



Practical E01 The ETL Process

Objectives of this practical

- Learn how to extract data from different types of data sources and load it into the data warehouse
- Learn how to generate Time dimension

Section 1: About this Practical

The main objective of this practical is to let you experience the ETL process and learn how data from an OLTP database can be extracted, transformed and loaded into a Data Warehouse.

This practical requires you to create two databases, one is an OLTP database while the other is for a data warehouse.

- 1) First, you will create the OLTP database. To save time, you will be provided with the SQL scripts to create the tables in this database. You will also be provided with the scripts to **create** most of the data inside this database, except for **two** of them.
- 2) Next, you will create the database for the Data Warehouse. To save time, you will also be provided with the SQL scripts to create the tables. You will however NOT be provided with any script to create the data. This is because, in this lab, you will learn how to **load the data from the OLTP database** to the Data Warehouse data, and thus, you have to do it the “hard way”.
- 3) You will learn how to use the **BULK INSERT** command to copy data from a **CSV file into SQL Server**.
- 4) One of the tasks you will need to do in this practical is to learn how to write a SQL script to **load JSON data** to the data warehouse.

Section 2: Northwind Traders

Northwind Traders is an international gourmet food distributor that imports and exports specialty foods from around the world.

Northwind's products include meats, seafood, dairy products, beverages, and produce. Keeping track of customers, vendors, orders, and inventory is no small task. For many years, the owners of Northwind have been depending on an order-processing database in Microsoft Access to help manage their customers, suppliers, products, and orders. As the company grew, they started to use diverse systems such as Microsoft SQL Server, MySQL databases as well. Due to convenience and lack of resources, some of their company data is even stored in primitive formats such as text files.

With their company data sitting on so many diverse systems, the management of Northwind Traders is finding it increasingly difficult to get an accurate picture of their global sales performance.

They have heard of the benefits of maintaining a Data Warehouse and would like to embark on building a Data Warehouse for their company's data.

As a start, you will be helping them to create a one-subject Data Warehouse to track their SALES. Most of the orders data resides in an OLTP database in SQL Server called Northwind, so you figure it would be easy to use Transact-SQL or the Import-Export Wizard in SQL Server to transfer the transactional data to the new data warehouse.

However, there are two data sources which are more troublesome:

- The Shippers data
- The Employees data which can only be retrieved via a webservice which returns the data in JSON format

As an ETL Specialist, you will assist Northwind Traders to create the new Data Warehouse as well as to perform the necessary ETL process to copy the transactional data to the data warehouse.

Section 3: Create Northwind OLTP Database

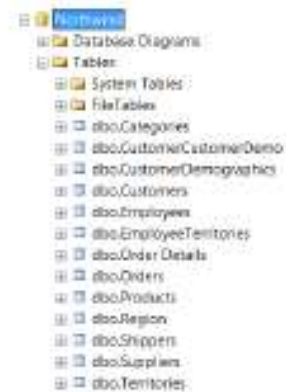
In this section, you will set up the Northwind OLTP Database in SQL Server. You are given the ERD diagram of this database, and your job is to create the physical database by following the ERD provided. If you have created the database correctly, you will then be able to create the tables and data very quickly using the scripts provided with this lab.

For the following task, use the SQL scripts provided in the 'NortWind_OLTP_Scripts'.

No.	Task
a)	<p>Before you begin, ensure that you have downloaded these files from Blackboard. These are the scripts which you will execute to create the tables in the OLTP database, and to populate the tables with the necessary data.</p> <ul style="list-style-type: none">• Northwind_OLTP_CreateTables.sql• 1.Categories.Table.sql• 1.CustomerDemographics.Table.sql• 1.Customers.Table.sql• 1.Region.Table.sql• 1.Suppliers.Table.sql• 2.CustomerCustomerDemo.Table.sql• 2.Products.Table.sql• 2.Territories.Table.sql• 3.EmployeeTerritories.Table.sql• 3.Orders.Table.sql• 4.Order Details.Table.sql
b)	<p>Note that I did not include "Shippers.Table.sql" or "Employees.Table.sql"</p> <p>The reason is because you will create the data for these two tables using a different method, which are outlined in Section 5 and 6 respectively.</p>
c)	<p>Study the structure of the Northwind database on the nextpage</p>
d)	<p>Create a new database in MS SQL Server called "Northwind"</p>

- e) Execute the SQL scripts from "Northwind_OLTP_CreateTables.sql" to create all tables in this database

