

ETS DATA LAKEHOUSE



NAMA : RETHA NOVIANTY SIPAYUNG

NRP : 5026211028

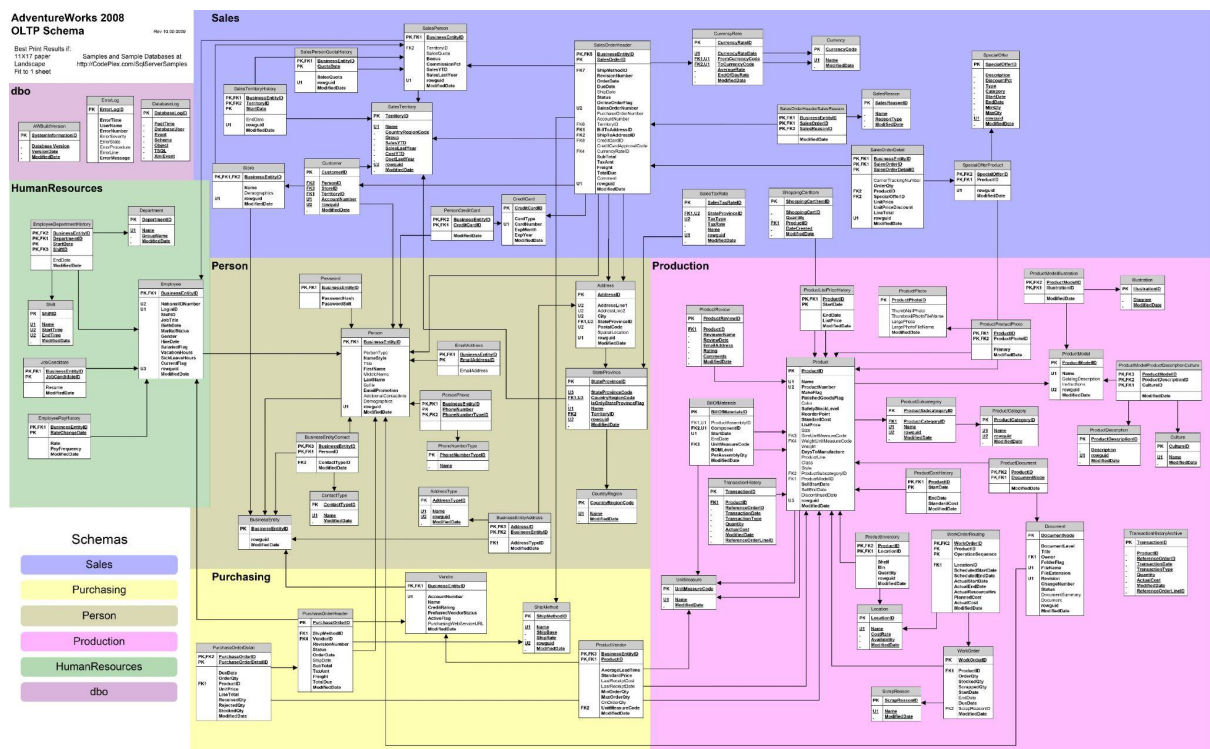
KELAS : DLH (B)

DESAIN & IMPLEMENTASI DATA WAREHOUSE

a. Menjelaskan Proses Bisnis

- Deteksi proses bisnis selengkap2nya, kalian bisa lihat dari OLTP, lalu menganalogikan dengan proses bisnis yang ada di Perusahaan manufaktur, ataupun di retail
- Hubungkan setiap komponen di bisnis proses dengan table yang ada di OLTP, sehingga anda tahu proses bisnis ini akan menggunakan table apa.

OLTP :



Proses Bisnis :

Ada beberapa proses bisnis dalam database AdventureWorks2016, yaitu :

Proses Bisnis	Deskripsi
Manufacturing	Manufaktur adalah proses produksi produk atau komponen. Proses ini dimulai dengan pembelian bahan baku dari pemasok, dilanjutkan dengan penerimaan bahan baku, penyimpanan bahan baku, produksi, dan pengiriman produk ke pelanggan.
Sales	Sales adalah proses penjualan produk atau layanan kepada pelanggan. Proses ini dimulai dengan pemasaran produk atau layanan, dilanjutkan dengan pembuatan penawaran, penerimaan pesanan penjualan, pengiriman produk, dan penerimaan pembayaran.

