**Table of Contents**

[**Table of Contents** 1](#_Toc88430485)

[**PART 1 - IN MEMORY DATA MODELLING – ETL & ANALYSIS:** 2](#_Toc88430486)

[**(a)** **TABLES – DATA MODEL:** 2](#_Toc88430487)

[**(b)** **DIAGRAM VIEW OF THE DATA MODEL:** 4](#_Toc88430488)

[**(c)** **LIST OF DAX FORMULAE:** 5](#_Toc88430489)

[**(d)** **SCREENSHOT OF EACH PIVOTTABLE:** 6](#_Toc88430490)

[**PIVOT TABLE 1: Top ranked products in terms of total sales amount, with a slicer by State:** 6](#_Toc88430491)

[**PIVOT TABLE 2: For 2018, the daily total sales and month to date total sales with Data bars:** 7](#_Toc88430492)

[**PIVOT TABLE 3 : Performance comparison in % month to month 2019 – 2020:** 8](#_Toc88430493)

[**(e)** **A DISCUSSION:** 9](#_Toc88430494)

[**PART 2: DATA MODELLING, ETL AND ANALYSIS WITH SSIS AND SSAS** 10](#_Toc88430495)

[**TASK 2.1: DATA MODELLING & ETL:** 10](#_Toc88430496)

[**2.1.1:** 10](#_Toc88430497)

[**2.1.2:** 10](#_Toc88430498)

[**Deliverables for Task 2.1:** 10](#_Toc88430499)

[**TASK 2.2: DATA ANALYSIS:** 11](#_Toc88430500)

[**2.2.1:** 11](#_Toc88430501)

[**2.2.2:** 11](#_Toc88430502)

[**DELIVERABLES FOR TASK 2.2:** 11](#_Toc88430503)

# **PART 1 - IN MEMORY DATA MODELLING – ETL & ANALYSIS:**

## **TABLES – DATA MODEL:**

| Table Name | No. of Rows | Type | List of Columns | | | Hierarchy Name | Hierarchy Columns | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Transactions | 70510 | Fact | 1 | **CustomerTransactionID** | **PK** | Amount | 1 | AmountExcludingTax |
| 2 | **CustomerID** | **FK** | 2 | TaxAmount |
| 3 | **StocketItemName** | **FK** | 3 | TransactionAmount |
| 4 | **TransactionDate** | **FK** |  | |
|  | TransactionDate | |
| 6 | SalesPersonName | |
| 7 | Quantity | |
| 8 | UnitPrice | |
| 9 | AmountExcludingTax | |
| 10 | TaxAmount | |
| 11 | TransactionAmount | |
|  | | | | | | | | |
| Customers | 663 | Transactions → Dimension  Cities → Fact | 1 | **CustomerID** | **PK** | Customer Address | 1 | DeliveryAddressLine1 |
| 2 | **PostalCityID** | **FK** | 2 | DeliveryAddressLine2 |
| 3 | CustomerName | | 3 | DeliveryPostalCode |
| 4 | PhoneNumber | |  | |
| 5 | DeliveryAddressLine1 | |
| 6 | DeliveryAddressLine2 | |
| 7 | DeliveryPostalCode | |
|  | | | | | | | | |
| StockProducts | 227 | Transactions → Dimension | 1 | **StocketItemName** | **PK** |  |  | |
| 2 | StockItemID | |
| 3 | UnitPrice | |
|  | | | | | | | | |
| Cities | 655 | Dimension | 1 | **PostalCityID** | **PK** | Place | 1 | CountryName |
| 2 | CityName | | 2 | Continent |
| 3 | StateProvinceName | | 3 | Region |
| 4 | CountryName | | 4 | Subregion |
| 5 | Continent | | 5 | Location |
| 6 | SalesTerritory | | 6 | Latitude |
| 7 | Region | | 7 | Longitude |
| 8 | Subregion | |  | |
| 9 | Location | |
| 10 | Latitude | |
| 11 | Longitude | |
| 12 | LatestRecordedPopulation | |
|  | | | | | | | | |
| Dates | 1069 | - Dimension for Transactions - Fact for TempDates | 1 | **TDate** | **PK** |  |  | |
| 2 | TDate (Month Index) | |
| 3 | TDate (Month) | |
| 4 | YEAR | |
| 5 | TDate (Day Index) | |
| 6 | TDate (Day) | |
| 7 | TDate (Year) | |
| 8 | TDate (Quarter) | |
|  | | | | | | | | |

## **DIAGRAM VIEW OF THE DATA MODEL:**

Graphical user interface

Description automatically generated with medium confidence

## **LIST OF DAX FORMULAE:**

### **DAX Formula - PivotTable1:**

To obtain rank of product by total sales, we use RANKX DAX Formula,

### **DAX Formula – PivotTable2:**

* To obtain monthly total sales we use MTD DAX formula,
* To obtain quarterly and yearly sales we use QTD and YTD formula,
* We can analyse daily total sales by updating PivotTable Fields