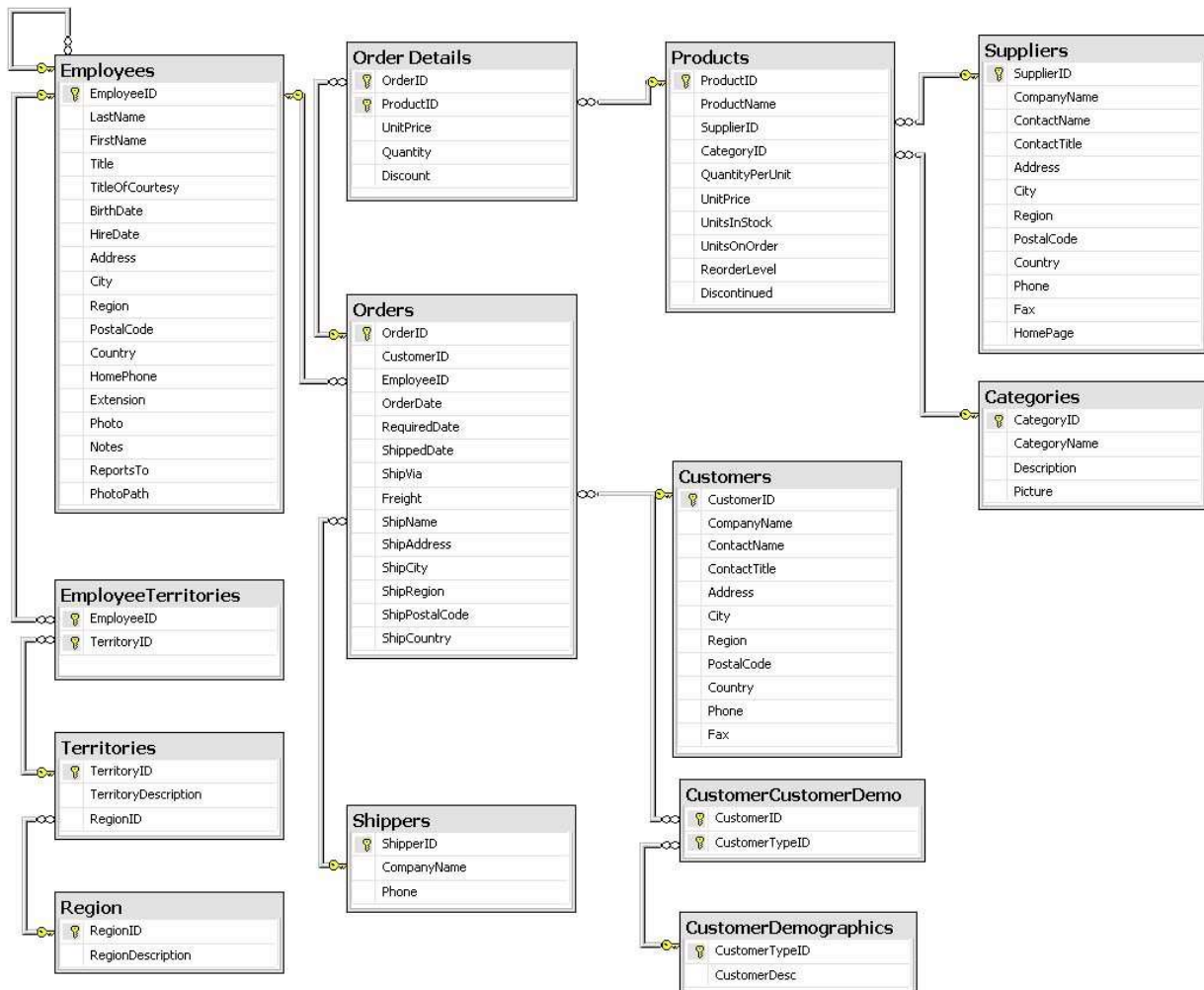
The following tables will be created upon successful execution of the script.



- f) Execute the SQL scripts from Part (a) to create **most of the data** in the Northwind OLTP database.

There would be **THREE** scripts that you would NOT execute at this point of time. Which are the three scripts? Why is it not appropriate to execute them now?

- g) Run "SELECT * FROM table_name" statements to test that the tables are populated with the required data.