Inventory	Inventory mencakup proses pemantauan, penyimpanan, dan pengeluaran persediaan. Pemantauan tingkat persediaan dilakukan untuk memastikan bahwa perusahaan memiliki persediaan barang yang cukup untuk memenuhi permintaan pelanggan. Penyimpanan persediaan dilakukan untuk menjaga agar barang tetap dalam kondisi baik dan aman. Pengeluaran persediaan dilakukan untuk memenuhi permintaan pelanggan atau untuk keperluan produksi.
Purchasing	Purchasing mencakup proses identifikasi kebutuhan, pemilihan vendor, negosiasi, pemesanan, dan penerimaan barang. Identifikasi kebutuhan dilakukan untuk menentukan barang apa saja yang dibutuhkan oleh perusahaan. Pemilihan vendor dilakukan untuk memilih vendor yang tepat untuk memenuhi kebutuhan barang tersebut. Negosiasi dilakukan untuk menentukan harga dan persyaratan pembelian dengan vendor. Pemesanan dilakukan untuk memesan barang kepada vendor. Penerimaan barang dilakukan untuk menerima barang dari vendor.
Customer Services	Customer Services adalah proses penanganan pertanyaan, keluhan, dan masalah pelanggan. Proses ini dimulai dengan penerimaan pertanyaan, dilanjutkan dengan jawaban pertanyaan, penerimaan keluhan, penyelesaian keluhan, dan penerimaan masalah.
Human Resources	Human resources adalah proses perekrutan, pelatihan, dan pengembangan karyawan. Proses ini dimulai dengan perekrutan karyawan baru, dilanjutkan dengan orientasi karyawan baru, pelatihan karyawan, penilaian kinerja karyawan, pengembangan karir karyawan, dan pemberian tunjangan.
Finance	Finance adalah proses akuntansi, pelaporan keuangan, dan perencanaan anggaran. Proses ini dimulai dengan pencatatan transaksi keuangan, dilanjutkan dengan penyusunan laporan keuangan, pembayaran kepada pemasok dan karyawan, penerimaan pembayaran dari pelanggan, investasi modal, dan pembiayaan modal.

b. Mendeklarasikan Granularitas

- Lakukan setiap proses bisnis
- Kalian harus memilih granularitas terdetail sebisa mungkin

Proses Bisnis	Granularitas	Reasoning
Manufacturing	1 row per product	Granularitas 1 row per product dipilih karena memungkinkan untuk melacak informasi lengkap tentang setiap produk yang diproduksi, termasuk bahan baku yang digunakan dan proses produksi yang digunakan
Sales	1 row per order	Memungkinkan untuk melacak informasi lengkap tentang setiap pesanan yang dibuat, termasuk

		pelanggan yang membuat pesanan dan produk yang dipesan
Inventory	1 row per product	Memungkinkan untuk melacak informasi lengkap tentang setiap produk yang ada di perusahaan, termasuk jumlah persediaan yang tersedia.
Purchasing	1 row per purchase order	Memungkinkan untuk melacak informasi lengkap tentang setiap pesanan pembelian yang dibuat, termasuk vendor yang dibeli dan produk yang dibeli
Customer Services	1 row per case	Memungkinkan untuk melacak informasi lengkap tentang setiap kasus layanan pelanggan yang dibuka, termasuk pelanggan yang terlibat dan produk yang terlibat
Human Resources	1 row per employee	Dapat melacak informasi lengkap tentang setiap karyawan di perusahaan, termasuk: nama karyawan, alamat karyawan, nomor telepon karyawan, email karyawan, posisi karyawan, dan gaji karyawan.
Finance	1 row per financial transaction	Memungkinkan untuk melacak informasi lengkap tentang setiap transaksi keuangan yang terjadi di perusahaan, termasuk tanggal transaksi dan jenis transaksi

c. Menentukan Dimensi

- Tentukan dimensi berdasarkan paparan proses bisnis yang kalian definisikan.
- Tahapan ini paling krusial, karena sering terlalu rigid atau keliru antara dimensi dan fakta

Proses Bisnis	Dimensi
Manufacturing	Date, Store, Product, Employee, Vendor, Material
Sales	Date, Product, Employee, Store, Customer
Inventory	Date, Product, Employee, Vendor
Purchasing	Date, Product, Vendor
Customer Services	Date, Employee, Customer, Product
Human Resources	Date, Employee, Department

