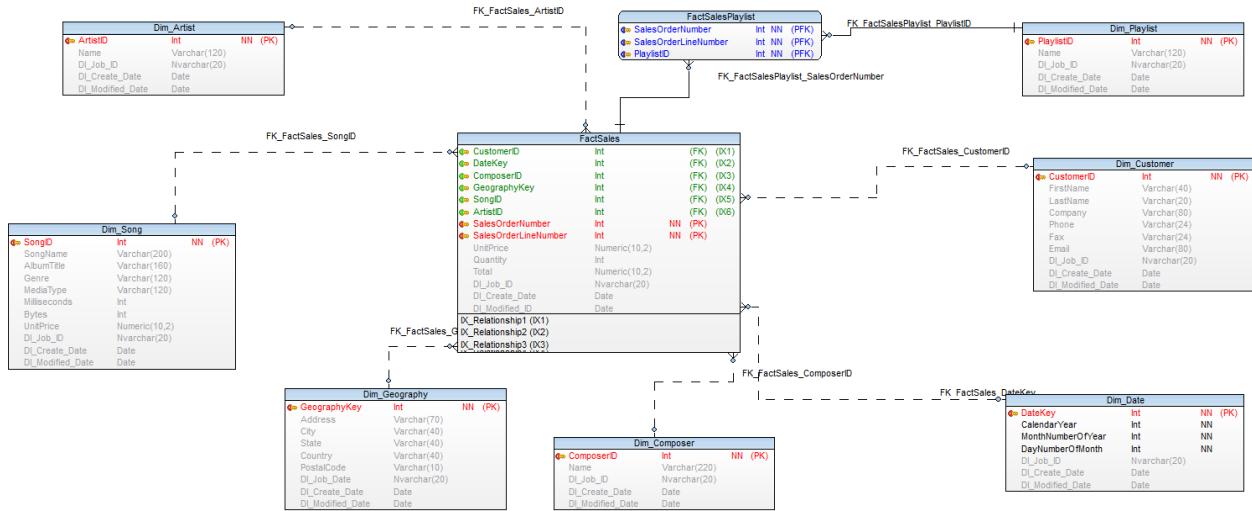


# Dimension Model of ChinookDW



Azure

Screenshot of Microsoft Azure SQL databases Data explorer (preview) showing a query result for Invoice data.

Query:

```

1 SELECT SUM(Total), BillingCountry, BillingState, BillingCity
2 FROM Invoice
3 GROUP BY BillingCountry, BillingState, BillingCity;
    
```

Results:

	BILLINGCOUNTRY	BILLINGSTATE	BILLINGCITY
37.62	Argentina		Buenos Aires
37.62	Australia	NSW	Sidney

SQL Server

SQLQuery1.sql - RISHI-YOGAPAD.Chinook (info7290 (53)) - Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Help

Chinook Execute Debug Generic Debugger

Registered Servers Database Engine

Object Explorer Connect RISHI-YOGAPAD (SQL Server 14.0.1000.169 - i)

- Databases
  - System Databases
  - Database Snapshots
  - AdventureWorks2014
  - AdventureWorksDW2014
  - Chinook
    - Database Diagrams
    - Tables
    - Views
    - External Resources
    - Synonyms
    - Programmability
    - Service Broker
    - Storage
    - Security
  - ChinookNew
  - Security

SQLQuery1.sql - RISHI-YOGAPAD (info7290 (53))

```
SELECT SUM(Total), BillingCountry, BillingState, BillingCity
FROM Invoice
GROUP BY BillingCountry, BillingState, BillingCity;
```

Results Messages

(No column name)	BillingCountry	BillingState	BillingCity
1	37.62	Argentina	NULL
2	37.62	Australia	NSW
3	42.62	Austria	NULL
4	37.62	Belgium	NULL
5	37.62	Brazil	DF
6	37.62	Brazil	RJ
7	39.62	Brazil	SP
8	75.24	Brazil	SP
9	37.62	Canada	AB
10	38.62	Canada	BC
			Vancouver

Query executed successfully. | RISHI-YOGAPAD (14.0 RTM) | info7290 (53) | Chinook | 00:00:00 | 53 rows

## PostGre

pgAdmin 4

File Object Tools Help

Browser Servers (1) PostgreSQL 10 Databases (2)

- Chinook
  - Casts
  - Catalogs
  - Event Triggers
  - Extensions
  - Foreign Data Wrappers
  - Languages
  - Schemas (1)
    - public
      - Collations
      - Domains
      - FTS Configurations
      - FTS Dictionaries
      - FTS Parsers
      - FTS Templates
      - Foreign Tables
      - Functions
      - Materialized Views
      - Sequences
      - Tables (11)
        - Album
        - Artist
        - Customer
        - Employee
        - Genre
        - Invoice
        - InvoiceLine
        - MediaType
        - Playlist
        - PlaylistTrack
        - Track
      - Trigger Functions
      - Types
      - Views
- postgres
  - Login/Group Roles
  - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Query - postgres ... Query - Chinook ... Select script

Chinook on postgres@PostgreSQL 10

```
1 SELECT SUM("total"), "BillingCountry", "BillingState", "BillingCity"
2 FROM public."Invoice"
3 GROUP BY "BillingCountry", "BillingState", "BillingCity";
```

Data Output Explain Messages Query History

sum	BillingCountry	BillingState	BillingCity
39.62	USA	WA	Redmond
38.62	Canada	BC	Vancouver
40.62	Netherlands	VV	Amsterdam
41.62	Finland	[null]	Helsinki
37.62	USA	MA	Boston
45.62	Ireland	Dublin	Dublin
37.62	USA	NY	New York
39.62	France	[null]	Bordeaux
42.62	USA	WI	Madison
37.62	Spain	[null]	Madrid
43.62	USA	UT	Salt Lake City
77.24	USA	CA	Mountain View
37.62	Canada	NT	Yellowknife
38.62	Sweden	[null]	Stockholm
37.62	Germany	[null]	Stuttgart
40.62	France	[null]	Dijon
36.64	India	[null]	Bangalore
37.62	USA	NV	Reno
38.62	India	[null]	Delhi
77.24	France	[null]	Paris
39.62	USA	FL	Orlando
75.24	Brazil	SP	Sao Paulo
39.62	Canada	QC	Montreal
47.62	USA	TX	Fort Worth

## MySql

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL57' is selected. The 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' menus are visible. Below the menu bar is a toolbar with various icons. The 'Navigator' pane on the left lists databases: dutta, mogambo, moviedb, sakila, sys, and world. The central area contains a 'SQL File 4' tab with the following SQL code:

```
1 • SELECT SUM(Total), BillingCountry, BillingState, BillingCity
2   FROM Invoice
3   GROUP BY BillingCountry, BillingState, BillingCity;
```

The results are displayed in a 'Result Grid' table:

SUM(Total)	BillingCountry	BillingState	BillingCity
37.62	Argentina	NULL	Buenos Aires
37.62	Australia	NSW	Sidney
42.62	Austria	NULL	Vienne
37.62	Belgium	NULL	Brussels
37.62	Brazil	DF	Brasilia
37.62	Brazil	RJ	Rio de Janeiro
39.62	Brazil	SP	São José dos Campos
75.24	Brazil	SP	São Paulo
37.62	Canada	AB	Edmonton

## Oracle

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes 'File', 'Edit', 'View', 'Navigate', 'Run', 'Source', 'Team', 'Tools', and 'Window'. The 'Help' menu is also present. On the left, the 'Connections' sidebar shows a connection to 'chnook'. The 'Connections' tree view includes 'Connections', 'Tables (Filtered)', 'Views', 'Editor', 'Indexes', 'Packages', 'Procedures', 'Functions', 'Queues', 'Queues Tables', 'Triggers', 'Crossedition Triggers', 'Types', 'Sequences', 'Materialized Views', 'Materialized View Logs', 'Synonyms', 'Public Synonyms', 'Database Links', 'Database Links', 'Directories', and 'Editors'. The 'Reports' sidebar includes 'All Reports', 'Analytic View Reports', 'Data Dictionary Reports', 'Data Modeler Reports', 'Dashboards', 'TimeTen Reports', and 'User Defined Reports'. The central workspace has a 'Worksheet - Query Builder' tab with the following SQL code:

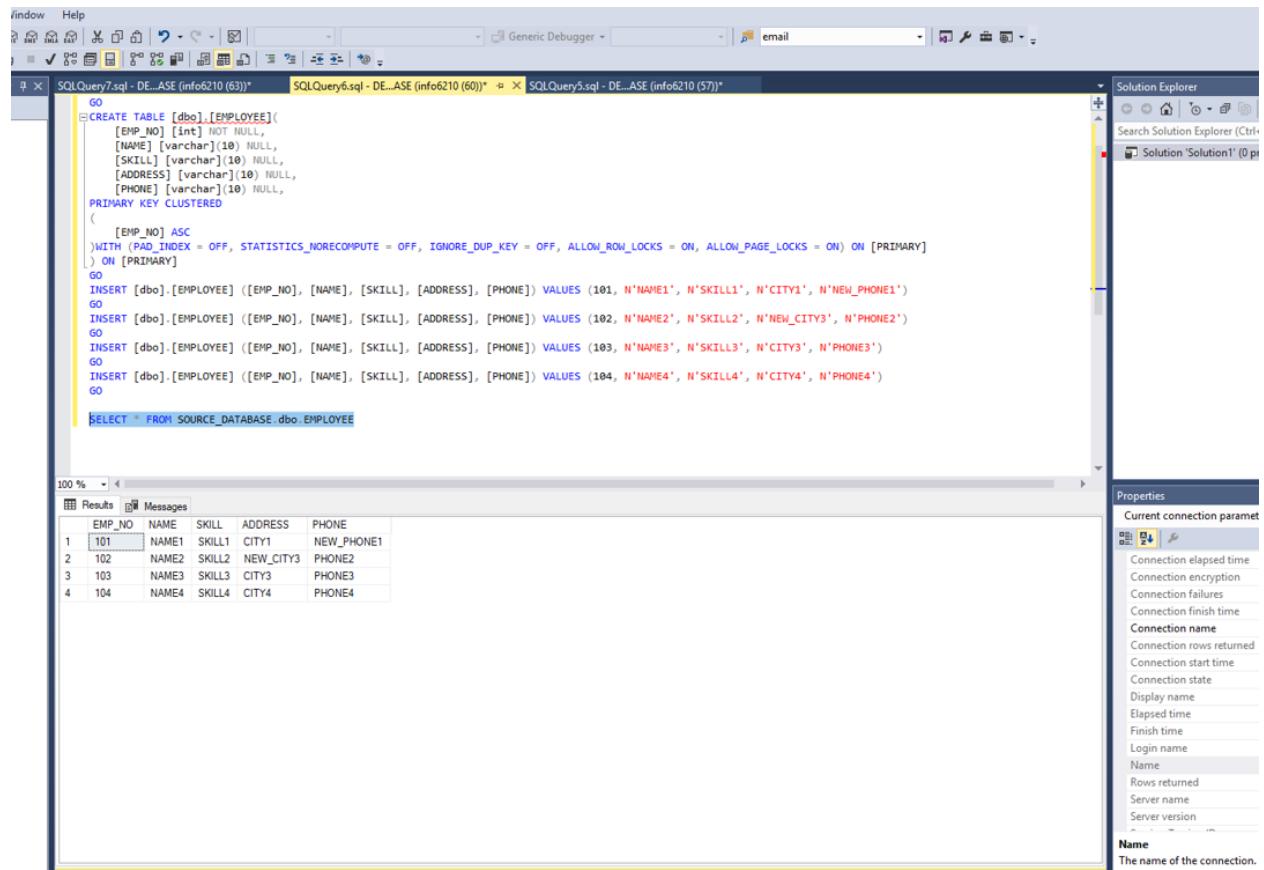
```
1 select BillingCountry as Country,BillingState as State,BillingCity as City,sum(total) as Sales from invoice
2 group by BillingCountry,BillingState,BillingCity;
```

The results are displayed in a 'Query Result' table:

COUNTRY	STATE	CITY	SALES
1 Chile	(null)	Santiago	46.62
2 USA	NV	Reno	37.62
3 France	(null)	Paris	77.24
4 Spain	(null)	Madrid	37.62
5 USA	WA	Redmond	39.62
6 Belgium	(null)	Brussels	37.62
7 USA	WI	Madison	42.62
8 Norway	(null)	Oslo	39.62
9 Germany	(null)	Munich	37.62
10 Czech Republic	(null)	Prague	90.24
11 Finland	(null)	Helsinki	41.62
12 France	(null)	Bordeaux	39.62
13 USA	CA	Mountain View	77.24
14 Canada	NT	YellowKnife	37.62
15 Germany	(null)	Berlin	75.24
16 Canada	AB	Edmonton	37.62
17 Sweden	(null)	Stockholm	38.62
18 Canada	QC	Montreal	39.62
19 France	(null)	Lyon	37.62
20 Canada	ON	Toronto	37.62
21 Australia	NSW	Sidney	37.62
22 Canada	BC	Vancouver	37.62
23 Netherlands	VV	Amsterdam	40.62
24 USA	UT	Salt Lake City	43.62
25 France	(null)	Dijon	40.62

# Slowly Changing Dimensions in Data Warehouse

## Source



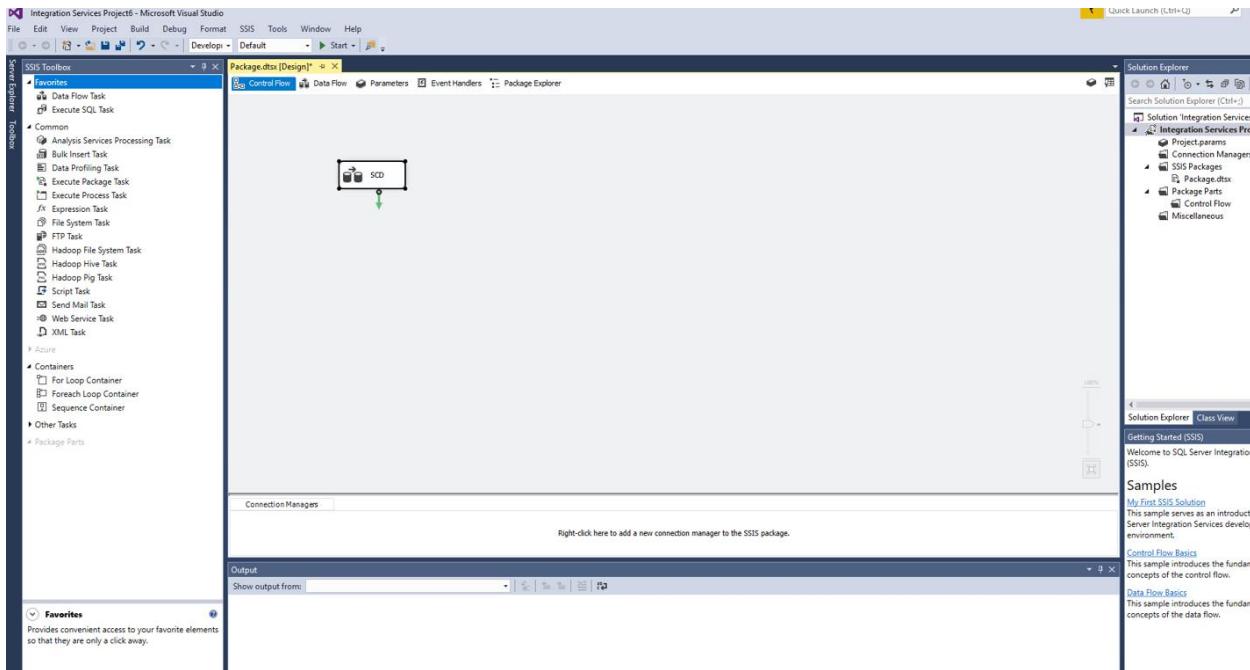
The screenshot shows the SQL Server Management Studio interface. In the center, there is a query window with three tabs: 'SQLQuery7.sql - DE...ASE (info6210 (63))', 'SQLQuery6.sql - DE...ASE (info6210 (60))', and 'SQLQuery5.sql - DE...ASE (info6210 (57))'. The 'SQLQuery7.sql' tab contains the following T-SQL code:

```
GO
CREATE TABLE [dbo].[EMPLOYEE]
(
    [EMP_NO] [int] NOT NULL,
    [NAME] [varchar](10) NULL,
    [SKILL] [varchar](10) NULL,
    [ADDRESS] [varchar](10) NULL,
    [PHONE] [varchar](10) NULL,
    PRIMARY KEY CLUSTERED
    (
        [EMP_NO] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
INSERT [dbo].[EMPLOYEE] ([EMP_NO], [NAME], [SKILL], [ADDRESS], [PHONE]) VALUES (101, N'NAME1', N'SKILL1', N'CITY1', N'PHONE1')
GO
INSERT [dbo].[EMPLOYEE] ([EMP_NO], [NAME], [SKILL], [ADDRESS], [PHONE]) VALUES (102, N'NAME2', N'SKILL2', N'NEW_CITY3', N'PHONE2')
GO
INSERT [dbo].[EMPLOYEE] ([EMP_NO], [NAME], [SKILL], [ADDRESS], [PHONE]) VALUES (103, N'NAME3', N'SKILL3', N'CITY3', N'PHONE3')
GO
INSERT [dbo].[EMPLOYEE] ([EMP_NO], [NAME], [SKILL], [ADDRESS], [PHONE]) VALUES (104, N'NAME4', N'SKILL4', N'CITY4', N'PHONE4')
GO
SELECT * FROM SOURCE_DATABASE.dbo.EMPLOYEE
```

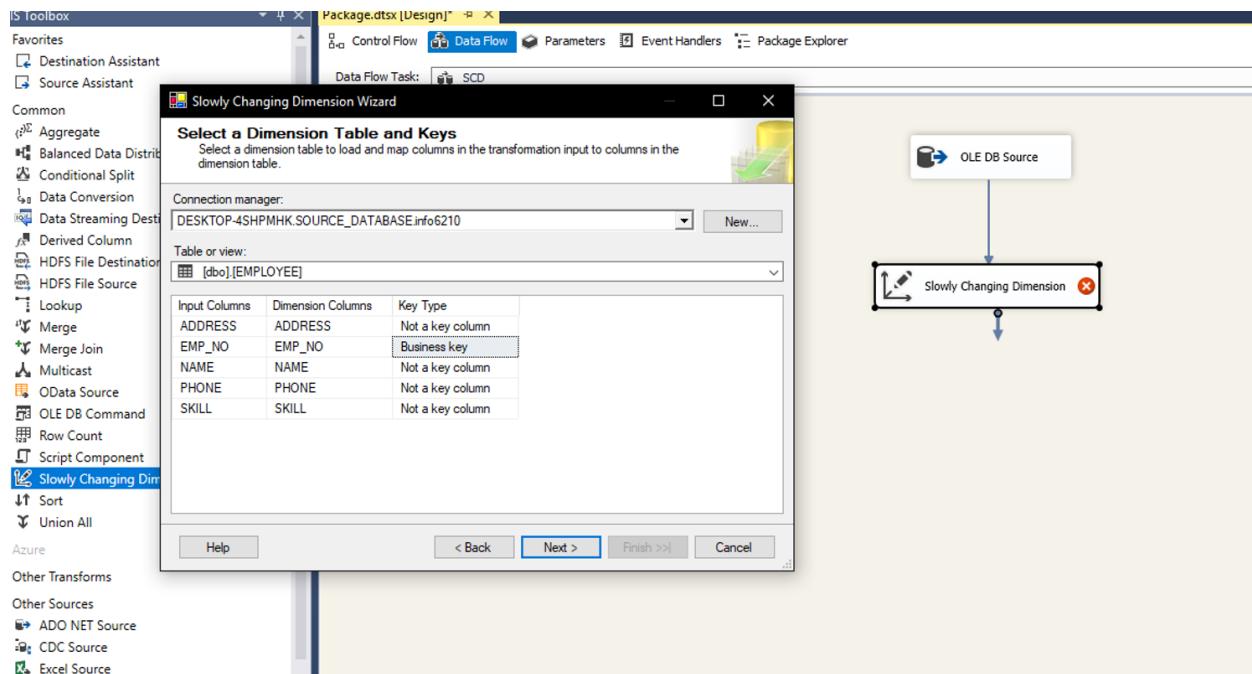
Below the code, the 'Results' tab displays a table with four rows of data:

	EMP_NO	NAME	SKILL	ADDRESS	PHONE
1	101	NAME1	SKILL1	CITY1	PHONE1
2	102	NAME2	SKILL2	NEW_CITY3	PHONE2
3	103	NAME3	SKILL3	CITY3	PHONE3
4	104	NAME4	SKILL4	CITY4	PHONE4

## Ssis



## SCD



## Scd attributes

 Slowly Changing Dimension Wizard

## Slowly Changing Dimension Columns

Manage the changes to column data in your slowly changing dimensions by setting the change type for dimension columns.

**Fixed Attribute**  
Select this type when the value in a column should not change. Changes are treated as errors.

**Changing Attribute**  
Select this type when changed values should overwrite existing values. This is a Type 1 change.

**Historical Attribute**  
Select this type when changes in column values are saved in new records. Previous values are saved in records marked as outdated. This is a Type 2 change.

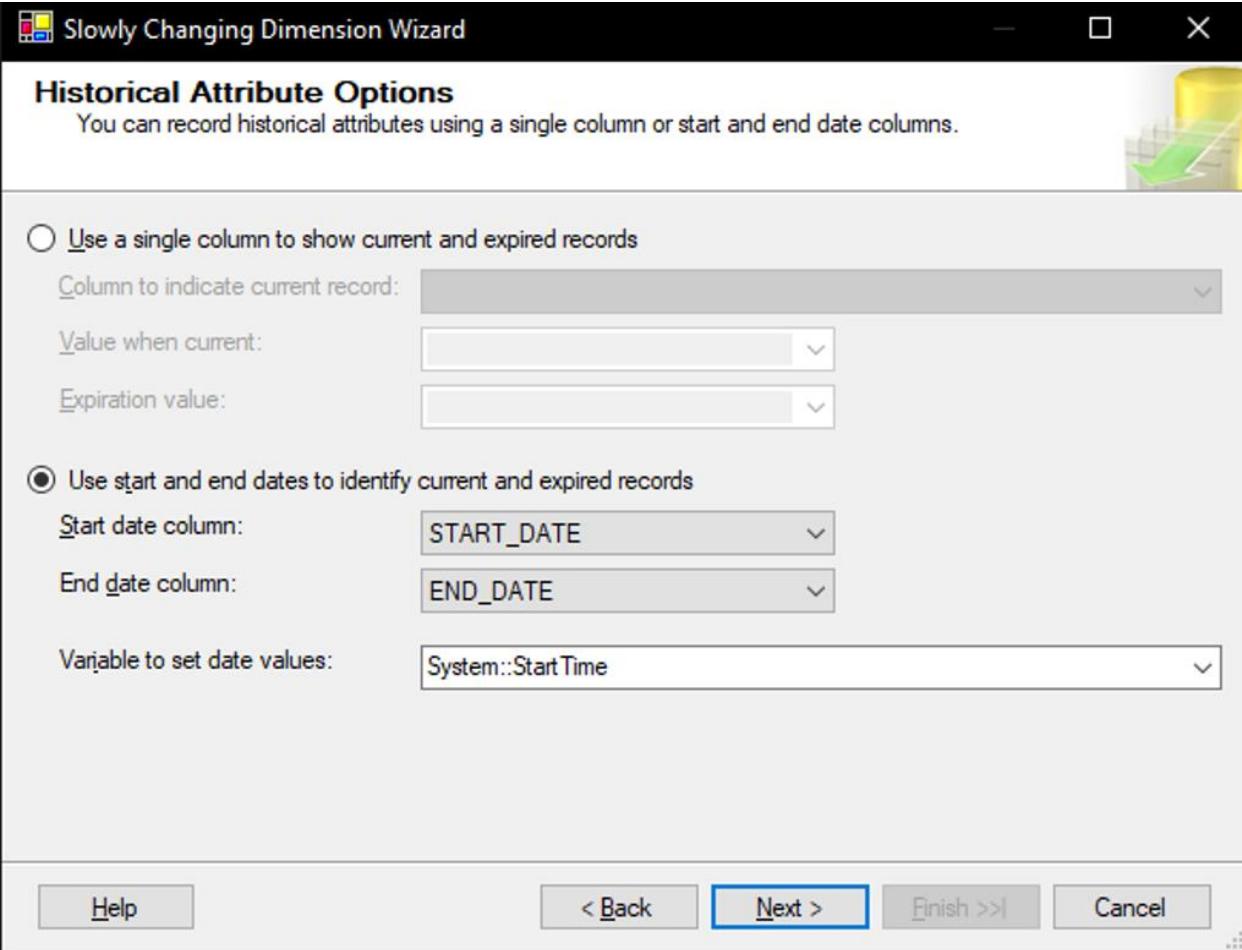
Select a change type for slowly changing dimension columns:

Dimension Columns	Change Type
ADDRESS	Historical att...
NAME	Fixed attribute
PHONE	Changing at...
SKILL	Fixed attribute

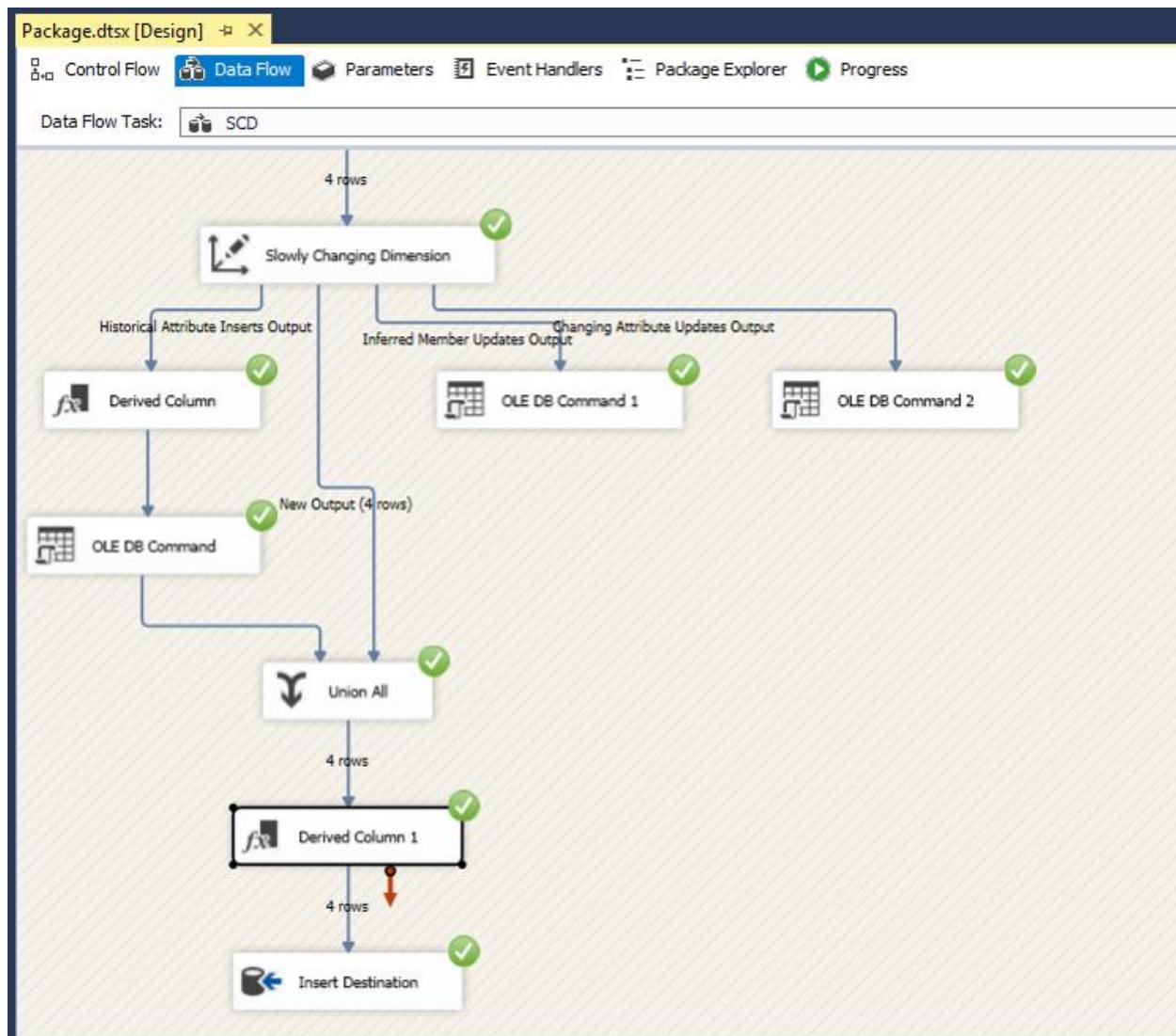
**Remove**

**Help** **< Back** **Next >** **Finish >>** **Cancel** 

Start n end date



Scd ssis



Displaying the first insertion

```

    select * from EMPLOYEE;
    --Do not execute untill directed
    -----
    -- for SCD 1 and 2 with current flag destination dimEMPLOYEE_2-----
    insert into Employee values(105,'NAME5','SKILL5','CITY5','PHONES')

    update Employee
    set Address = 'NEW_CITY5'
    where EMP_NO = 105

    update Employee
    set Phone = 'NEW_PHONE5'
    where EMP_NO = 105

    -- for SCD 1 and 2 with start and End date destination dimEMPLOYEE_1-----
    insert into Employee values(106,'NAME6','SKILL6','CITY6','PHONE6')

    update Employee
    set Address = 'NEW_CITY6'
    where EMP_NO = 106

    update Employee
    set Phone = 'NEW_PHONE6'
    where EMP_NO = 106

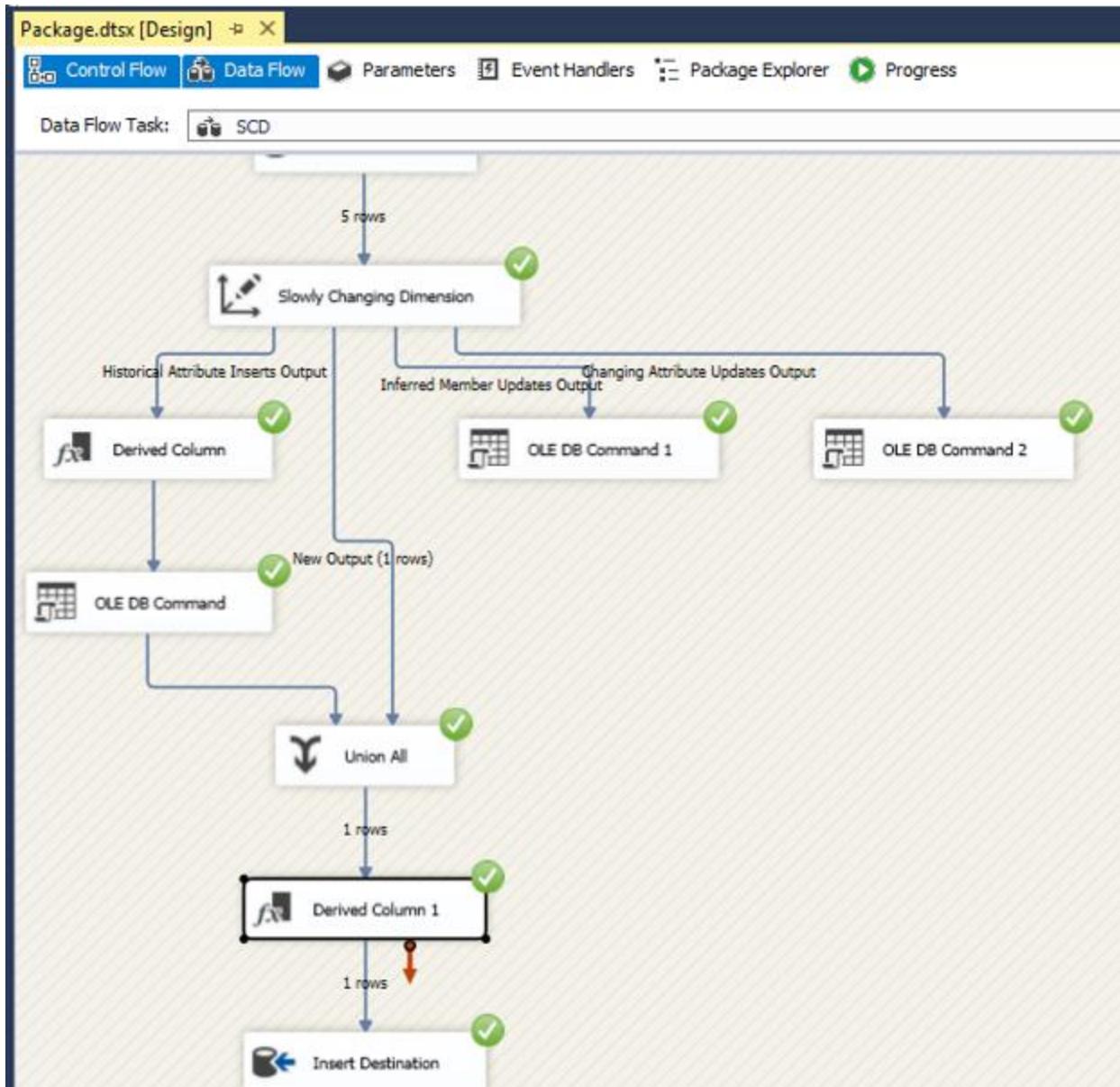
```

100 % <

Results Messages

	EMP_NO	NAME	SKILL	ADDRESS	PHONE
1	101	NAME1	SKILL1	CITY1	NEW_PHONE1
2	102	NAME2	SKILL2	NEW_CITY3	PHONE2
3	103	NAME3	SKILL3	CITY3	PHONE3
4	104	NAME4	SKILL4	CITY4	PHONE4
5	105	NAME5	SKILL5	CITY5	PHONE5