Section 4: Create Northwind Data Warehouse

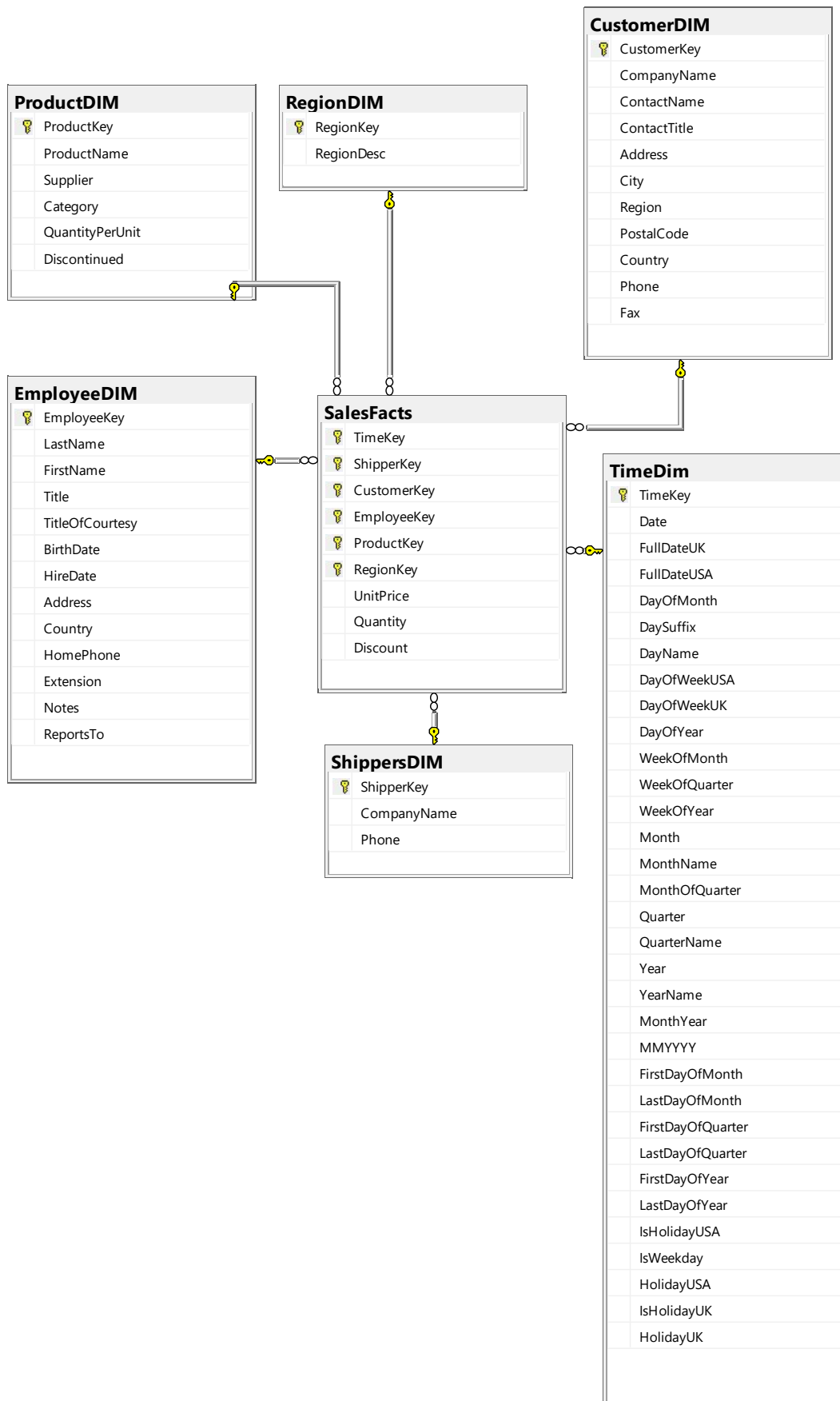
In this section, you will create the second database – the Northwind Data Warehouse in SQL Server. You are given the Star Schema of this database, and your job is to create the physical database using the schema provided. If you have created the database correctly, you will then be able to create the tables very quickly using the script provided with this lab.

However, unlike the previous database, most of the data would to be stored in the data warehouse would NOT be provided to you.

You will have to use various ETL methods outline.

For the following task, use the SQL scripts provided in the 'NorthWind_DW_Scripts'.