Finance	Date, Account, Department, Vendor, Currency
---------	---

d. Menentukan Fakta

- Fokus pada fakta numerik yang bisa ditemukan pada proses bisnis (bukan yang diharapkan oleh laporan saja)

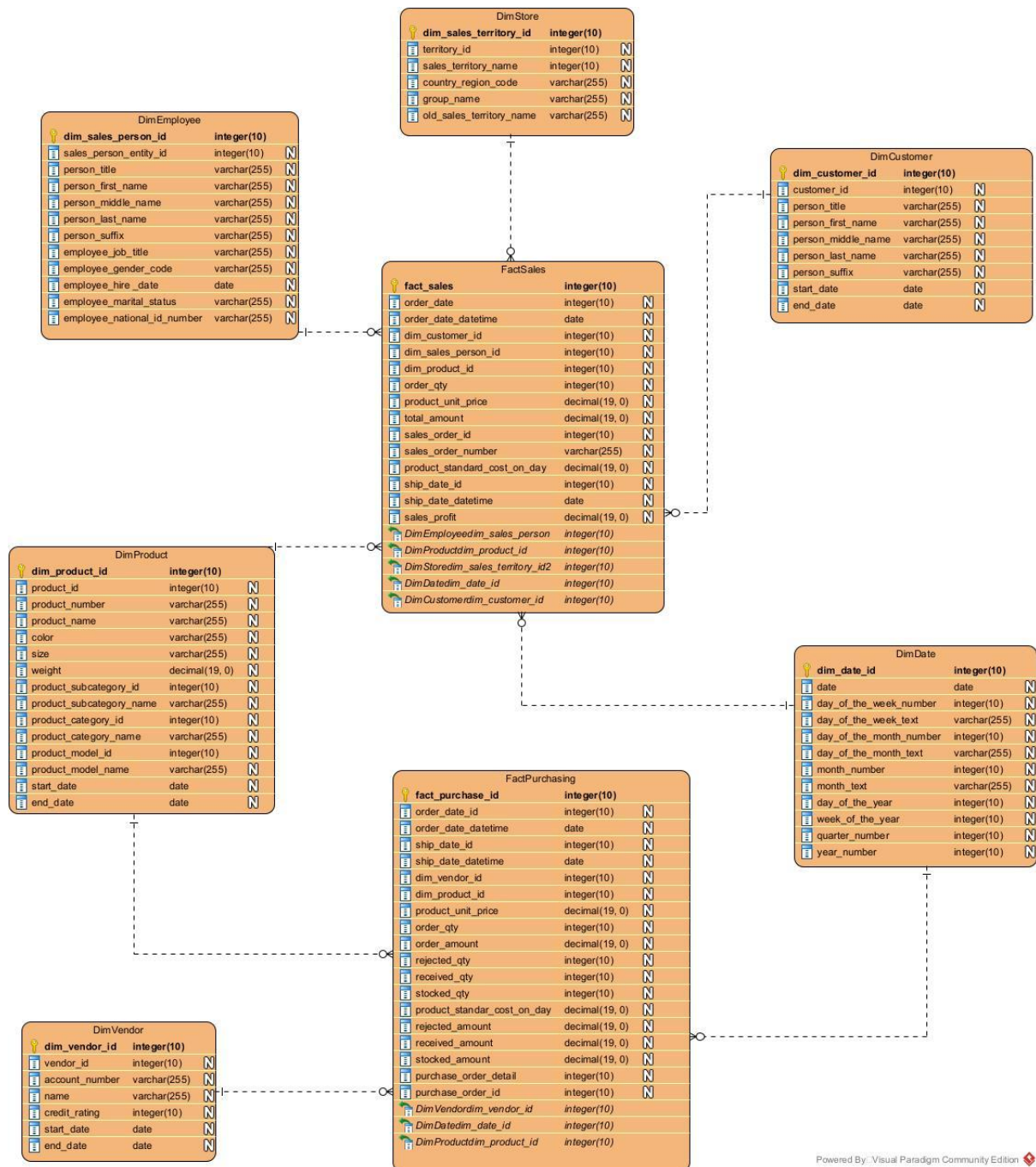
Proses Bisnis	Fact
Manufacturing	Quantity, Cost, Scrapped units
Sales	Quantity, Unit Price, Total Product Cost
Inventory	Quantity, Amount In, Amount Out, Sales Amount Quota
Purchasing	Quantity, Received Amount, Rejected Amount, Stocked Amount
Customer Services	Number of Customer Contacts, Customer Satisfaction Score
Human Resources	Rating performance, Commission
Finance	Revenue, Cost of Sold, Gross Profit, Net profit