Graphical user interface, application

Description automatically generated

### **DAX Formula – PivotTable3:**

To obtain previous month’s total sales in 2020, we use DATEADD function in DAX,

## **SCREENSHOT OF EACH PIVOTTABLE:**

### **PIVOT TABLE 1: Top ranked products in terms of total sales amount, with a slicer by State:**

Graphical user interface

Description automatically generated with medium confidence

### **PIVOT TABLE 2: For 2018, the daily total sales and month to date total sales with Data bars:**

Graphical user interface, application, table, Excel

Description automatically generated

### **PIVOT TABLE 3 : Performance comparison in % month to month 2019 – 2020:**

Graphical user interface

Description automatically generated

## **A DISCUSSION:**

As per KinetEco’s business requirement to analyse transactions by customer’s location, product, and date to run number of reports, we are successfully able to analyse required data.

To seamlessly enter the data into any parent table from excel sheet. We can combine new data into the parent sheets seamlessly by normalising them into Transactions, products, customers, dates, and cities and transforming. Power Query will give us the flexibility to load the new data into it seamlessly.

**Strengths:**

* Our Datawarehouse is in almost normalised state. I wasn’t intended to normalise it before. Rather ended up in normalising the tables to get the analysation of data correctly.
* Queries can be simpler with simplified common business reporting logic.

**Weakness:**

* To calculate Time intelligence functions, I have normalised the date table and executed time intelligence calculations in it. When I was calling the date function from transaction table it wasn’t being identified.
* Normalised data avoids insert and updates anomalies. We cannot update this with surrogate keys likewise in SQL server manager to better normalise the tables as it has very limited unique columns to be a primary key.
* Data integrity is not well – enforced due to its de-normalised state.

# **PART 2: DATA MODELLING, ETL AND ANALYSIS WITH SSIS AND SSAS**

## **TASK 2.1: DATA MODELLING & ETL:**

### **Deliverables for Task 2.1:**

#### **SQL Script for creating the data warehouse tables:**

USE [KinetEco]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dates] Script Date: 03/12/2021 12:45:12 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dates](

[DatesID] [int] IDENTITY(1,1) NOT NULL,

[TransactionDate] [nvarchar](20) NOT NULL,

PRIMARY KEY CLUSTERED

(

[DatesID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dim\_Cities] Script Date: 03/12/2021 12:45:12 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dim\_Cities](

[CityKey] [int] IDENTITY(1,1) NOT NULL,

[CityName] [nvarchar](40) NULL,

[StateProvinceName] [nvarchar](40) NULL,

[CountryName] [nvarchar](40) NULL,

[Continent] [nvarchar](40) NULL,

[SalesTerritory] [nvarchar](40) NULL,

[Region] [nvarchar](40) NULL,

[Subregion] [nvarchar](40) NULL,

[Location] [nvarchar](255) NULL,

[Latitude] [int] NULL,

[Longitude] [int] NULL,

[LatestRecordedPopulation] [int] NULL,

[PostalCityID] [int] NULL,

PRIMARY KEY CLUSTERED

(

[CityKey] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dim\_Customers] Script Date: 03/12/2021 12:45:12 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dim\_Customers](

[CustomerKey] [int] IDENTITY(1,1) NOT NULL,

[CustomerID] [nvarchar](40) NULL,

[PhoneNumber] [nvarchar](40) NULL,

[DeliveryAddressLine1] [nvarchar](40) NULL,

[DeliveryAddressLine2] [nvarchar](40) NULL,

[DeliveryPostalCode] [nvarchar](20) NULL,

[PostalCityID] [nvarchar](20) NULL,

[CustomerName] [nvarchar](255) NULL,

PRIMARY KEY CLUSTERED

(

[CustomerKey] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Fact\_Transactions] Script Date: 03/12/2021 12:45:12 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Fact\_Transactions](

[CustomerKey] [int] NULL,

[DatesID] [int] NULL,

[CityKey] [int] NULL,

[CustomerTransactionID] [int] NULL,

[TransactionDate] [nvarchar](40) NULL,

[SalesPersonName] [nvarchar](40) NULL,

[StockItemID] [int] NULL,

[StockItemName] [nvarchar](255) NULL,

[Quantity] [int] NULL,

[UnitPrice] [int] NULL,

[AmountExcludingTax] [int] NULL,

[TaxAmount] [int] NULL,

[TransactionAmount] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Fact\_Transactions] WITH CHECK ADD FOREIGN KEY([CityKey])

REFERENCES [dbo].[Dim\_Cities] ([CityKey])

GO

ALTER TABLE [dbo].[Fact\_Transactions] WITH CHECK ADD FOREIGN KEY([CustomerKey])

REFERENCES [dbo].[Dim\_Customers] ([CustomerKey])

GO

ALTER TABLE [dbo].[Fact\_Transactions] WITH CHECK ADD FOREIGN KEY([DatesID])

REFERENCES [dbo].[Dates] ([DatesID])

GO

#### **A screenshot of the overall ETL workflow + for each ETL task:**

**DATES ETL**

Graphical user interface, application

Description automatically generated

**CITIES ETL**

Graphical user interface, application

Description automatically generated

**CUSTOMERS ETL**

A screenshot of a computer

Description automatically generated with medium confidence

**TRANSACTIONS ETL**

Graphical user interface, application

Description automatically generated

SSIS includes a **Slowly Changing Dimension** transform, which is designed to make loading dimension data easier when data changes in the source.