Running the ssis scd again after first insertion



Performing the first update on a historical attribute

SQLQuery7.sql - DE...ASE (info6210 (63))\*

SQLQuery6.sql - DE...ASE (info6210 (60))\*

```
select * from EMPLOYEE;
--Do not execute untill directed
-----
-- for SCD 1 and 2 with current flag destination dimEMPLOYEE_2-----
insert into Employee values(105,'NAME5','SKILL5','CITY5','PHONE5')

update Employee
set Address = 'NEW_CITY5'
where EMP_NO = 105

update Employee
set Phone = 'NEW_PHONE5'
where EMP_NO = 105

-- for SCD 1 and 2 with start and End date destination dimEMPLOYEE_1
insert into Employee values(106,'NAME6','SKILL6','CITY6','PHONE6')

update Employee
set Address = 'NEW_CITY6'
where EMP_NO = 106

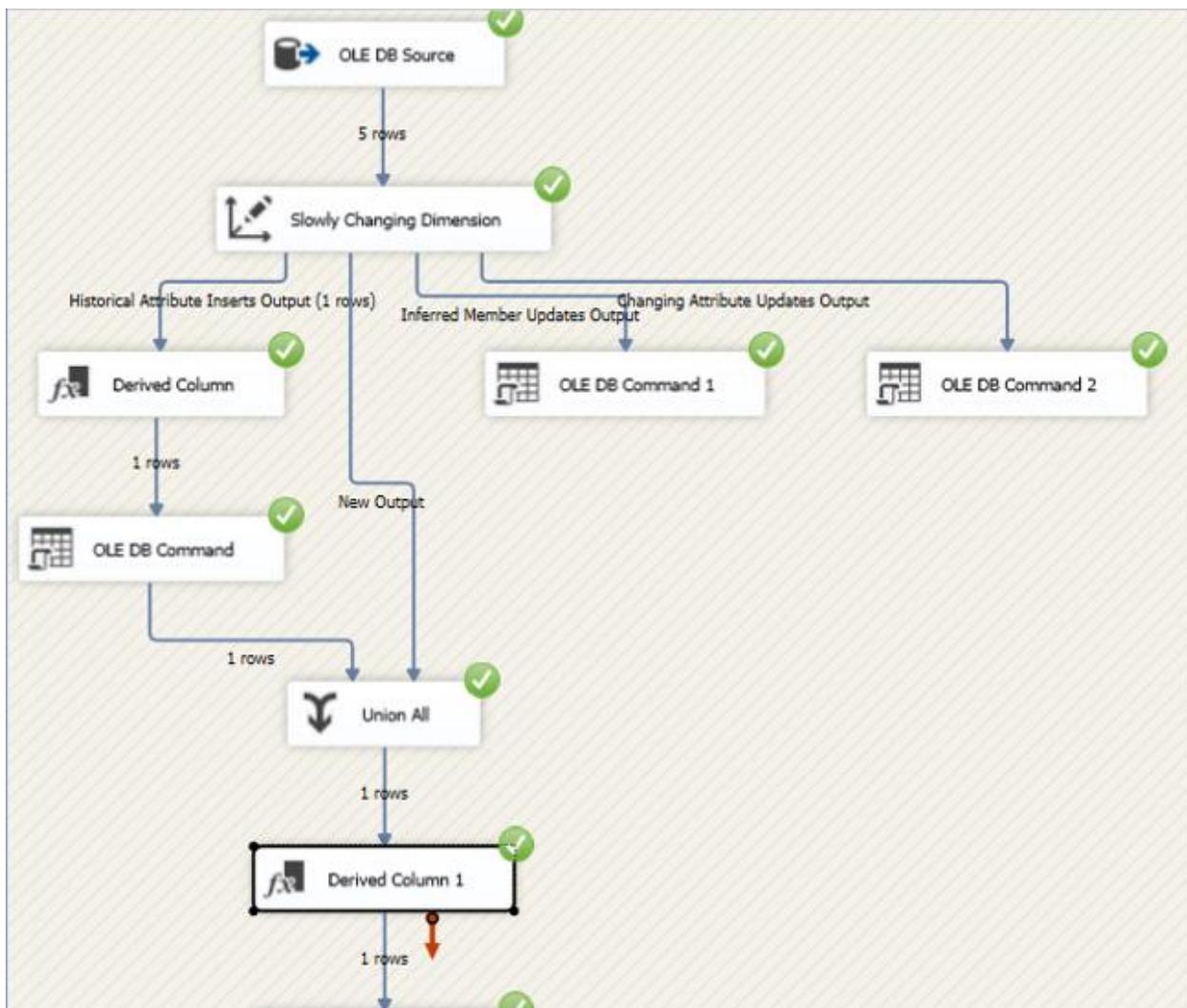
update Employee
set Phone = 'NEW_PHONE6'
where EMP_NO = 106
```

100 %

Results Messages

	EMP_NO	NAME	SKILL	ADDRESS	PHONE
1	101	NAME1	SKILL1	CITY1	NEW_PHONE1
2	102	NAME2	SKILL2	NEW_CITY3	PHONE2
3	103	NAME3	SKILL3	CITY3	PHONE3
4	104	NAME4	SKILL4	CITY4	PHONE4
5	105	NAME5	SKILL5	NEW_CITY5	PHONE5

Running the ssis scd again after the first update



Performing the second update on a changing attribute

SQLQuery7.sql - DE...ASE (info6210 (63)) \* X SQLQuery6.sql - DE...ASE (info6210 (60)) \*

```
select * from EMPLOYEE;
--Do not execute untill directed
-----
-- for SCD 1 and 2 with current flag destination dimEMPLOYEE_2-----
insert into Employee values(105,'NAME5','SKILLS5','CITY5','PHONES5')

update Employee
set Address = 'NEW_CITY5'
where EMP_NO = 105

update Employee
set Phone = 'NEW_PHONE5'
where EMP_NO = 105

-- for SCD 1 and 2 with start and End date destination dimEMPLOYEE_1--
insert into Employee values(106,'NAME6','SKILL6','CITY6','PHONE6')

update Employee
set Address = 'NEW_CITY6'
where EMP_NO = 106

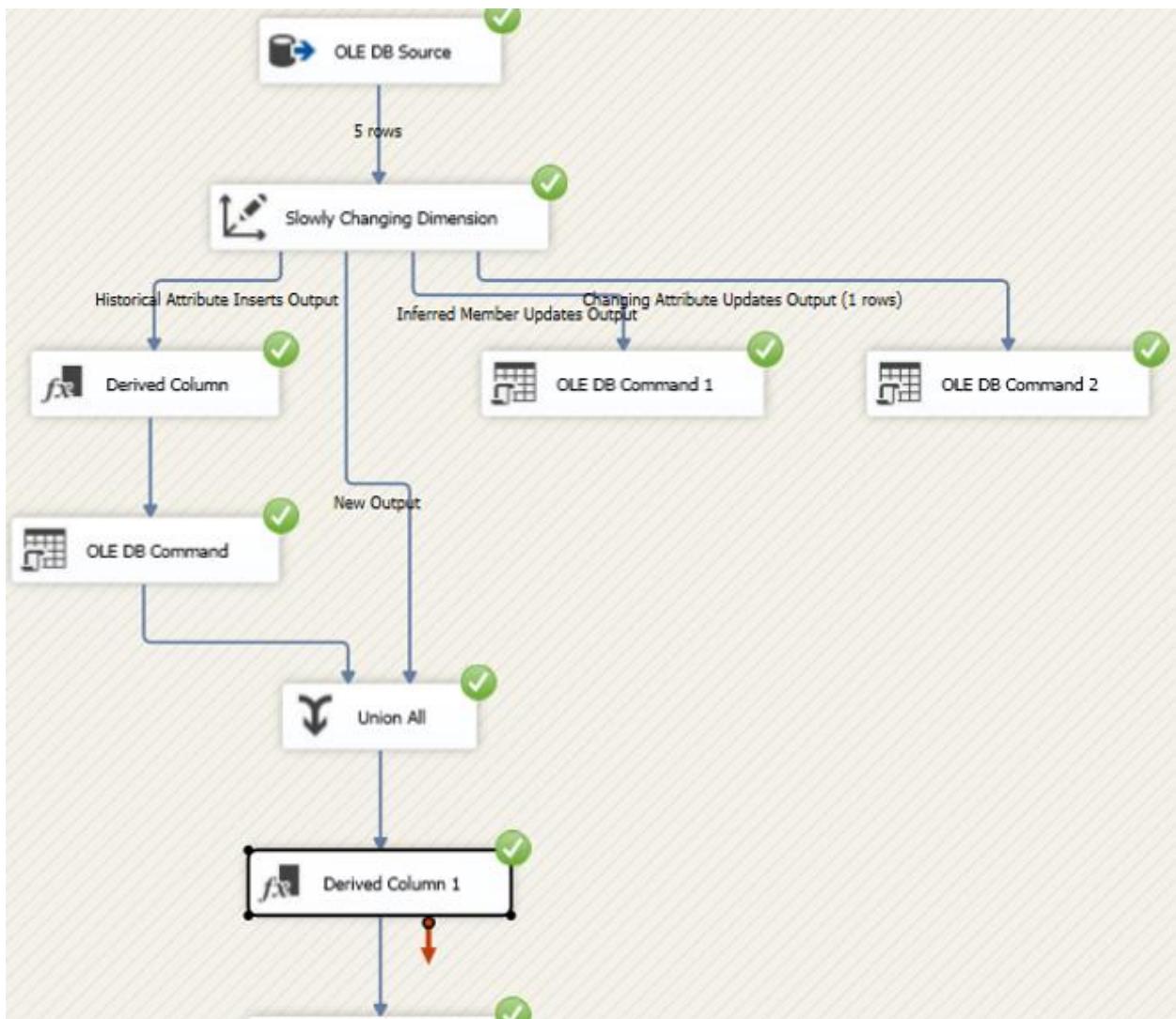
update Employee
set Phone = 'NEW_PHONE6'
where EMP_NO = 106
```

100 %

Results Messages

	EMP_NO	NAME	SKILL	ADDRESS	PHONE
1	101	NAME1	SKILL1	CITY1	NEW_PHONE1
2	102	NAME2	SKILL2	NEW_CITY3	PHONE2
3	103	NAME3	SKILL3	CITY3	PHONE3
4	104	NAME4	SKILL4	CITY4	PHONE4
5	105	NAME5	SKILLS5	NEW_CITY5	NEW_PHONE5

Running the ssis scd again to reflect the start n end date changes



Displaying the start n end date changes in ssms

```

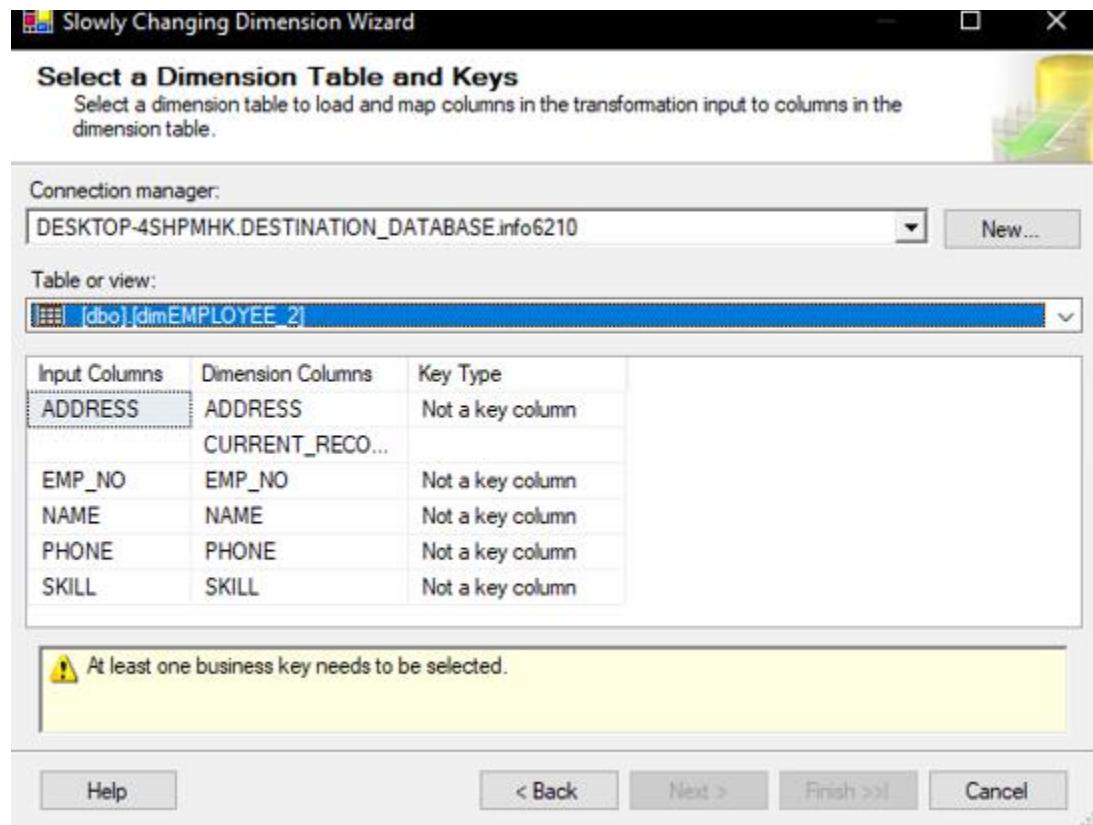
SELECT * FROM SOURCE_DATABASE.dbo.EMPLOYEE
SELECT * FROM DESTINATION_DATABASE.dbo.dimEMPLOYEE_1