BUS MATRIX

Proses bisnis	Dimensi								
	Date	Product	Employee	Customer	Store	Department Group	Vendor	Account	Material
Manufacturing	x	x	x		x		x		x
Sales	x	x	x	x	x				
Inventory	x	x	x				x		
Purchasing	x	x					x		
Customer Services	x	x	x	x					
Human Resources	x		x			x			
Finance	x					x	x	x	

e. Menghasilkan Tabel pada RDBMS (RDBMS nya bebas)

- Kemampuan kalian pada RDBMS yang kalian pilih sangat penting
- FK, PK, Index, Check condition , trigger dll juga dilihat sesuai konteks dari desain



DESAIN & IMPLEMENTASI ETL

a. Data & OLTP Profiling

- Lihat dari data OLTP, lihat hubungan antar table dan struktur lainnya
- Lihat data2 yang ada pada database tersebut, kemungkinannya, nilai null dan anomaly lainnya

Pada proses pembelian (sales) di tabel SalesOrderHeader, apabila pembelian dilakukan online, maka kolom SalesPersonID akan berisi null

b. High Level ETL Design

- Level table ke table , antar OLTP ke Star Schema yang kalian sudah desain

Proses Bisnis	Source (OLTP)	Target (OLAP)
Sales	<ul style="list-style-type: none">- Sales.SalesOrderHeader- Sales.SalesOrderDetail- Production.ProductCostHistory	FactSales
Purchasing	<ul style="list-style-type: none">- Purchasing.ProductVendor- Purchasing.PurchaseOrderDetail- Purchasing.ShipMethod- Purchasing.Vendor	FactPurchasing
Sales & Purchasing	<ul style="list-style-type: none">- Sales.OrderHeader- Purchasing.OrderDetail- Purchasing.OrderHeader- Purchasing.Product.Vendor	DimDate
Sales	<ul style="list-style-type: none">- Sales.SalesTerritory	DimStore
Sales & Purchasing	<ul style="list-style-type: none">- Production.Product- Production.ProductSubCategory- Production.ProductCategory- Production.ProductModel	DimProduct
Sales	<ul style="list-style-type: none">- Person.Person- Person.EmailAddress- Person.PersonPhone- Sales.Customer- Sales.Store	DimCustomer
Sales	<ul style="list-style-type: none">- HumanResources.Employee- Person.Person- Sales.SalesPerson	DimEmployee
Purchasing	<ul style="list-style-type: none">- Purchasing.Vendor	Vendor

c. Source to Target Mapping

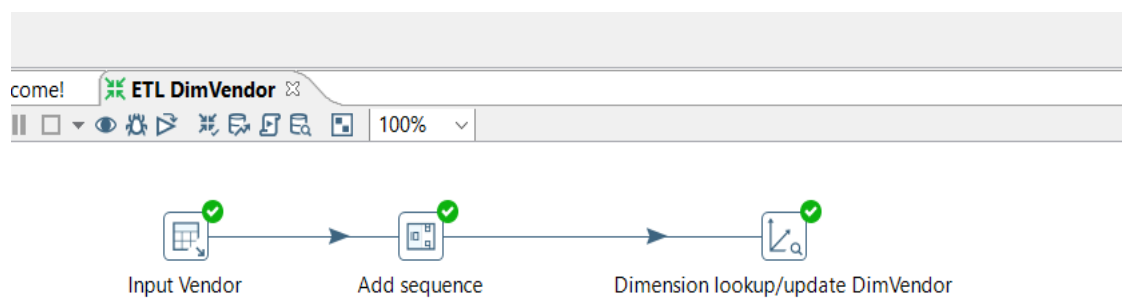
- Level attribute per table tujuan

Source Table	Level Atribut	Target Table
DimProduct	dim_product_id	FactSales
		FactPurchasing
DimDate	dim_date_id	FactSales
		FactPurchasing
DimStore	dim_sales_territory_id	FactSales
DimCustomer	dim_customer_id	FactSales
DimEmpllooye	dim_sales_person_id	FactSales
DimVendor	dim_vendor_id	FactPurchasing

d. ETL one time historical (Teknologi ETL nya bebas)