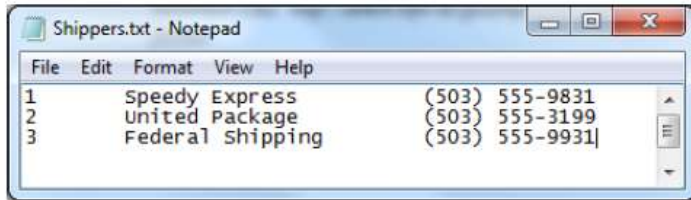
No.	Task
a)	Before you begin, ensure that you have downloaded this file from Blackboard: <ul style="list-style-type: none">• Northwind_DW_CreateTables.sql
b)	Study the Star schema of the Northwind Data Warehouse on the next page
c)	Create a new database in MS SQL Server called "NorthwindDW"
d)	Execute the SQL scripts from "Northwind_DW_CreateTables.sql" to create all the tables in the data warehouse



Section 5: ETL data from Shippers.txt to Northwind DW

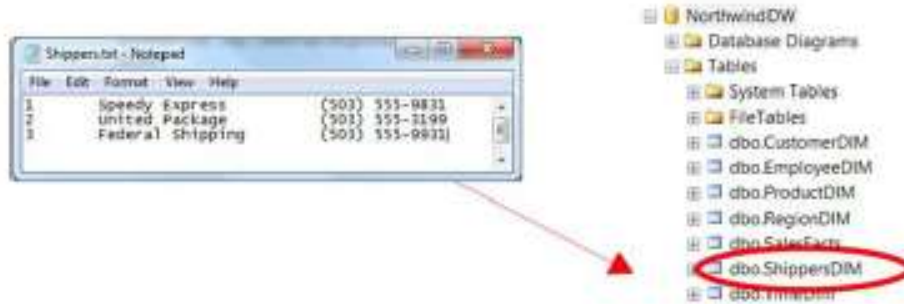
If you have completed Section 2 correctly, you would have noted that almost all of the tables in the Northwind OLTP database have been populated with data, except the **Shippers** table and the **Employees** table. Unlike the rest of the tables, the data for Shippers and Employees will be loaded directly to the Data Warehouse.

In this section, you will load the Shippers data. The Shippers data is stored in a flat-file (CSV). Let us focus on how we can ETL data that is stored in a flat-file (CSV) using the BULK INSERT method to the data warehouse.

No	Task
a)	Before you begin, ensure that you have downloaded these files from Blackboard: <ul style="list-style-type: none">• Shippers.txt• Northwind_DW_BULKINSERT.sql
b)	<ul style="list-style-type: none">• Copy Shippers.txt to the root of your C:\ such that you can access it at "C:\Shippers.txt"• Open up "C:\Shippers.txt"
c)	The contents of the Shippers.txt should look like this. Notice that each column is separated by a tab character ("t") 

No Task

- d) Our objective in this section is to load the content in Shippers.txt into the shippersDim table in the NorthwindDW database.



- e) To load the data from flat-files such as Shippers.txt into your MS SQL Server dataware house table, you can use the BULK INSERT command.

To save you time, the command has been created for you.

Open up Northwind_DW_BULKINSERT.sql

- f) You should see a script similar to that on the right.

```
1 USE NorthwindDW
2
3 BULK INSERT shippersdim
4 FROM 'C:\Shippers.txt'
5 WITH (fieldterminator='\t', rowterminator='\n')
6
```

Note that fieldterminator parameter of the BULK INSERT command is to indicate the separator character of each data field.

In our text file, the separator character is a tab and hence '\t'. If you are working with CSV file, the field separator is comma. Therefore, the parameter value should be ','.

- g) If you have executed this step correctly, congratulations!

You should see 3 rows of data in ShippersDim, which is the data loaded from Shippers.txt

ShipperKey	CompanyName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931

Section 6: ETL data from JSON to Northwind DW

If you have completed Section 2 correctly, you would have noted that almost all of the tables in the Northwind OLTP database have been populated with data, except the **Shippers** table and the **Employees** table.

In Section 5, you learnt how you can use the BULK INSERT command in SQL Server to load tab-delimited files into your SQL Server table. What if the data source is in another format, in this case, JSON?

Unfortunately, BULK INSERT does not recognise JSON format. Hence, we have to figure out other ways to load the JSON data into SQL Server instead.

In this section, let's focus on how we can ETL data that is stored in a JSON format.

Make use of the following script to load the JSON file into SQL Server. If you have executed the step correctly, you should see **9 rows of data in EmployeeDIM table**.

```
Use NorthWindDW

Declare @Employee varchar(max)

Select @Employee =
    BulkColumn
    from OPENROWSET(BULK 'employeesdata.json', SINGLE_BLOB) JSON

Insert into EmployeeDIM
    Select * From OpenJSON(@Employee, '$')
    with (
        EmployeeKey int '$.EmployeeKey',
        LastName varchar(20) '$.LastName',
        FirstName varchar(10) '$.FirstName',
        Title varchar(30) '$.Title',
        TitleOfCourtesy varchar(25) '$.TitleOfCourtesy',
        BirthDate varchar(50) '$.BirthDate',
        HireDate varchar(50) '$.HireDate',
        [Address] varchar(60) '$.Address',
        Country varchar(15) '$.Country',
        HomePhone varchar(24) '$.HomePhone',
        Extension varchar(4) '$.Extension',
        Notes varchar(255) '$.Notes',
        ReportsTo int '$.ReportsTo')
```

Section 7: Create data in the Time dimension table

The Time Dimension is a very important table present in all data warehouses to allow information to be tracked over time. The records in this table should include all the date values from the first day to the last day of the records in the Fact Table.