Let’s execute the following SQL script in SQL Server Management Studio (we have a source database with one table, and a data warehouse destination with one dimension table):

**Incremental load Script**

USE [sourceDB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[CustomerSource] Script Date: 03/12/2021 13:08:51 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[CustomerSource](

[CustomerKey] [int] IDENTITY(1,1) NOT NULL,

[CustomerID] [nvarchar](40) NULL,

[PhoneNumber] [nvarchar](40) NULL,

[DeliveryAddressLine1] [nvarchar](40) NULL,

[DeliveryAddressLine2] [nvarchar](40) NULL,

[DeliveryPostalCode] [nvarchar](20) NULL,

[PostalCityID] [nvarchar](20) NULL,

[CustomerName] [nvarchar](255) NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Source\_Transactions] Script Date: 03/12/2021 13:08:51 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Source\_Transactions](

[CustomerKey] [int] NULL,

[DatesID] [int] NULL,

[CityKey] [int] NULL,

[CustomerTransactionID] [int] NULL,

[TransactionDate] [nvarchar](40) NULL,

[SalesPersonName] [nvarchar](40) NULL,

[StockItemID] [int] NULL,

[StockItemName] [nvarchar](255) NULL,

[Quantity] [int] NULL,

[UnitPrice] [int] NULL,

[AmountExcludingTax] [int] NULL,

[TaxAmount] [int] NULL,

[TransactionAmount] [int] NULL

) ON [PRIMARY]

GO

**Data warehouse and dimension Script**

USE [sourceDB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[CustomerSource] Script Date: 03/12/2021 13:08:51 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[CustomerDest](

[CustomerKey] [int] IDENTITY(1,1) NOT NULL,

[CustomerID] [nvarchar](40) NULL,

[PhoneNumber] [nvarchar](40) NULL,

[DeliveryAddressLine1] [nvarchar](40) NULL,

[DeliveryAddressLine2] [nvarchar](40) NULL,

[DeliveryPostalCode] [nvarchar](20) NULL,

[PostalCityID] [nvarchar](20) NULL,

[CustomerName] [nvarchar](255) NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Source\_Transactions] Script Date: 03/12/2021 13:08:51 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TransactionsDest](

[CustomerKey] [int] NULL,

[DatesID] [int] NULL,

[CityKey] [int] NULL,

[CustomerTransactionID] [int] NULL,

[TransactionDate] [nvarchar](40) NULL,

[SalesPersonName] [nvarchar](40) NULL,

[StockItemID] [int] NULL,

[StockItemName] [nvarchar](255) NULL,

[Quantity] [int] NULL,

[UnitPrice] [int] NULL,

[AmountExcludingTax] [int] NULL,

[TaxAmount] [int] NULL,

[TransactionAmount] [int] NULL

) ON [PRIMARY]

GO

**ETL PROCESS FOR SLOWLY CHANGING DIMENSIONS:**

In order for KinetEco to regularly update the new data without affecting the source system all the time. We just write the query in SQL server management studio. And tested with the below query to check the record getting updated in the source destination table.

Use sourceDB

Go

INSERT [dbo].[CustomerSource] VALUES (101,100, '0144532156', 'BROOK STREET', 'CASTLE PLACE', 'AB1356A', 'MAG', CAST(N'2000-09-19' AS Date), CAST(N'2000-09-19' AS Date));

GO

Diagram

Description automatically generated Graphical user interface, text, application

Description automatically generated

**Fact table Incremental load:**

Diagram

Description automatically generated

## **TASK 2.2: DATA ANALYSIS:**

### **DELIVERABLES FOR TASK 2.2:**

Two screenshots: one for the dimension structure and hierarchy, and one for the browser view (See Lab 6 for examples). • A screenshot of the cube, in browser mode, showing the total sales amount for each of the months of 2018.

#### Browser mode:

I have managed to get the individual members but hierarchy isn’t showing up with filters. Each and every member is showing up in hierarchy search bar as given below in browser mode.

A picture containing text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

#### **c.,**

I can conclude the warehouse has less primary keys due to which the unique identification of each attribute is difficult in distributing them while hierarchies.

(ahriz, n.d.)

# Bibliography

# Bibliography

# Bibliography

ahriz, L. n. -. H., n.d. s.l.:s.n.