- Tampilkan pada laporan, step by step proses pembuatannya selain hasil akhirnya
- Kemampuan pada level teknis sangat penting, termasuk comment dll

Proses DimVendor



1. Input Vendor

SELECT

BusinessEntityID

, AccountNumber

, Name

, CreditRating

, PreferredVendorStatus

, ActiveFlag

, ModifiedDate

FROM Purchasing.Vendor

2. Add Sequence

- name of value : dim_vendor_id

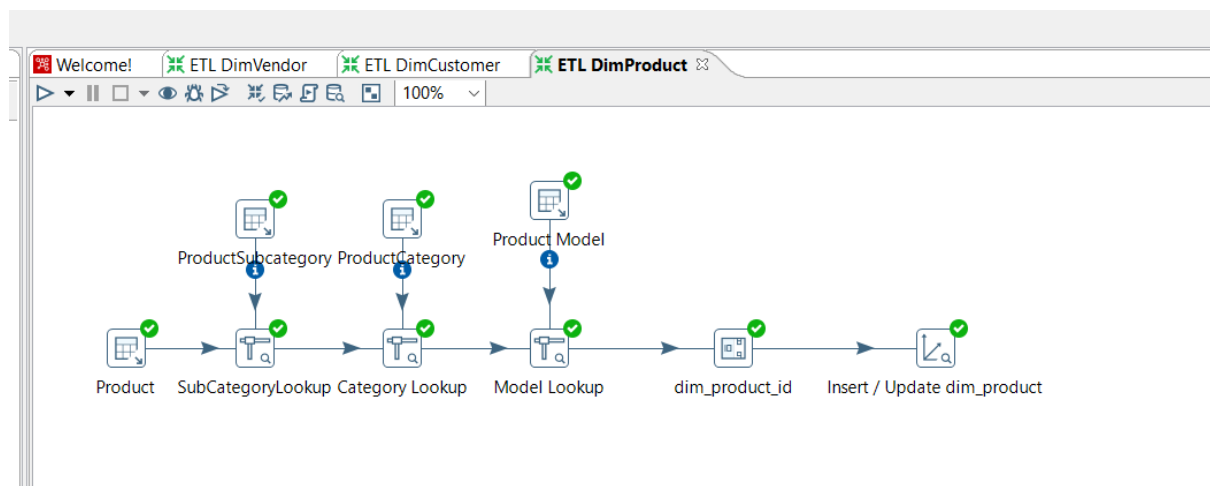
3. DimVendor Lookup

- connection : AdventureWorks2016olap
- target schema : dbo
- target table : DimProduct
- technical key field : dim_vendor_id
- tab KEY, get fields → dimension field : vendor_id, Field : BusinessEntityID
- tab FIELDS, get fields :

Lookup/Update fields

#	Dimension field	Stream field to compare with	Type of dimension update
1	preferred_vendor	PreferredVendorStatus	Insert
2	account_number	AccountNumber	Insert
3	name	Name	Insert
4	credit_rating_id	CreditRating	Insert

Proses DimProduct



1. Input Product

```
SELECT [ProductID]

      ,[Name]

      ,[ProductNumber]

      ,[MakeFlag]

      ,[FinishedGoodsFlag]

      ,[Color]

      ,[SafetyStockLevel]

      ,[ReorderPoint]

      ,[StandardCost]

      ,[ListPrice]

      ,[Size]

      ,[SizeUnitMeasureCode]

      ,[WeightUnitMeasureCode]

      ,[Weight]

      ,[DaysToManufacture]

      ,[ProductLine]

      ,[Class]

      ,[Style]

      ,[ProductSubcategoryID]

      ,[ProductModelID]

FROM [Production].[Product]
```

2. Input Product Subcategory

- step name : ProductSubcategory

```
SELECT [ProductSubcategoryID]

      ,[ProductCategoryID]

      ,[Name]
```

,[rowguid]

,[ModifiedDate]

FROM [Production].[ProductSubcategory]

