ID	Order_Date	Due_Date	Ship_Date	Customer_ID	SalesPersonID	TerritoryID	ProductID	Qty	Price	Discount	Total
1	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	776	1	2024.994	0.00	2369.24
2	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	777	3	2024.994	0.00	7107.72
3	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	778	1	2024.994	0.00	2369.24
4	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	771	1	2039.994	0.00	2386.79
5	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	772	1	2039.994	0.00	2386.79
6	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	773	2	2039.994	0.00	4773.58
7	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	774	1	2039.994	0.00	2386.79
8	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	714	3	28.8404	0.00	101.22
9	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	716	1	28.8404	0.00	33.74
10	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	709	6	5.70	0.00	40.01
11	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	712	2	5.1865	0.00	12.13
12	43659	2011-05-31	2011-06-12	2011-06-07	29825	279	5	711	4	20.1865	0.00	94.47
13	43660	2011-05-31	2011-06-12	2011-06-07	29672	279	5	762	1	419.4589	0.00	490.76
14	43660	2011-05-31	2011-06-12	2011-06-07	29672	279	5	758	1	874.794	0.00	1023.50
15	43661	2011-05-31	2011-06-12	2011-06-07	29734	282	6	745	1	809.76	0.00	947.41
16	43661	2011-05-31	2011-06-12	2011-06-07	29734	282	6	743	1	714.7043	0.00	836.20

Query executed successfully.   DESKTOP-4DADJPM\MSSQLSERVER...   DESKTOP-4DADJPM\nulla ...   SSIS_DB   00:00:00   121,318 rows

- FactRetailOrders:

In the FactRetailOrders table we have the following columns:

1. OrderID: The ID of the order.
2. Store_ID: The ID of the store that made the order. Foreign Key to DimStores table
3. City: The city that the order took place at.
4. ProductID: The ID of the purchased product, Foreign Key to DimProducts Table.
5. UnitPrice: The price of each unit of the purchased product.
6. Quantity: The quantity of the selected product related to the specific order.
7. Discount: The percentage of the discount in the given order for the given product.
8. Total: The total amount that the order was cost per product inside the order, including the unitprice, discount, quantity and an additional VAT payment in a rate of 17%.
9. InsertDate: The date that the order was inserted to the system.

	orderid	StoreID	City	ProductID	UnitPrice	Quantity	Discount	Total	InsertDate
137	11031	70004	Jerusalem	NULL	18.00	45	0.00	947.70	2023-12-21 14:29:37.807
138	11031	70004	Jerusalem	NULL	6.00	80	0.00	561.60	2023-12-21 14:29:37.807
139	11031	70004	Jerusalem	NULL	4.00	21	0.00	98.28	2023-12-21 14:29:37.807
140	11031	70004	Jerusalem	NULL	33.00	20	0.00	772.20	2023-12-21 14:29:37.807
141	11031	70004	Jerusalem	NULL	21.00	16	0.00	393.12	2023-12-21 14:29:37.807
142	11064	70004	Jerusalem	NULL	39.00	77	0.00	3513.51	2023-12-21 14:29:37.807
143	11064	70004	Jerusalem	NULL	9.00	12	0.00	126.36	2023-12-21 14:29:37.807
144	11064	70004	Jerusalem	NULL	32.00	25	0.00	936.00	2023-12-21 14:29:37.807
145	11064	70004	Jerusalem	NULL	24.00	4	0.00	112.32	2023-12-21 14:29:37.807
146	11064	70004	Jerusalem	NULL	12.00	55	0.00	772.20	2023-12-21 14:29:37.807
147	11065	50002	Tel-Aviv	NULL	25.00	4	0.00	117.00	2023-12-21 14:29:37.807
148	11065	50002	Tel-Aviv	NULL	7.00	20	0.00	163.80	2023-12-21 14:29:37.807
149	11071	50002	Tel-Aviv	NULL	30.00	15	0.00	526.50	2023-12-21 14:29:37.807
150	11071	50002	Tel-Aviv	NULL	6.00	10	0.00	70.20	2023-12-21 14:29:37.807

- DimProducts:

In the DimProducts table we have the following columns:

1. productID: Behaves as a primary key of the table. Represents the unique ID of the product.
2. Product_Name: The name of the product.
3. Sub_Category_Name: The name of the subcategory that the product related to.
4. Category_Name: The name of the category that the product related to.
5. UpdateDate: A field that shows if a product has been deleted from the original database or not.
6. isActive: A field that shows if the product is still active for sales or not.

	productID	Product_Name	Sub_Category_Name	Category_Name	UpdateDate	isActive
502	997	Road-750 Black, 44	Road Bikes	Bikes	2014-02-06 10:01:36.827	Y
503	998	UpdateRoad750	Road Bikes	Bikes	2014-02-06 10:01:36.827	Y
504	999	Updatata22fg00000	Road Bikes	Bikes	2014-02-06 10:01:36.827	Y
505	1016	NULL	NULL	NULL	2023-12-21 20:26:50.623	N
506	1017	NULL	NULL	NULL	2023-12-21 20:26:50.623	N
507	1020	BanNOT5	Mountain Bikes	Bikes	2023-12-16 12:57:37.113	Y
508	1023	NULL	NULL	NULL	2023-12-21 20:26:50.623	N
509	1024	Banana5	Mountain Bikes	Bikes	2023-12-21 20:26:50.623	N
510	1025	Banana5	Mountain Bikes	Bikes	2024-01-02 18:43:56.180	N

- DimEmployees:

In the DimEmployees table we have the following columns:

1. emp_id: Behaves as a primary key of the table, Represents the unique ID of the employee.

2. First_Name: The first name of the employee.

3. Last_Name: The last name of the employee.

4. Job_Title: The job title of the employee.

5. Hire_Date: The date that the employee was hired to the company.

6. Phone_Number: The employee's phone number.

7. Email_Address: The employee's email address.

8. Territory_Name: The territory that the employee was assigned to.

9. isEngineer: A flag column, checks if the employee's title include Engineer field or not.

10. isActive: A flag column that displays if the employee is still in the company or not.

11. UpdateDate: The date that the table was updated per employee.

	emp_id	First_Name	Last_Name	Job_Title	Hire_Date	Phone_Number	Email_Address	Territory_Name	isEngineer	isActive	Updat
1	1	Ken	Sánchez	Chief Executive Officer	2009-01-14	697-555-0142	ken0@adventure-works.com	NULL	N	1	2023-
2	2	Terri	Duffy	Vice President of Engineering	2008-01-31	819-555-0175	terri0@adventure-works.com	NULL	Y	1	2023-
3	3	Roberto	Tamburello	Engineering Manager	2007-11-11	212-555-0187	roberto0@adventure-works.com	NULL	Y	1	2023-
4	4	Rob	Walters	Senior Tool Designer	2007-12-05	612-555-0100	rob0@adventure-works.com	NULL	N	1	2023-
5	5	Gail	Erickson	Design Engineer	2008-01-06	849-555-0139	gail0@adventure-works.com	NULL	Y	1	2023-
6	6	Jossef	Goldberg	Design Engineer	2008-01-24	122-555-0189	jossef0@adventure-works.com	NULL	Y	1	2023-
7	7	Dylan	Miller	Research and Development Manager	2009-02-08	181-555-0156	dylan0@adventure-works.com	NULL	N	1	2023-
8	8	Diane	Margheim	Research and Development Engineer	2008-12-29	815-555-0138	diane1@adventure-works.com	NULL	Y	1	2023-
9	9	Gigi	Matthew	Research and Development Engineer	2009-01-16	185-555-0186	gigi0@adventure-works.com	NULL	Y	1	2023-
10	10	Michael	Raheem	Research and Development Manager	2009-05-03	330-555-2568	michael6@adventure-works.com	NULL	N	1	2023-
11	11	Ovidiu	Cracium	Senior Tool Designer	2010-12-05	719-555-0181	ovidiu0@adventure-works.com	NULL	N	1	2023-
12	12	Thierry	D'Hers	Tool Designer	2007-12-11	168-555-0183	thierry0@adventure-works.com	NULL	N	1	2023-
13	13	Janice	Galvin	Tool Designer	2010-12-23	473-555-0117	janice0@adventure-works.com	NULL	N	1	2023-
14	14	Michael	Sullivan	Senior Design Engineer	2010-12-30	465-555-0156	michael8@adventure-works.com	NULL	Y	1	2023-
15	15	Sharon	Salavaria	sasasas	2011-01-18	970-555-0138	sharon0@adventure-works.com	NULL	N	1	2023-