```

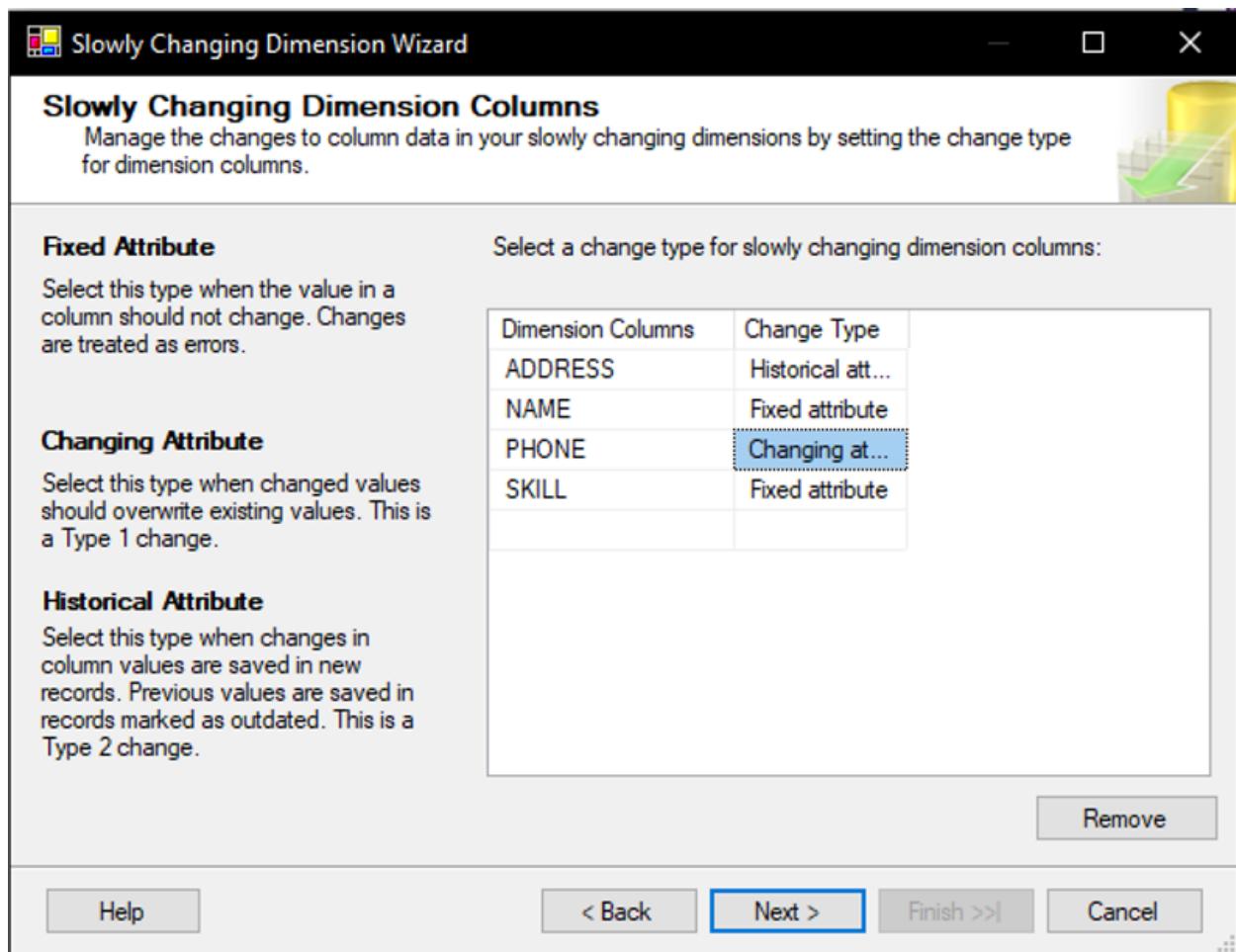
	EMP_KEY	EMP_NO	NAME	SKILL	ADDRESS	PHONE	START_DATE	END_DATE
1	1	101	NAME1	SKILL1	CITY1	NEW_PHONE1	2018-02-15 22:30:14.000	NULL
2	2	102	NAME2	SKILL2	NEW_CITY3	PHONE2	2018-02-15 22:30:14.000	NULL
3	3	103	NAME3	SKILL3	CITY3	PHONE3	2018-02-15 22:30:14.000	NULL
4	4	104	NAME4	SKILL4	CITY4	PHONE4	2018-02-15 22:30:14.000	NULL
5	5	105	NAME5	SKILL5	CITY5	PHONE5	2018-02-15 22:32:07.000	2018-02-15 22:33:40.000
6	6	105	NAME5	SKILL5	NEW_CITY5	NEW_PHONE5	2018-02-15 22:33:40.000	NULL