To save time, the Time dimension tables in most data warehouses are created via calculated formulas in automated scripts. In this section, you will execute a SQL script that will create the required date values for the Northwind DW for you.

No.	Task
a)	<p>Before you begin, ensure that you have downloaded these files from Blackboard:</p> <ul style="list-style-type: none"> Northwind_DW_CreateDataForTimeDimension.sql <p>Open up this file in SQL Server and run it. Notice that we are creating time values from 01 January 1995 to 01 January 2000.</p> <pre> 1 Use NorthwindDW 2 3 /***** 4 --Specify Start Date and End date here 5 --Value of Start Date Must be Less than Your End Date 6 7 DECLARE @StartDate DATETIME = '19950101' --Starting value of Date Range 8 DECLARE @EndDate DATETIME = '20000101' --End Value of Date Range </pre>
b)	<p>Now, check the TimeDIM in NorthwindDW database. The time values data should already be inside.</p> <p>If you have done this step correctly, you should have 1826 rows of Time Dimension data.</p> <p>Do you know why it is 1826 rows?</p> <p>The answer is simple ☞ you are creating a row of date data for each day in 1995, 1996, 1997, 1998 and 1999. Since there are approximately 365 days in each year, (365 days*5 years) = 1825 values. Since 1998 is a leap year, there is an additional value for that year, hence there are a total of 1826 values!</p>

Section 8: Load data into all other dimension tables

In the previous sections, you loaded data into three of the dimensional tables, namely the ShippersDim, the EmployeesDim and the TimeDim.

In this section, you will use “INSERT INTO” statement of Transact-SQL to load data from the Northwind OLTP database to the CustomerDim, ProductDIM and RegionDIM tables in the NorthwindDW database.

No. Task

- a) Before you begin, ensure that you have downloaded these files from Blackboard:
- Northwind_DW_ETLDataToDimensionTables.sql

Open up this script in SQL Server

- b) You should see a script similar to that below.

```

INSERT INTO
    NorthwindDW..ShippersDIM(ShipperKey, CompanyName, Phone)
SELECT
    ShipperID, CompanyName, Phone
FROM
    Northwind..Shippers

INSERT INTO
    NorthwindDW..ProductDIM(ProductKey, ProductName, Supplier, Category, QuantityPerUnit, Discontinued)
SELECT
    p.ProductID, p.ProductName, s.CompanyName, c.CategoryName, p.QuantityPerUnit, p.Discontinued
FROM
    Northwind..Products p, Northwind..Suppliers s, Northwind..Categories c
WHERE
    p.SupplierID=s.SupplierID AND p.CategoryID=c.CategoryID

INSERT INTO NorthwindDW..CustomerDIM SELECT * FROM Northwind..Customers n

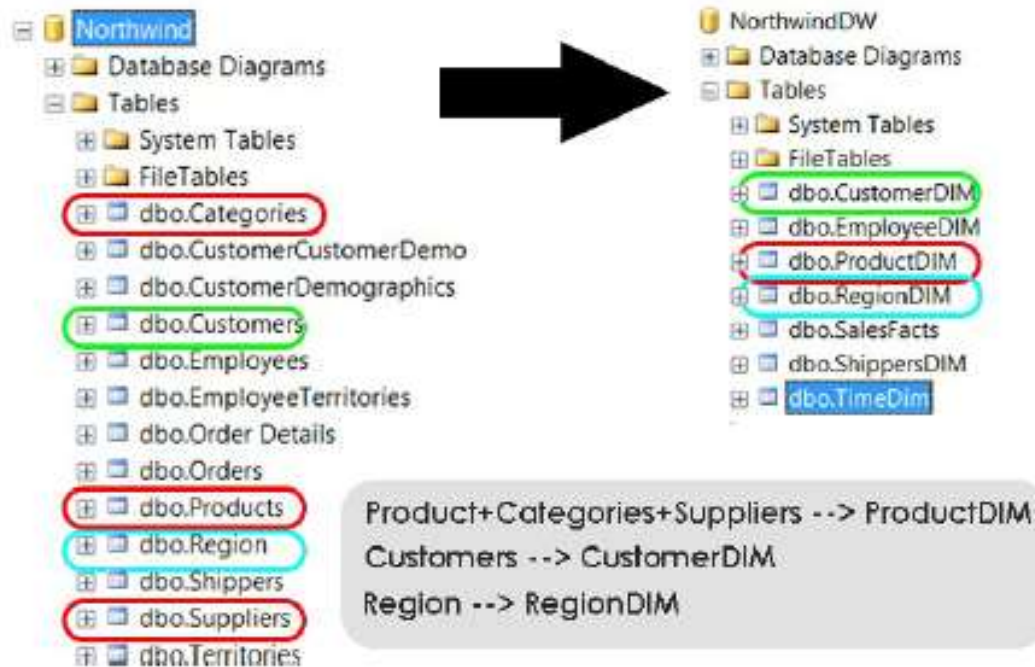
INSERT INTO
    NorthwindDW..RegionDIM(RegionKey, RegionDesc)
SELECT
    RegionID, RTRIM(RegionDescription) FROM Northwind..Region
  