Query executed successfully.   DESKTOP-4DADJPM\MSSQLSERVER...   DESKTOP-4DADJPM\nulla ...   SSIS_DB   00:00:00   291 rows

- DimCustomers:

In the DimCustomers table we have the following columns:

1. Customer_ID: Behaves as a primary key of the table, Represents the unique ID of customer.

2. Name: The full name of the customer.

3. Address: The address of the customer.

4. City: The city that the customer lives at.

5. Region: The religion that the customer is related to.

6. Country: The country of the customer.

7. UpdateDate: Column that checks when the customer's details were updated most recently.

	CustomerID	Name	Address	City	Region	Country	UpdateDate
1	11000	Jon Yang	3761 N. 14th St	Rockhampton	Queensland	Australia	2023-12-18 08:54:34.953
2	11001	Eugene Huang	2243 W St.	Seaford	Victoria	Australia	2023-12-18 15:28:59.900
3	11002	Ruben Torres	5844 Linden Land	Hobart	Tasmania	Australia	2023-12-17 15:28:59.900
4	11003	Christy Zhu	1825 Village Pl.	North Ryde	New South Wales	Australia	2023-12-17 15:28:59.900
5	11004	Elizabeth Johnson	7553 Harness Circle	Wollongong	New South Wales	Australia	2023-12-17 15:28:59.900
6	11005	Julio Ruiz	7305 Humphrey Drive	East Brisbane	Queensland	Australia	2023-12-17 15:28:59.900
7	11006	Janet Alvarez	2612 Berry Dr	Matraville	New South Wales	Australia	2023-12-17 15:28:59.900
8	11007	Marco Mehta	942 Brook Street	Warrnambool	Victoria	Australia	2023-12-17 15:28:59.900
9	11008	Rob Verhoff	624 Peabody Road	Bendigo	Victoria	Australia	2023-12-17 15:28:59.900
10	11009	Shannon Carlson	3839 Northgate Road	Hervey Bay	Queensland	Australia	2023-12-17 15:28:59.900

1. TransferTable:

In the TransferTable table we have the following columns:

1. Packagename: The name of the recorded package.
2. Tablename: The name of the recorded table.
3. Start_Date: The date that the load started the action.
4. End_date: The date that the transaction finished the action.
5. Count: The amount of rows that were transferred in the specific transaction.

	PackageName	TableName	Start_Date	End_Date	Count
1	DWHProduct History.dtsx	dimProductsHistory	2024-01-03 18:19:40.000	2024-01-03 18:19:39.973	0
2	DimProducts.dtsx	DimProducts	2024-01-03 18:19:36.000	2024-01-03 18:19:36.313	0
3	stgProducts.dtsx	stgProducts	2024-01-03 18:19:31.000	2024-01-03 18:19:30.963	0
4	DimEmployees.dtsx	DimEmployees	2024-01-03 18:19:26.000	2024-01-03 18:19:26.300	0
5	stgEmployees.dtsx	stgEmployees	2024-01-03 18:19:23.000	2024-01-03 18:19:22.913	292
6	dimCustomers.dtsx	dimCustomers	2024-01-03 18:19:18.000	2024-01-03 18:19:18.270	0
7	stgCustomers.dtsx	stgCustomers	2024-01-03 18:19:14.000	2024-01-03 18:19:14.070	19119
8	mrrDimTables.dtsx	mrrRetailersStores	2024-01-03 18:19:09.000	2024-01-03 18:19:08.600	6
9	mrrDimTables.dtsx	mrrStores	2024-01-03 18:19:08.000	2024-01-03 18:19:08.510	701
10	mrrDimTables.dtsx	mrrBusinessEntityAddress	2024-01-03 18:19:08.000	2024-01-03 18:19:08.510	19614

