

```
-- 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;
```