```


c) This is what SQL Server will do if you run this script:

- Extract records from Products, Suppliers and Categories in Northwind OLTP database and load them into ProductsDim table in NorthwindDW
- Extract records from Customers in Northwind OLTP and load them into CustomerDIM in Northwind DW
- Extract the data from Region in Northwind OLTP and load it into RegionDim

Northwind OLTP Database

Northwind Data Warehouse



d) Run the script

e) Check the ProductDIM, CustomerDim and RegionDim tables in NorthwindDW database.

The records from the OLTP database should already have been loaded inside.

If you have done everything correctly for this step, you should have:

- 91 rows in CustomerDIM
- 77 rows in ProductDim
- 4 rows in RegionDim

```
INSERT INTO NorthwindDW..CustomerDIM SELECT * FROM Northwind..Customers

INSERT INTO
  NorthwindDW..ProductDIM(ProductKey, ProductName, Supplier, Category, QuantityPerUnit,
  Discontinued)
SELECT
  p.ProductID, p.ProductName, s.CompanyName, c.CategoryName, p.QuantityPerUnit,
  p.Discontinued
FROM
  Northwind..Products p, Northwind..Suppliers s, Northwind..Categories c
WHERE
  p.SupplierID=s.SupplierID AND p.CategoryID=c.CategoryID

INSERT INTO
  NorthwindDW..RegionDIM(RegionKey, RegionDesc)
SELECT
  RegionID, RTRIM(RegionDescription)
FROM
```

100 % -

Messages

(91 row(s) affected)

(26 row(s) affected)

(8 row(s) affected)

Section 9: Load data into the Fact table

You have come to the final step of this practical before your Data Warehouse is ready! In this step, you will need to load data from the OLTP database into the Salesfacts table.

This section involves loading data from several tables in the OLTP database, such as the Orders table, the OrderDetails table, Region table, EmployeeTerritories table and Territories table.

No	Task
a)	<p>Before you begin, ensure that you have downloaded these files from Blackboard:</p> <ul style="list-style-type: none"> • Northwind_DW_ETLDataToFactTable.sql <p>Open up this script in SQL Server. You should see a script similar to the following:</p> <pre> 1 use NorthwindDW 2 3 DELETE FROM SalesFacts 4 5 INSERT INTO NorthwindDW..SalesFacts 6 (7 TimeKey, ShipperKey, CustomerKey, EmployeeKey, 8 ProductKey, RegionKey, UnitPrice, Quantity, Discount) 9 SELECT 10 replace(CONVERT(DATE, o.OrderDate, 112), '-', ''), 11 s.ShipperKey, 12 c.CustomerKey, 13 e.EmployeeKey, 14 p.ProductKey, 15 r.RegionID, 16 od.UnitPrice, 17 od.Quantity, 18 od.Discount 19 FROM 20 Northwind..[Order Details] od </pre>
b)	Run the script.
c)	<p>Check the SalesFacts table in NorthWindDW database. The records from the OLTP database should have been loaded.</p> <p>If no records were loaded to the SalesFacts table, check the OrderDetails table in the OLTP (NorthWind) database. Recall that in Section 2 (f), there would be Three scripts that you would not execute at that point of time. Refer to Appendix A for tips how to execute the three scripts.</p> <p>If you have performed all steps correctly, you should see that exactly 2155 rows of data, the same number of rows of data as that in the OLTP OrderDetails table have been ETL'ed to the Northwind Data Warehouse.</p>

Section 10: Test your Data Warehouse

Hurray, your Data Warehouse is finally ready for testing.