- **DimProductsHistory:**

In the DimProductsHistory table we have the following columns:

1. productID: Behaves as a primary key of the table. Represents the unique ID of the product.
2. Product_Name: The name of the product.
3. Sub_Category_Name: The name of the subcategory that the product related to.
4. Category_Name: The name of the category that the product related to.
5. Insert_Date: A field that shows when the record was updated in the dimProducts table.
6. End_Date: A field that shows if a product has been deleted from the original database or not. NULL = Active Product, not null = product that is still active

	ProductID	ProductName	SubCategoryName	CategoryName	Insert_Date	End_Date
1	996	HL Bottom Bracket	Bottom Brackets	Components	2023-12-13 09:46:50.093	NULL
2	997	Road-750 Black, 44	Road Bikes	Bikes	2023-12-13 09:46:50.093	NULL
3	998	Road-750 Black, 48	Road Bikes	Bikes	2023-12-13 09:46:50.093	2023-12-13 11:15:18.000
4	998	UpdateRoad750	Road Bikes	Bikes	2023-12-13 11:15:18.000	NULL
5	999	Road-750 Black, 52	Road Bikes	Bikes	2023-12-13 09:46:50.093	2023-12-13 09:53:53.000
6	999	Updatata2200000	Road Bikes	Bikes	2023-12-13 09:53:53.000	2023-12-13 11:00:28.000
7	999	Updatata22fg00000	Road Bikes	Bikes	2023-12-13 11:00:28.000	2023-12-13 11:02:36.107

- **DimStores:**

In the DimStores table we have the following columns:

1. storeID: Behaves as a primary key of the table. Represents the unique ID of the store.
2. StoreName: The name of the store.
3. Emp_id: The id of the employee that related to the store, Represents as a Foreign Key to dimEmployees.
4. RetailerID: The id of the retailer that related to the store, Represents as a Foreign Key to dimRetailers.
5. City: The city that the store is located at.
6. isRetailer: A flag that checks if the related person is an employee or retailer.
7. UpdateDate: A field that shows if a product has been updated from the original database or not.
8. isActive: A field that shows if the store is still active for service or not.

698	1990	Technical Parts Manufacturing	281	NULL	San Francisco	N	2014-09-12 11:15:07.497	Y
699	1992	Totes & Baskets Company	277	NULL	Duluth	N	2014-09-12 11:15:07.497	Y
700	1994	World of Bikes	277	NULL	Duluth	N	2014-09-12 11:15:07.497	Y
701	2051	A Bicycle Association	275	NULL	Detroit	N	2014-09-12 11:15:07.497	Y
702	50001	Mega § Jerusalem	NULL	1	Jerusalem	Y	2024-01-02 14:03:46.770	Y
703	50002	Mega § Tel-Aviv	NULL	1	Tel-Aviv	Y	2024-01-02 14:03:46.770	Y
704	70001	Mashbir § Tel-Aviv	NULL	2	Tel-Aviv	Y	2024-01-02 14:03:46.770	Y
705	70002	Mashbir § Haifa	NULL	2	Haifa	Y	2024-01-02 14:03:46.770	Y
706	70003	Mashbir § Boise	NULL	2	Boise	Y	2024-01-02 14:03:46.770	Y
707	70004	Mashbir § Jerusalem	NULL	2	Jerusalem	Y	2024-01-02 14:03:46.770	Y

** Store can't have both Employee and Retailer that related to it.

- **DimRetailers:**

In the DimRetailers table we have the following columns:

1. Retailer_ID: Behaves as a primary key of the table, Represents the unique ID of a retailer.
2. RetailerName: The name of the retailer.
3. RetailerType: The type of the retailer.
4. isActive: A field that shows if the retailer is still active or not.
5. UpdateDate: Column that checks when the retailer's details were updated most recently.

	RetailerID	RetailerName	RetailerType	isActive	UpdateDate
1	1	Mega	Reshet	Y	2023-12-19 00:00:00.000
2	2	Mashbir	Reshet	Y	2023-12-19 00:00:00.000