## Part 2

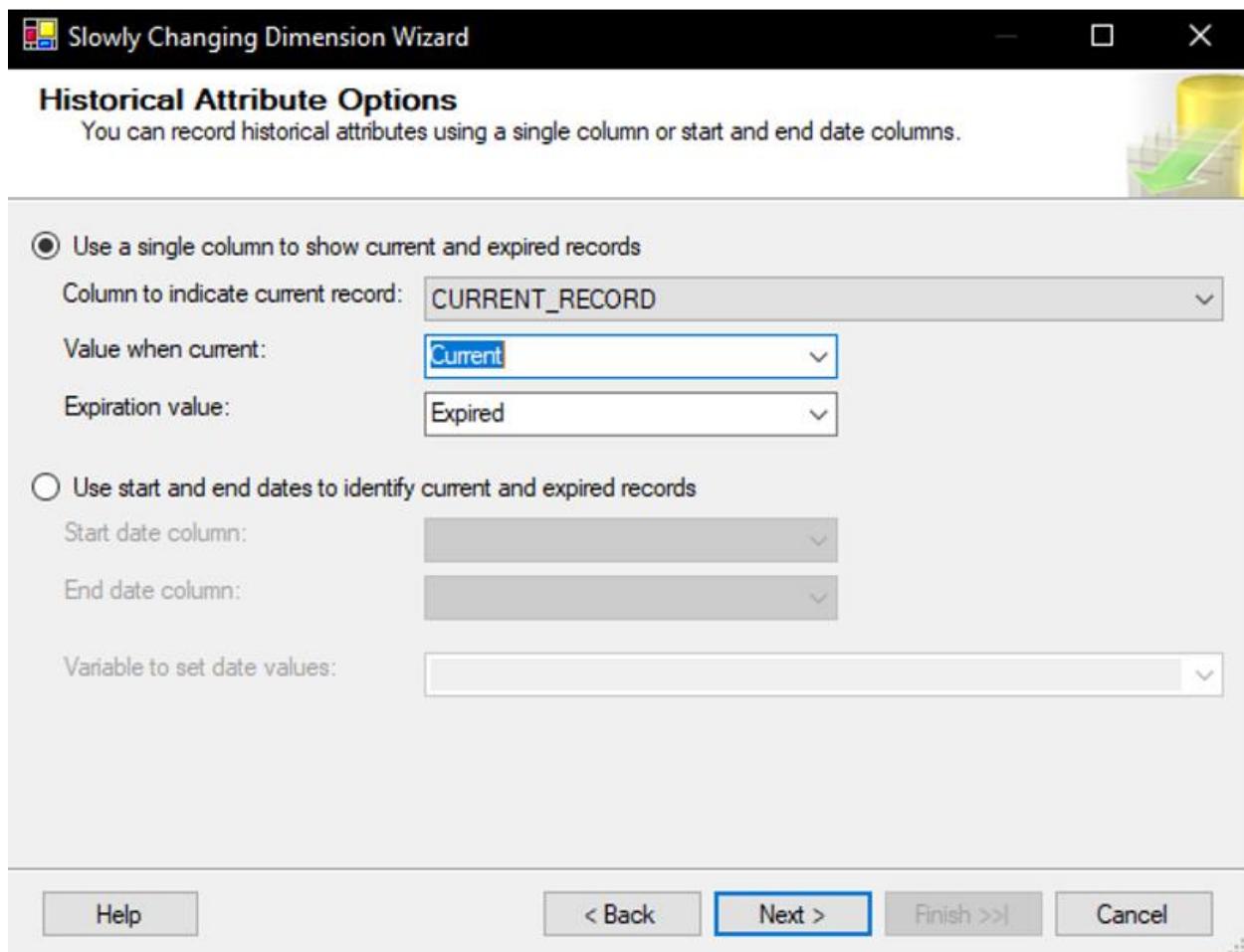
### DimEmployee\_2



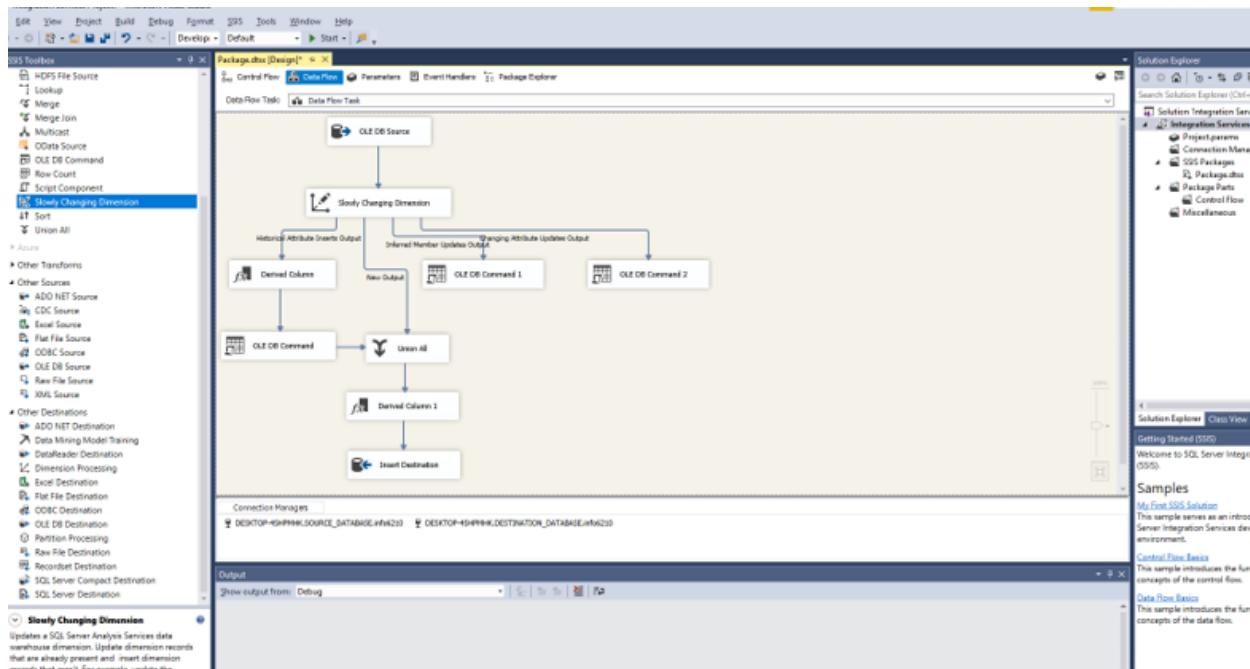
Setting attributes properties



Setting historical attributes properties



## Scd\_ssis



Inserting values into dimEmployee

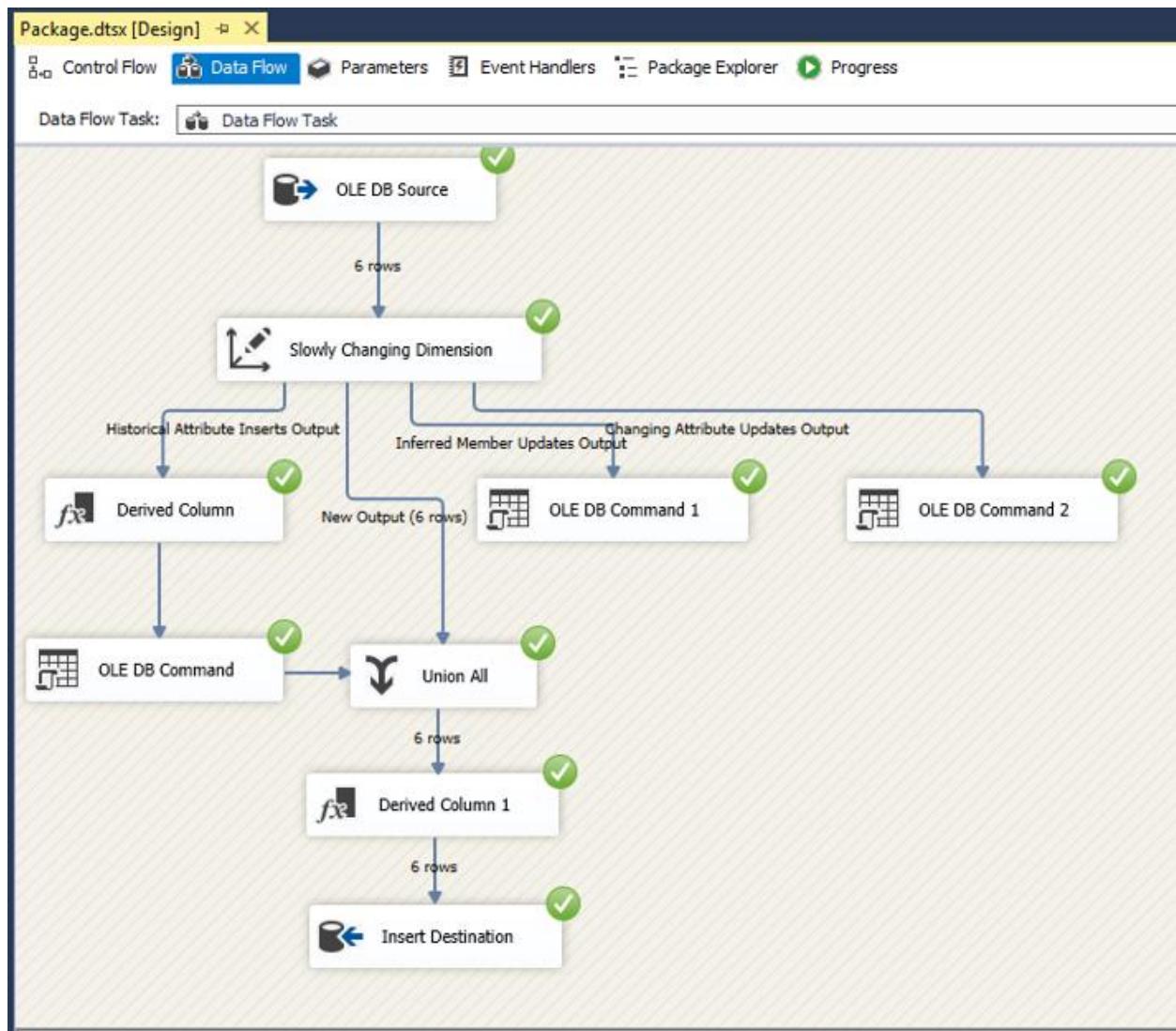
```
-- for SCD 1 and 2 with start and End date destination dimEMPLOYEE_1-----  
insert into Employee values(106,'NAME6','SKILL6','CITY6','PHONE6')  
  
update Employee  
set Address = 'NEW_CITY6'  
where EMP_NO = 106  
  
update Employee  
set Phone = 'NEW_PHONE6'  
where EMP_NO = 106
```

00 % <

Results Messages

	EMP_NO	NAME	SKILL	ADDRESS	PHONE
1	101	NAME1	SKILL1	CITY1	NEW_PHONE1
2	102	NAME2	SKILL2	NEW_CITY3	PHONE2
3	103	NAME3	SKILL3	CITY3	PHONE3
4	104	NAME4	SKILL4	CITY4	PHONE4
5	105	NAME5	SKILL5	NEW_CITY5	NEW_PHONE5
6	106	NAME6	SKILL6	CITY6	PHONE6

Running scd ssis after insertion



Displaying the scd properties in ssms

```
SELECT * FROM DESTINATION_DATABASE.dbo.dimEMPLOYEE_2
```

100 % ▾

Results Messages

	EMP_KEY	EMP_NO	NAME	SKILL	ADDRESS	PHONE	CURRENT_RECORD
1	1	101	NAME1	SKILL1	CITY1	NEW_PHONE1	Current
2	2	102	NAME2	SKILL2	NEW_CITY3	PHONE2	Current
3	3	103	NAME3	SKILL3	CITY3	PHONE3	Current
4	4	104	NAME4	SKILL4	CITY4	PHONE4	Current
5	5	105	NAME5	SKILL5	NEW_CITY5	NEW_PHONE5	Current
6	6	106	NAME6	SKILL6	CITY6	PHONE6	Current

Updating a historical attribute in ssms

```
update Employee
set Address = 'NEW_CITY6'
where EMP_NO = 106

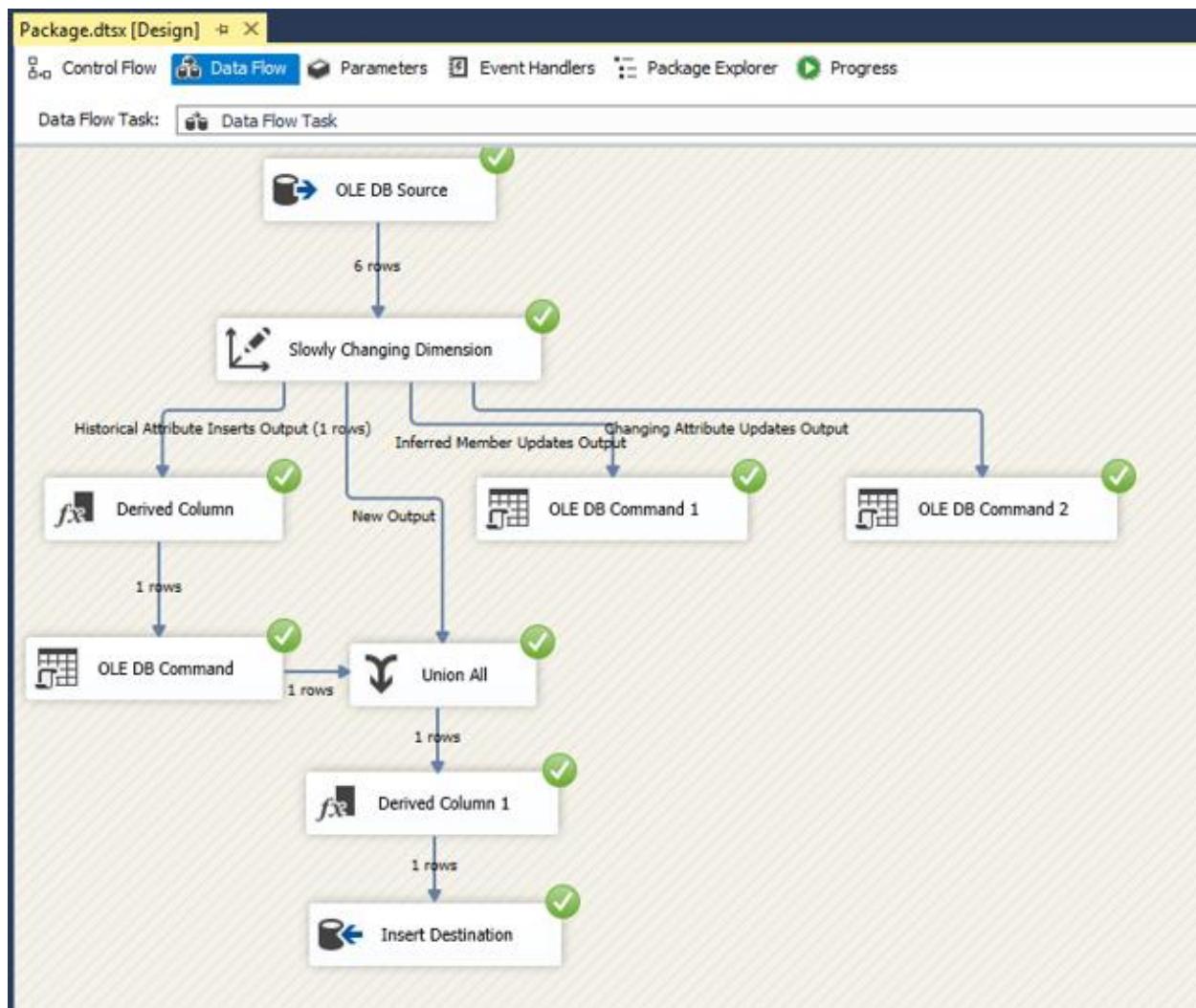
update Employee
set Phone = 'NEW_PHONE6'
where EMP_NO = 106
```

100 %

Results Messages

	EMP_NO	NAME	SKILL	ADDRESS	PHONE
1	101	NAME1	SKILL1	CITY1	NEW_PHONE1
2	102	NAME2	SKILL2	NEW_CITY3	PHONE2
3	103	NAME3	SKILL3	CITY3	PHONE3
4	104	NAME4	SKILL4	CITY4	PHONE4
5	105	NAME5	SKILL5	NEW_CITY5	NEW_PHONE5
6	106	NAME6	SKILL6	NEW_CITY6	PHONE6

Running the scd ssis again after the update



Displaying the scd attribute changes in current and expired

```
SELECT * FROM DESTINATION_DATABASE.dbo.dimEMPLOYEE_2
```

100 % ▶

Results Messages

	EMP_KEY	EMP_NO	NAME	SKILL	ADDRESS	PHONE	CURRENT_RECORD
1	1	101	NAME1	SKILL1	CITY1	NEW_PHONE1	Current
2	2	102	NAME2	SKILL2	NEW_CITY3	PHONE2	Current
3	3	103	NAME3	SKILL3	CITY3	PHONE3	Current
4	4	104	NAME4	SKILL4	CITY4	PHONE4	Current
5	5	105	NAME5	SKILL5	NEW_CITY5	NEW_PHONE5	Current
6	6	106	NAME6	SKILL6	CITY6	PHONE6	Expired
7	7	106	NAME6	SKILL6	NEW_CITY6	PHONE6	Current