To test your data warehouse, try to come up with two SQL queries to test the data warehouse you just created. Here are 2 examples you can try out:

- "What is the bestselling product on Tuesday?"
- "Who is the best salesman in 1996?"

No	Task
a)	<p>What is the bestselling product on Tuesday?</p> <pre> Use NorthWindDW select top(1) productName, sum(quantity) as TotalQuantity from SalesFacts SF inner join TimeDim TD on SF.TimeKey = TD.TimeKey inner join productDIM PD on SF.ProductKey = PD.ProductKey where TD.DayName = 'Tuesday' group by productName order by TotalQuantity desc </pre>
b)	<p>Who is the salesman with highest Sales in 1996?</p> <pre> Use NorthWindDW select concat(ED.firstName, ' ', ED.lastName) as Salesman, round(sum(quantity * UnitPrice * (1 - discount)), 0) as Total_Sales from SalesFacts SF inner join TimeDim TD on SF.TimeKey = TD.TimeKey inner join EmployeeDim ED on SF.employeeKey = ED.employeeKey where TD.Year = '1996' group by ED.employeeKey, concat(ED.firstName, ' ', ED.lastName) order by Total_sales desc </pre>
c)	"Which was the customer who gave the most revenue in Quarter 1 of 1998?"
d)	"Which was the region with the least sales in the month of December in 1997?"

Appendix A: Removing Constraints temporarily


If you recall, in Section 3 of this document, we populated **8 out of 13** tables in the OTLP database and left 5 tables unpopulated with data, because of foreign key constraints.

The **three** tables, namely the EmployeeTerritories table, the Orders and OrderDetails table could not be populated because they depended on the records in the **Shippers** and **Employees** table to be populated. However, it is so important that the Orders and OrderDetails tables are populated because we will need them to populate the SalesFacts Fact table in Section 9!

To get around this constraint, we are going to "play cheat" by altering the constraints of the Orders table and the EmployeeTerritories tables, so that these tables can be populated with data first, even though there are no matching records in the Employees and Shippers table.

Take note that removing constraints like this, is NOT a good practice, however, it is a skillset that SQL developers have to know for those 'in-case of desperate situations' scenarios, so that's why I am teaching this to you.

Part 1: Cheating the Orders table

No.	Task
a)	<p>Open "3.Orders.Table.sql" in the Northwind_OLTP_Scripts folder and run the code</p> <p>After executing the script, you should see error messages such as these below</p> <div><pre>Msg 547, Level 16, State 0, Line 6 The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Orders_Employees". The statement has been terminated. Msg 547, Level 16, State 0, Line 7 The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Orders_Employees". The statement has been terminated. Msg 547, Level 16, State 0, Line 8 The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Orders_Employees". The statement has been terminated.</pre></div>
b)	<p>Ask yourself this question -- why do these errors occur?</p> <p>The answer is this:</p>

- The Orders table contains 3 foreign keys: CustomerID, EmployeeID and ShippersID, which are the primary keys to the Customer table, Employee table and the Shippers table.
- Whenever a table A contains a foreign key that is the primary key of another table B, table A cannot be populated until table B is populated because it has a foreign key constraint.
- So in the case of the Orders table, we encountered an error because the script tried to insert a record that references a certain Employee in the Employees table, but it is not correct to do so, because the Employees table is empty and does not contain the corresponding record

Is there a way to get around this issue then? i.e. insert an Orders record that references an Employee into the Order table, even though the Employees table is empty?

Yes, there is, though it is not a practice we would encourage SQL developers to exercise unless in dire situations!

Check out the next step to find out how to get around this!

- c) As mentioned in the previous step, we encountered the errors because of the foreign key constraints.

To get around this constraint, we can “play cheat” and remove it temporarily.

To remove the constraint temporarily, insert the line highlighted in the red border in the code as shown

ALTER TABLE Orders NOCHECK CONSTRAINT ALL

```
1 USE [NORTHWIND]
2 GO
3 SET IDENTITY_INSERT [dbo].[Orders] ON
4
5 ALTER TABLE Orders NOCHECK CONSTRAINT ALL
6
7 INSERT [dbo].[Orders] ([OrderID], [CustomerID], [EmployeeID],
```

- d) Run the code with the newly inserted line.

You should now be able to insert the records into the Orders table successfully as shown below. If all went well, there would be 830 records inserted into the Orders table.

