

```
-- *****
-- ***** NEW PRODUCTS *****
-- *****
```

```
-- Search and insert newly added product into staging added product table
```

```
INSERT INTO StagingDatabase.staging.stage_dim_product_added (product_id, name, valid_from, valid_to)
SELECT ProductID, Name, SellStartDate, SellEndDate
FROM AdventureWorks2017.Production.Product
WHERE productID IN (SELECT productID
                    FROM AdventureWorks2017.Production.Product
                    EXCEPT
                    SELECT product_id
                    FROM AdventureWorks_DW.star_schema.d_product);
```

```
-- Replace all NULL values with date 31.12.9999
```

```
UPDATE StagingDatabase.staging.stage_dim_product_added
SET valid_to='9999-12-31'
WHERE valid_to IS NULL;
```

```
-- Load newly added and modified rows into the Data Warehouse
```

```
INSERT INTO AdventureWorks_DW.star_schema.d_product
SELECT *
FROM StagingDatabase.staging.stage_dim_product_added;
```

```
-- *****
-- ***** DELETED PRODUCTS *****
-- *****
```

```
-- Retrieve and update data warehouse, set valid_to attribute to yesterdays date for deleted
-- products.
```

```
UPDATE AdventureWorks_DW.star_schema.d_product
SET valid_to = DATEADD(dd, -1, GETDATE())
WHERE product_id IN (
    SELECT product_id
    FROM AdventureWorks_DW.star_schema.d_product
    WHERE product_id IN (SELECT product_id
                        FROM AdventureWorks_DW.star_schema.d_product
                        EXCEPT
                        SELECT productID
                        FROM AdventureWorks2017.Production.Product)
)
```

```
-- *****
-- ***** UPDATED PRODUCTS *****
-- *****
```

```
-- Inserting updated rows into the temporary table to handle changes
```

```
INSERT INTO StagingDatabase.staging.stage_dim_product_changed
(product_id, name) (SELECT ProductID, Name
                   FROM AdventureWorks2017.Production.Product
                   EXCEPT
                   SELECT product_id, name
                   FROM StagingDatabase.staging.stage_dim_product
                   EXCEPT (
                       SELECT ProductID, Name
                       FROM AdventureWorks2017.Production.Product
                       WHERE productID IN
                           (SELECT productID
                            FROM AdventureWorks2017.Production.Product
                            EXCEPT
                            SELECT product_id
                            FROM StagingDatabase.staging.stage_dim_product)
                   ));
```

```
-- Update valid_to attribute to '9999-12-31'
```

```
UPDATE StagingDatabase.staging.stage_dim_product_changed
SET valid_to = '9999-12-31'
WHERE valid_to IS NULL;
```

```
-- Update valid_from to today's date
```

```
UPDATE StagingDatabase.staging.stage_dim_product_changed
SET valid_from = GETDATE()
WHERE valid_from IS NULL;
```

```
-- Alter changed rows in Data Warehouse
```

```
UPDATE AdventureWorks_DW.star_schema.d_product
SET valid_to = DATEADD(dd, -1, GETDATE())
WHERE product_id in (SELECT product_id FROM StagingDatabase.staging.stage_dim_product_changed);
```

```
-- Insert new product to Data Warehouse
```

```
INSERT INTO AdventureWorks_DW.star_schema.d_product
SELECT *
FROM StagingDatabase.staging.stage_dim_product_changed;
```

```
-- Delete data in temporary table
```

```
DELETE FROM StagingDatabase.staging.stage_dim_product_changed;
```

```
-- *****  
-- ***** FACT TABLE UPDATE *****  
-- *****
```

```
DECLARE @LAST_UPDATE as DATETIME = (SELECT lastUpdate  
      FROM StagingDatabase.staging.LastUpdate);
```

```
-- Insert newly updated rows into temp_f_sales table. Select only the ones newer than the last update.
```

```
INSERT INTO StagingDatabase.staging.stage_f_sales  
(business_customer_id, business_product_id, business_order_date, quantity, line_total)  
  (SELECT C.CustomerID, P.ProductID, OrderDate, SOD.OrderQty, SOD.LineTotal  
    FROM AdventureWorks2017.Sales.SalesOrderHeader SOH  
      JOIN AdventureWorks2017.Sales.SalesOrderDetail SOD on SOH.SalesOrderID = SOD.SalesOrderID  
      JOIN AdventureWorks2017.Sales.Customer C on SOH.CustomerID = C.CustomerID  
      JOIN AdventureWorks2017.Production.Product P on SOD.ProductID = P.ProductID  
    WHERE OnlineOrderFlag = 1  
      AND OrderDate > @LAST_UPDATE);
```

```
-- Find corresponding surrogate keys.
```

```
-- ***** Customer *****
```

```
UPDATE StagingDatabase.staging.stage_f_sales  
SET customer_id = (SELECT dimension_customer_id  
      FROM AdventureWorks_DW.star_schema.d_customer AS dim_C_id  
      WHERE dim_C_id.customer_id = business_customer_id  
        AND valid_to = '9999-12-31')  
WHERE customer_id IS NULL;
```

```
-- ***** Product *****
```

```
UPDATE StagingDatabase.staging.stage_f_sales  
SET product_id = (SELECT dimension_product_id  
      FROM AdventureWorks_DW.star_schema.d_product AS dim_P_id  
      WHERE dim_P_id.product_id = business_product_id  
        AND valid_to = '9999-12-31')  
WHERE product_id IS NULL;
```

```
-- ***** Date *****
```

```
UPDATE StagingDatabase.staging.stage_f_sales  
SET date_id = (SELECT dimension_date_id  
      FROM AdventureWorks_DW.star_schema.d_date AS dim_D_id  
      WHERE dim_D_id.date = business_order_date)  
WHERE date_id IS NULL;
```

```
-- Insert data into Data Warehouse Fact Sales table
```

```
INSERT INTO AdventureWorks_DW.star_schema.f_sales(customer_id, product_id, date_id, quantity, line_total)  
SELECT customer_id, product_id, date_id, quantity, line_total  
FROM StagingDatabase.staging.temp_f_sales;
```

```
-- Update last update table with the newest date
```

```
UPDATE StagingDatabase.staging.LastUpdate  
SET lastUpdate = GETDATE()  
WHERE lastUpdate = @LAST_UPDATE;
```

```

-- Search and insert newly added customer into staging added Customer table
INSERT INTO StagingDatabase.staging.stage_dim_customer_added(customer_id, title, first_name, middle_name, last_name)
SELECT CustomerID, Title, FirstName, MiddleName, LastName
FROM AdventureWorks2017.Sales.Customer JOIN AdventureWorks2017.Person.Person ON Customer.PersonID =
Person.BusinessEntityID
WHERE CustomerID IN (SELECT CustomerID
FROM AdventureWorks2017.Sales.Customer
EXCEPT
SELECT customer_id
FROM StagingDatabase.staging.stage_dim_customer);