Performing the last update on the changing attribute

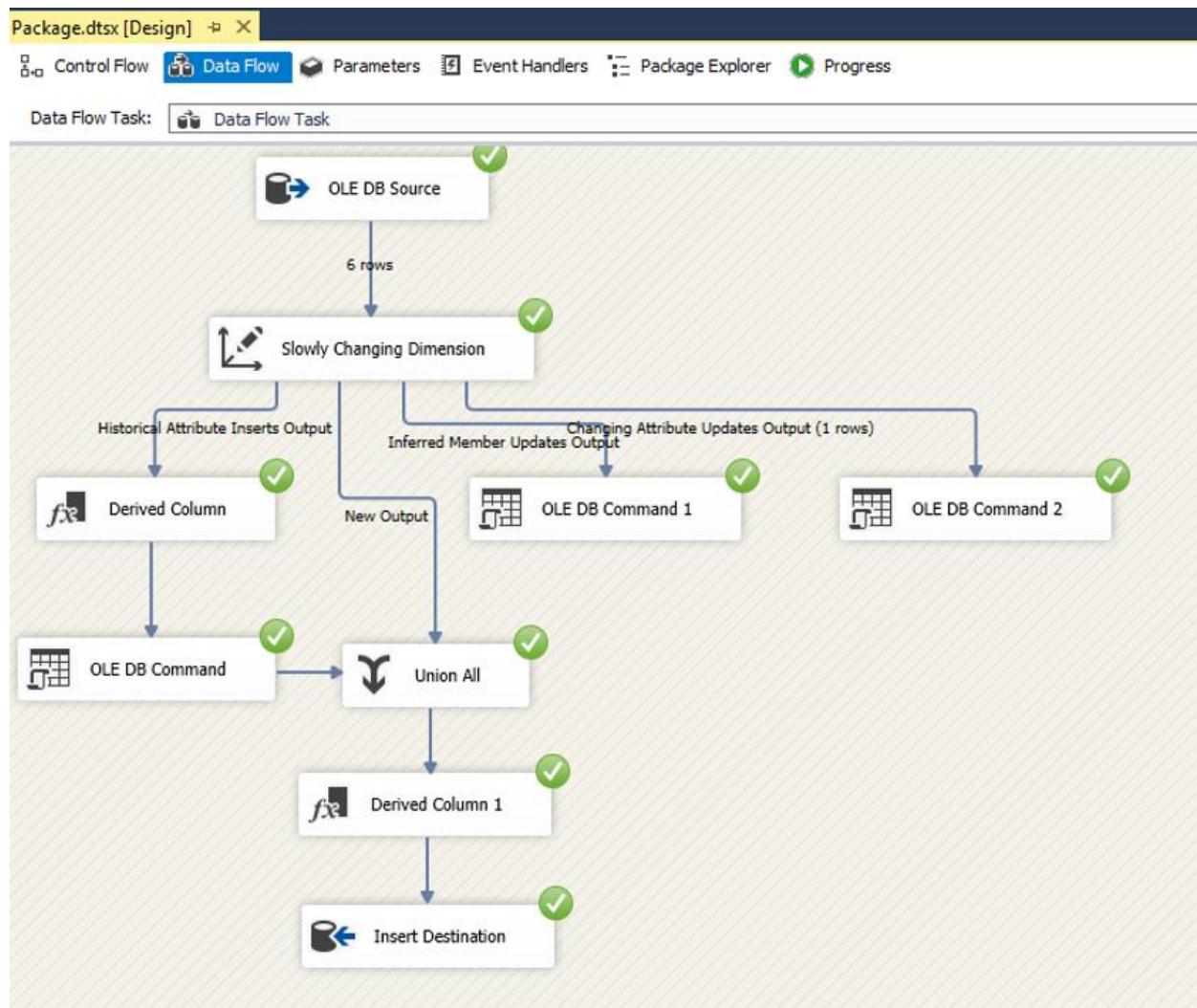
```
update Employee  
set Phone = 'NEW_PHONE6'  
where EMP_NO = 106
```

100 % <

Results Messages

	EMP_NO	NAME	SKILL	ADDRESS	PHONE
1	101	NAME1	SKILL1	CITY1	NEW_PHONE1
2	102	NAME2	SKILL2	NEW_CITY3	PHONE2
3	103	NAME3	SKILL3	CITY3	PHONE3
4	104	NAME4	SKILL4	CITY4	PHONE4
5	105	NAME5	SKILL5	NEW_CITY5	NEW_PHONE5
6	106	NAME6	SKILL6	NEW_CITY6	NEW_PHONE6

Running the scd ssis again after the update for reflecting the changes



Displaying the SCD transformation on historical and Changing attributes in SSMS

```
SELECT * FROM DESTINATION_DATABASE.dbo.dimEMPLOYEE_2
```

100 % ▾

Results Messages

	EMP_KEY	EMP_NO	NAME	SKILL	ADDRESS	PHONE	CURRENT_RECORD
1	1	101	NAME1	SKILL1	CITY1	NEW_PHONE1	Current
2	2	102	NAME2	SKILL2	NEW_CITY3	PHONE2	Current
3	3	103	NAME3	SKILL3	CITY3	PHONE3	Current
4	4	104	NAME4	SKILL4	CITY4	PHONE4	Current
5	5	105	NAME5	SKILL5	NEW_CITY5	NEW_PHONE5	Current
6	6	106	NAME6	SKILL6	CITY6	PHONE6	Expired
7	7	106	NAME6	SKILL6	NEW_CITY6	NEW_PHONE6	Current

\*\*\*\*\*

\*\*\*

## SQL Queries

- --Q1>o Ranked order of Vendors by purchase amount \$
- ```
SELECT pv.Name AS "Vendor Name",
      SUM(pro.ListPrice) AS "Purchase Amount",
      RANK() OVER (ORDER BY sum(pro.ListPrice) DESC) AS "Rank"
     FROM Production.Product AS pro
    JOIN Purchasing.ProductVendor AS prov
   ON pro.ProductID=prov.ProductID
  JOIN Purchasing.Vendor AS pv
 ON prov.BusinessEntityID=pv.BusinessEntityID
 GROUP BY pv.Name
 ORDER BY "Rank";
```

--Q2>o Ranked order of products purchased by amount \$

```
SELECT pro.Name AS "Product Name",
pro.ListPrice AS "Purchased Amount",
```

```
RANK() OVER (ORDER BY pro.ListPrice DESC) AS "Rank"  
FROM Production.Product AS pro;
```

```
--Q2.2> ④ By category  
SELECT pro.Name AS "Product Name",  
SUM(pro.ListPrice) AS "Purchased Amount",  
pc.Name AS "Category",  
RANK() OVER (ORDER BY SUM(pro.ListPrice) DESC) AS "Rank"  
FROM Production.Product AS pro  
JOIN Production.ProductSubcategory AS ps  
ON pro.ProductSubcategoryID=ps.ProductSubcategoryID  
JOIN Production.ProductCategory AS pc  
ON ps.ProductCategoryID=pc.ProductCategoryID  
GROUP BY pro.Name, pc.Name  
ORDER BY "Rank";
```

```
--Q2.3> ④ By subcategory  
SELECT pro.Name AS "Product Name",  
SUM(pro.ListPrice) AS "Purchased Amount",ps.Name AS "SubCategory",  
RANK() OVER (ORDER BY SUM(pro.ListPrice) DESC) AS "Rank"  
FROM Production.Product AS pro  
JOIN Production.ProductSubcategory AS ps  
ON pro.ProductSubcategoryID=ps.ProductSubcategoryID  
GROUP BY pro.Name, ps.Name  
ORDER BY "Rank";
```

```
--Q2.4> ④ By product model (top 20)  
select TOP 20 pm.Name,sum(po.TotalDue) as 'Amount',  
rank() over (order by sum(po.TotalDue)) Rank  
from Production.ProductModel pm  
join Production.Product p on p.ProductModelID=pm.ProductModelID  
join Purchasing.PurchaseOrderDetail pd on pd.ProductID=p.ProductID  
join Purchasing.PurchaseOrderHeader po on po.PurchaseOrderID=pd.PurchaseOrderID  
group by pm.Name  
order by rank;
```

```
--Q2.5> ④ By product (top 20)  
select top 20 p.Name,sum(po.TotalDue) as 'Amount',  
rank() over (order by sum(po.TotalDue)) Rank  
from Production.Product p  
join Purchasing.PurchaseOrderDetail pd on pd.ProductID=p.ProductID  
join Purchasing.PurchaseOrderHeader po on po.PurchaseOrderID=pd.PurchaseOrderID  
group by p.Name  
order by rank;
```

```
--Q3> o      List of employees who purchased products with phone, email & address
```

```

select p.firstname,p.LastName,ph.PhoneNumber,em.EmailAddress,
concat(ad.AddressLine1,ad.AddressLine2,ad.City,ad.PostalCode) as Address,
rank() over (order by p.firstname) Rank
from Person.person p
join HumanResources.Employee e on e.BusinessEntityID=p.BusinessEntityID
join Person.PersonPhone ph on ph.BusinessEntityID=p.BusinessEntityID
join Person.EmailAddress em on em.BusinessEntityID=p.BusinessEntityID
join Person.BusinessEntityAddress bd on bd.BusinessEntityID=p.BusinessEntityID
join person.Address ad on ad.AddressID=bd.AddressID
order by Rank

```

--Q4> o List of employees who purchased products with pay rate & raises (SCD)

```

select (pps.FirstName +space(1)+pps.MiddleName+space(1)+pps.LastName)as 'employeename',
eph.Rate as 'rate',(eph.PayFrequency)
from Purchasing.PurchaseOrderHeader ph
join Purchasing.PurchaseOrderDetail po
on ph.PurchaseOrderID=po.PurchaseOrderID
join HumanResources.Employee he
on ph.EmployeeID=he.BusinessEntityID
join Person.Person pps
on pps.BusinessEntityID=he.BusinessEntityID
join HumanResources.EmployeePayHistory eph
on eph.BusinessEntityID=he.BusinessEntityID
join HumanResources.EmployeePayHistory efg
on eph.BusinessEntityID=efg.BusinessEntityID
join HumanResources.EmployeePayHistory esg
on efg.BusinessEntityID=esg.BusinessEntityID
where efg.Rate<eph.Rate and efg.Rate>esg.rate
group by pps.FirstName,pps.LastName,pps.MiddleName,eph.PayFrequency,eph.rate;

```

--Q5> o List of purchasing vendor contacts with  
-- vendor name, phone, email & address

```

SELECT
    distinct
    pv.Name as 'Vendor Name',
    ad.AddressLine1 as Address,
    ad.City,
    ad.PostalCode,
    sp.Name as State,
    ead.EmailAddress,
    pp.PhoneNumber

FROM
    Purchasing.Vendor pv,
    Person.Address ad,
    Person.BusinessEntity BE,
    Person.BusinessEntityContact bec,
    Person.BusinessEntityAddress bea,
    Person.EmailAddress ead,

```

```

Person.StateProvince sp,
Person.PersonPhone pp,
Person.Person p
Where
pv.BusinessEntityID=be.BusinessEntityID
AND
be.BusinessEntityID=bea.BusinessEntityID
AND
bea.AddressID=ad.AddressID
AND
ad.StateProvinceID=sp.StateProvinceID
AND
BE.BusinessEntityID=BEC.BusinessEntityID
AND
BEC.PersonID = p.BusinessEntityID
AND
p.BusinessEntityID=ead.BusinessEntityID
AND
p.BusinessEntityID=pp.BusinessEntityID
And p.PersonType='VC'
Order By
'Vendor Name';

```

```

--Q6> o      List of product prices by product order by product
--          and SCD effective ascending
SELECT
    pro.Name as 'Product Name',
    pro.ListPrice as Price
FROM
    Production.Product pro
WHERE
    pro.FinishedGoodsFlag=1
GROUP BY
    Pro.Name,
    Pro.ListPrice
ORDER BY
    pro.Name;

```

```

--Q7> o      List of standard costs by product order by product
--          and SCD effective ascending
SELECT
    pro.Name as 'Product Name',
    pro.StandardCost
FROM
    Production.Product pro
WHERE
    pro.FinishedGoodsFlag=1
GROUP BY
    Pro.Name,
    Pro.StandardCost
ORDER BY

```

```
pro.Name;
```

---

## *Data Visualizations*

Bikes

Accessories

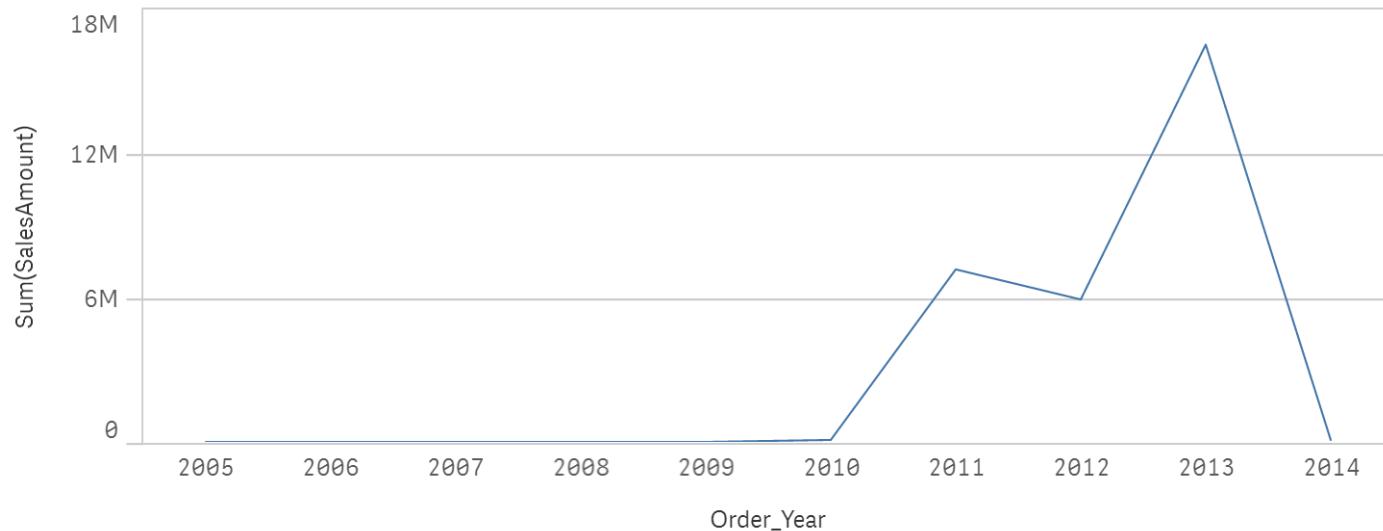
Total Sales

29.36M

\* The data set contains negative or zero values that cannot be shown in this chart.