OrderID	Customer	Employee	OrderDate	RequiredDate	ShippedDate	Ship	Freight	ShipName	ShipAddress
10248	SWART	0	1996-07-04 00:00:00	1996-08-01 00:00:00	1996-07-18 00:00:00	3	32.28	Wine at Abbots Cheshire	30 rue de l'Abbaye
10249	COMPE	0	1996-07-05 00:00:00	1996-08-10 00:00:00	1996-07-10 00:00:00	1	11.61	Terra Superalimentari	Luisenweg 45
10250	HAMAO	4	1996-07-06 00:00:00	1996-08-05 00:00:00	1996-07-12 00:00:00	0	65.83	Harumi Ganari	Rua do Paço, 57
10251	WATFI	3	1996-07-08 00:00:00	1996-08-05 00:00:00	1996-07-18 00:00:00	1	41.34	WineWines en direct	2, rue du Commerce
10252	SUPPE	4	1996-07-08 00:00:00	1996-08-05 00:00:00	1996-07-11 00:00:00	2	81.30	Suppelen offices	Emileweg 100A, 2105
10253	HAMAR	3	1996-07-12 00:00:00	1996-07-24 00:00:00	1996-07-18 00:00:00	2	30.17	Harumi Ganari	Rua do Paço, 57
10254	CHOPS	8	1996-07-15 00:00:00	1996-08-05 00:00:00	1996-07-23 00:00:00	2	32.88	Chop away Chinese	Haugen 31
10255	PICRU	0	1996-07-12 00:00:00	1996-08-05 00:00:00	1996-07-18 00:00:00	3	148.35	Richier Supermarché	Steinweg 5
10256	WELLS	3	1996-07-19 00:00:00	1996-08-12 00:00:00	1996-07-17 00:00:00	2	15.07	Wallingford Importadora	Rua de Mercão, 11
10257	HLAA	4	1996-07-18 00:00:00	1996-08-15 00:00:00	1996-07-22 00:00:00	3	81.80	HLANON Alimentos	Carrera 22 con Ave. Centro Seleccion 40C
10258	SLHSEA	1	1996-07-17 00:00:00	1996-08-14 00:00:00	1996-07-25 00:00:00	1	148.51	Sand Handel	Reichsgraben 6
10259	OSANTO	4	1996-07-18 00:00:00	1996-08-15 00:00:00	1996-07-25 00:00:00	0	8.26	Cartho commercial Marketing	Strada 10, 1400000000
10260	CHITIK	4	1996-07-18 00:00:00	1996-08-16 00:00:00	1996-07-26 00:00:00	1	30.19	CRISOL Kiosketery	Mohlenstr. 30B
10261	SLHSEA	4	1996-07-18 00:00:00	1996-08-16 00:00:00	1996-07-26 00:00:00	2	3.00	Chai Delicat	Rua da Figueiredo, 12
10262	SLHSEA	4	1996-07-18 00:00:00	1996-08-16 00:00:00	1996-07-26 00:00:00	2	401.18	Sancho e Silva - Produtos Alimentares	5017 Avenida Pa

Part 2: Populate the OrderDetails table

- | No | Task |
|----|---|
| a) | Now that you have "cheated" the SQL serve to let you populate the Order table, even though we have no records in the Employee table or Shipper table, you can now proceed to run the "4.Order Details.Table.sql" as well. |
| b) | If all went well, there would be 2155 records inserted into the Order Details table. |

Part 3: Cheating the EmployeesTerritories table

- | No | Task |
|----|--|
| a) | Open "3.EmployeeTerritories.Table.sql" in the Northwind_OLTP_Scripts folder and run the code. |
| b) | After executing the script, you should see error messages such as these below <div data-bbox="297 1165 1395 1461" data-label="Code-Block"> <pre> Messages Msg 547, Level 16, State 0, Line 3 The INSERT statement conflicted with the FOREIGN KEY constraint "FK_EmployeeTerritories_Empl The statement has been terminated. Msg 547, Level 16, State 0, Line 4 The INSERT statement conflicted with the FOREIGN KEY constraint "FK_EmployeeTerritories_Empl The statement has been terminated. Msg 547, Level 16, State 0, Line 5 The INSERT statement conflicted with the FOREIGN KEY constraint "FK_EmployeeTerritories_Empl The statement has been terminated. </pre> </div> |
| c) | Follow the steps outline in Part 1 to get around this problem. If OK, there should be 49 records loaded. 😊 |

Reference:

- SQL Server 2016: Bulk Import JSON file data to Table,
<https://social.technet.microsoft.com/wiki/contents/articles/37292.sql-server-2016-bulk-import-json-file-data-to-table.aspx>, Accessed on 22/03/2020