```

```

-- Replace all NULL values with current date
UPDATE StagingDatabase.staging.stage_dim_customer_added
SET valid_from = GETDATE()
WHERE valid_to IS NULL;

```

```

-- Replace all NULL values with date 31.12.9999
UPDATE StagingDatabase.staging.stage_dim_customer_added
SET valid_to = '9999-12-31'
WHERE valid_to IS NULL;

```

```

-- Load newly added and modified rows into the Data Warehouse
INSERT INTO AdventureWorks_DW.star_schema.d_customer
SELECT *
FROM StagingDatabase.staging.stage_dim_customer_added;

```

```

-- Retrieve and update data warehouse, set valid_to attribute to yesterdays date for deleted
-- customers.

```

```

UPDATE AdventureWorks_DW.star_schema.d_customer
SET valid_to = DATEADD(dd, -1, GETDATE())
WHERE customer_id IN (
SELECT customer_id
FROM AdventureWorks_DW.star_schema.d_customer
WHERE customer_id IN (SELECT customer_id
FROM AdventureWorks_DW.star_schema.d_customer
EXCEPT
SELECT CustomerID
FROM AdventureWorks2017.Sales.Customer)
)

```

```

-- Inserting updated rows into the temporary table to handle changes

```

```

INSERT INTO StagingDatabase.staging.stage_dim_customer_changed
(customer_id, title, first_name, middle_name, last_name) (SELECT CustomerID, Title, FirstName, MiddleName, LastName
FROM AdventureWorks2017.Sales.Customer
JOIN AdventureWorks2017.Person.Person ON Customer.PersonID = Person.BusinessEntityID
EXCEPT
SELECT customer_id, title, first_name, middle_name, last_name
FROM StagingDatabase.staging.stage_dim_customer
EXCEPT (
SELECT CustomerID, Title, FirstName, MiddleName, LastName
FROM AdventureWorks2017.Sales.Customer
JOIN AdventureWorks2017.Person.Person ON Customer.PersonID = Person.BusinessEntityID
WHERE CustomerID IN
(SELECT CustomerID
FROM AdventureWorks2017.Sales.Customer
EXCEPT
SELECT customer_id
FROM StagingDatabase.staging.stage_dim_customer)
));

```

```

-- Update valid_to attribute to '9999-12-31'

```

```

UPDATE StagingDatabase.staging.stage_dim_customer_changed

```

```
SET valid_from = GETDATE()  
WHERE valid_from IS NULL;
```

```
-- Update title attribute to 'N/A'
```

```
UPDATE StagingDatabase.staging.stage_dim_customer_changed  
SET valid_to = '9999-12-31'  
WHERE valid_to IS NULL;
```

```
-- Update middle_name attribute to 'N/A'
```

```
UPDATE StagingDatabase.staging.stage_dim_customer_changed  
SET title = 'N/A'  
WHERE title IS NULL;
```

```
-- Update valid_to attribute to 'N/A'
```

```
UPDATE StagingDatabase.staging.stage_dim_customer_changed  
SET middle_name = 'N/A'  
WHERE middle_name IS NULL;
```

```
-- Alter changed rows in Data Warehouse
```

```
UPDATE AdventureWorks_DW.star_schema.d_customer  
SET valid_to = DATEADD(dd, -1, GETDATE())  
WHERE customer_id in (SELECT customer_id FROM StagingDatabase.staging.stage_dim_customer_changed);
```

```
-- Insert new customer to Data Warehouse
```

```
INSERT INTO AdventureWorks_DW.star_schema.d_customer  
SELECT *  
FROM StagingDatabase.staging.stage_dim_customer_changed;
```