### Trend by Year-Month Drilldown

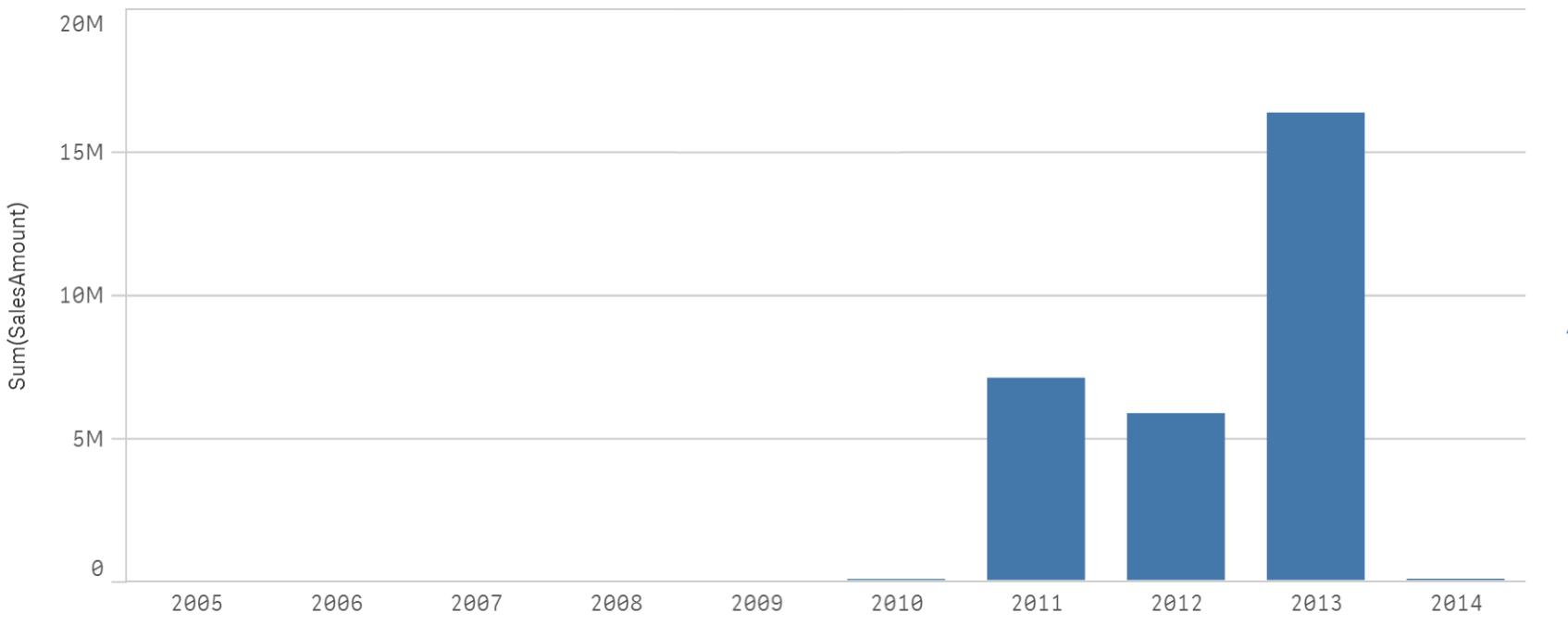


Total Sales

**29.36M**

### Geo Contribution to Sales

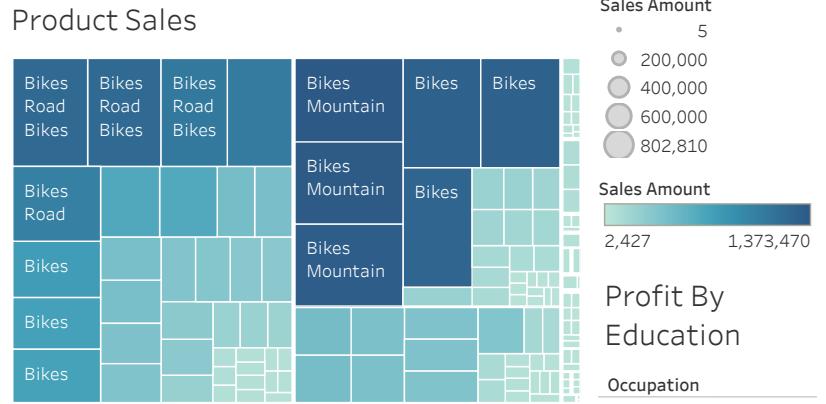




### Geo Sales



### Product Sales



### Sales Amount

- 5
- 200,000
- 400,000
- 600,000
- 802,810

### Sales Amount

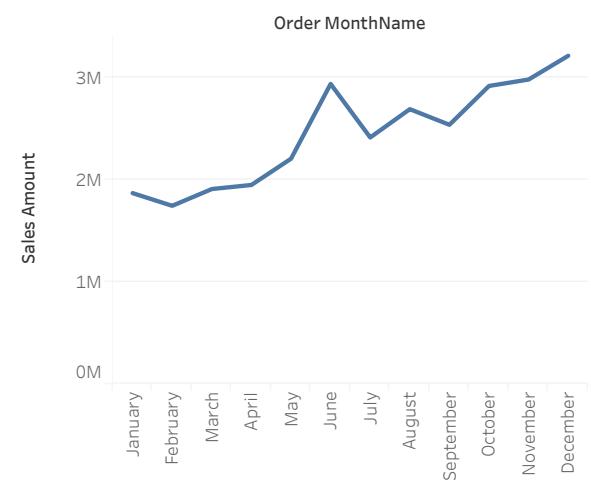
- 2,427 1,373,470

### Profit By Education

#### Occupation

|               |           |
|---------------|-----------|
| Clerical      | 1,910,436 |
| Manageme..    | 2,260,335 |
| Manual        | 1,170,020 |
| Professional  | 4,099,568 |
| Skilled Man.. | 2,640,524 |

### Sales Order



### Sales by Education & Occupation

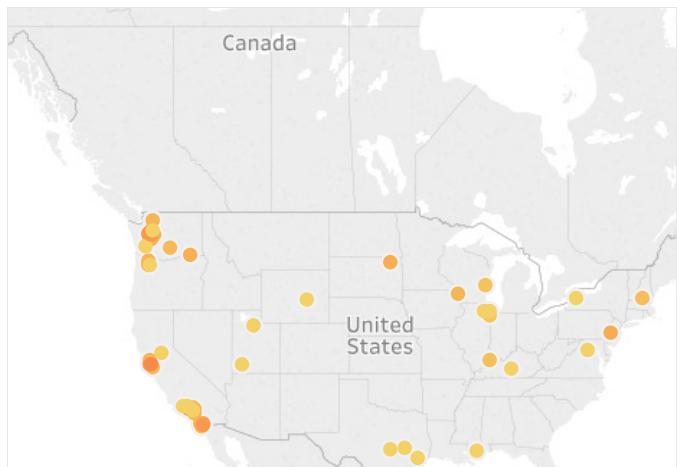
#### Education

|                |    |
|----------------|----|
| Bachelors      | 5M |
| Graduate D..   | 5M |
| High School    | 5M |
| Partial Coll.. | 5M |
| Partial High.. | 0M |

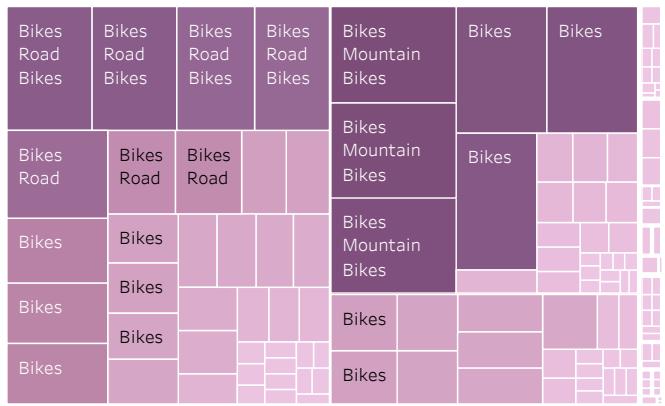
#### Sales Amount

- 0M 5M

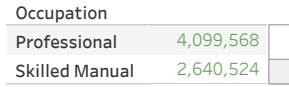
## Product Sales by Geo



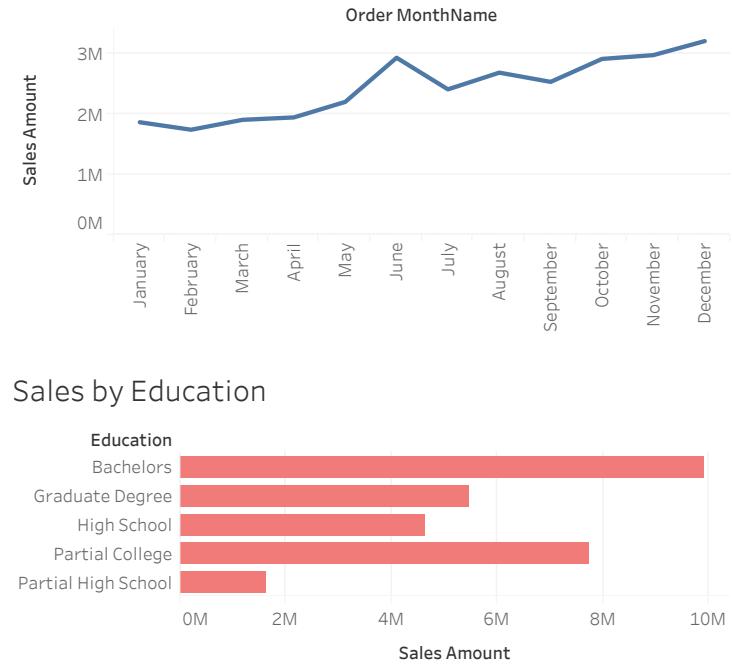
## Product Contribution of Sales



## Profit By Education

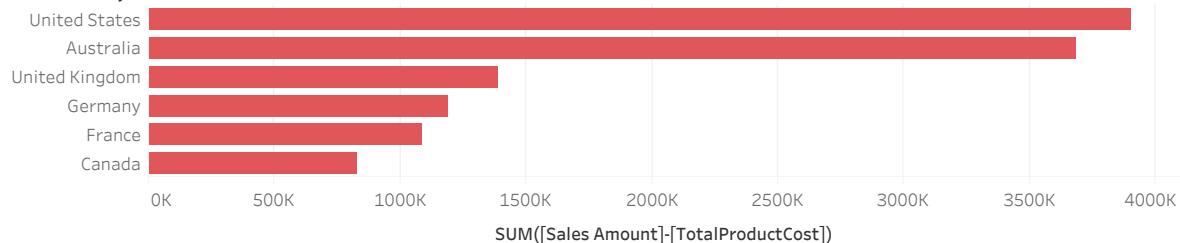


## Sales Trends & Forecast by Month



## Product Profits by Sales Territory

### Sales Territory ..



Discount Pct  
0.00 42.36

## Sales Promotions & Discounts

### Promotion ..

|    |       |
|----|-------|
| 1  | 0.00  |
| 2  | 42.36 |
| 13 | 3.00  |
| 14 | 2.60  |

## Comparison of Yearly Sales