3. Subcategory Lookup

- Lookup Step : ProductSubcategory
- Get fields → Field : ProductSubcategoryID, LookupField : ProductSubcategoryID
- Get Lookup Fields :

Specify the fields to retrieve :

#	Field	New name	Default	Type	
1	Name	ProductSubCategoryName		String	
2	ProductCategoryID	ProductCategoryID		Integer	

4. InputCategory

- step name : ProductCategory

SELECT [ProductCategoryID]

,[Name]

FROM [Production].[ProductCategory]

5. Category Lookup

- lookup step : Product Category
- Get fields → Field : ProductCategoryID, LookupField : ProductCategoryID
- Get Lookup Fields :

Specify the fields to retrieve :

#	Field	New name	Default	Type	
1	Name	ProductCategoryName		String	

6. Input ProductModel

- step name : Product Model

SELECT [ProductModelID]

,[Name]

FROM [Production].[ProductModel]

7. Model Lookup

- lookup step : Product Model
- Get fields → Field : ProductModelID, LookupField : ProductModelID
- Get Lookup Fields :

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	Name	ProductModelName		String

8. Add sequence

- step name : dim_product_id
- name of value : dim_product_id

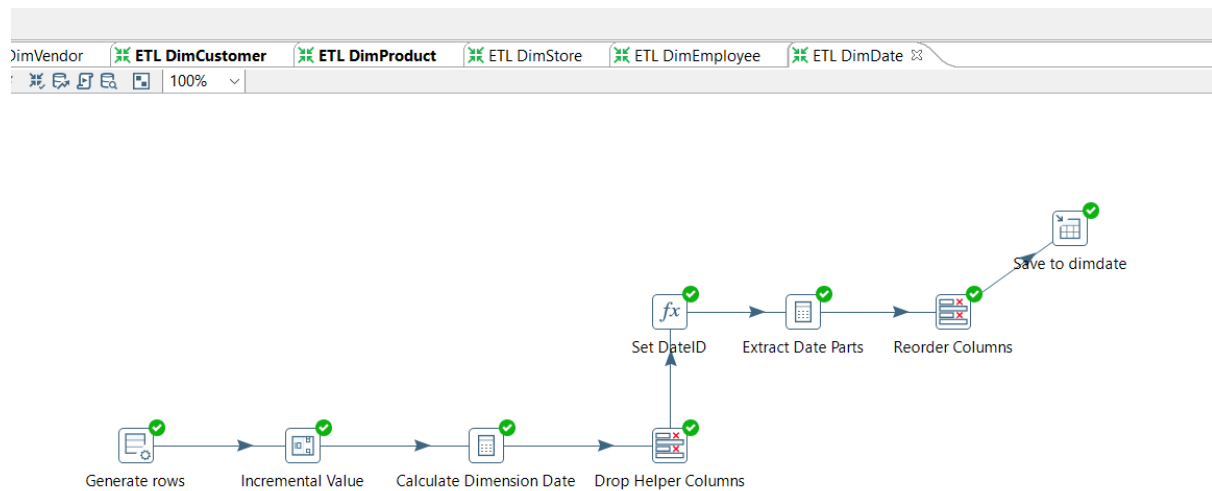
9. DimProduct Lookup

- connection : AdventureWorks2016olap
- target schema : dbo
- target table : DimProduct
- technical key field : dim_product_id
- tab KEY, get fields → dimension field : product_id, Field : productID
- tab FIELDS, get fields :

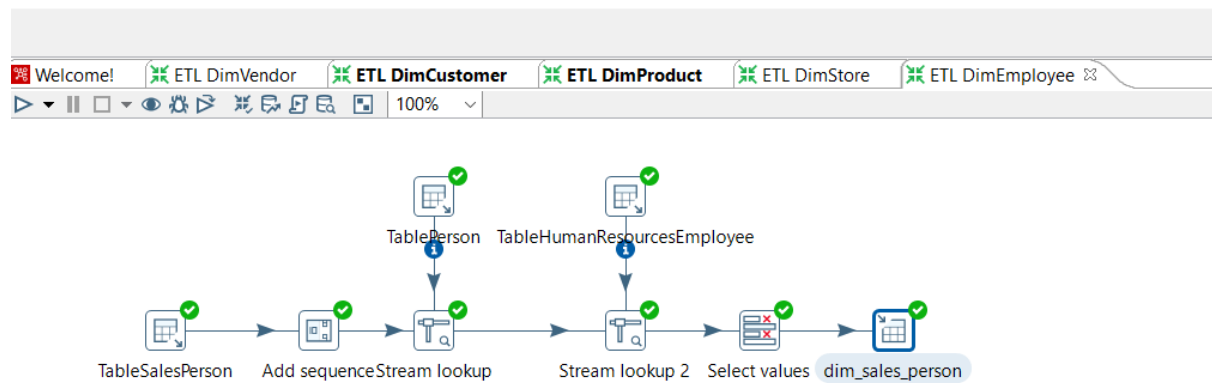
Keys Fields			
Lookup/Update fields			
#	Dimension field	Stream field to compare with	Type of dimension update
1	product_number	ProductNumber	Insert
2	product_name	Name	Insert
3	make_flag	MakeFlag	Insert
4	finished_goods_flag	FinishedGoodsFlag	Insert
5	color	Color	Insert
6	size	Size	Insert
7	size_unit_measure_code	SizeUnitMeasureCode	Insert
8	weight	Weight	Insert
9	weight_unit_measure_code	WeightUnitMeasureCode	Insert
10	product_line_code	ProductLine	Insert
11	style	Style	Insert
12	product_subcategory_id	ProductSubcategoryID	Insert
13	product_subcategory_name	ProductSubCategoryName	Insert
14	product_category_id	ProductCategoryID	Insert
15	product_category_name	ProductCategoryName	Insert
16	product_model_id	ProductModelID	Insert
17	product_model_name	ProductModelName	Insert
18	class_code	Class	Insert

- Execute SQL

Proses DimDate



Proses DimEmployee



1. Input Sales Person
 - SELECT
 - BusinessEntityID
 - FROM Sales.SalesPerson
2. Add Sequence
 - name of value : dim_sales_person_id
3. Input TablePerson

```
SELECT
    BusinessEntityID
, Title
, FirstName
, MiddleName
, LastName
, Suffix
FROM Person.Person
```

4. Stream Table Person LookUP

- Lookup Step : TablePerson
- Get fields → Field : BusinessEntityID, LookupField : BusinessEntityID

5. Input TableHumanResourcesEmployee

```
SELECT
    BusinessEntityID
, NationalIDNumber
, JobTitle
, MaritalStatus
, Gender
, HireDate
FROM HumanResources.Employee
```

6. Stream TabTableHumanResourcesEmployee Lookup

- Lookup Step : TableHumanResourcesEmployee
- Get Fields → Field : BusinessEntityID, LookupField : BusinessEntityID
- Get Lookup Fields →

Field	New name	Default	Type
JobTitle	employee_job_title		None
Gender	employee_gender		None
HireDate	employee_hire_date		None
MaritalStatus	employee_marital_status		None
NationalIDNumber	employee_national_id_number		None

7. Select Values

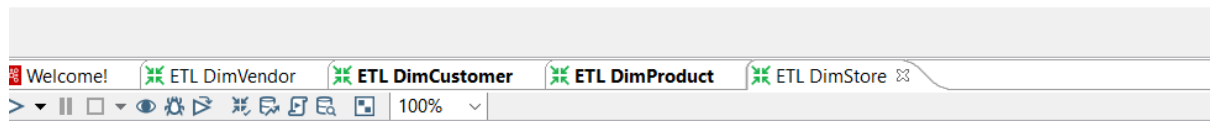
- Get fields →

Fieldname	Rename to	Length
BusinessEntityID	sales_person_entity_id	
person_title		
dim_sales_person_id		
person_first_name		
person_last_name		
person_middle_name		
person_suffix		
employee_job_title		
employee_gender	employee_gender_code	
employee_hire_date		
employee_marital_status	employee_marital_status	
employee_national_id_number		

8. Table DimEmployee Output

- step name : dim_sales_person
- Execute SQL

Proses DimStore



1. Input Sales Territory (dari AdventureWorks2016)

- nama step : Sales Territory

SELECT

TerritoryID

, Name

, CountryRegionCode

, "Group"

, SalesYTD

, SalesLastYear

, CostYTD

, CostLastYear

, rowguid

, ModifiedDate

FROM Sales.SalesTerritory

2. Add Sequence

- Name of value : Sales_territory_ID

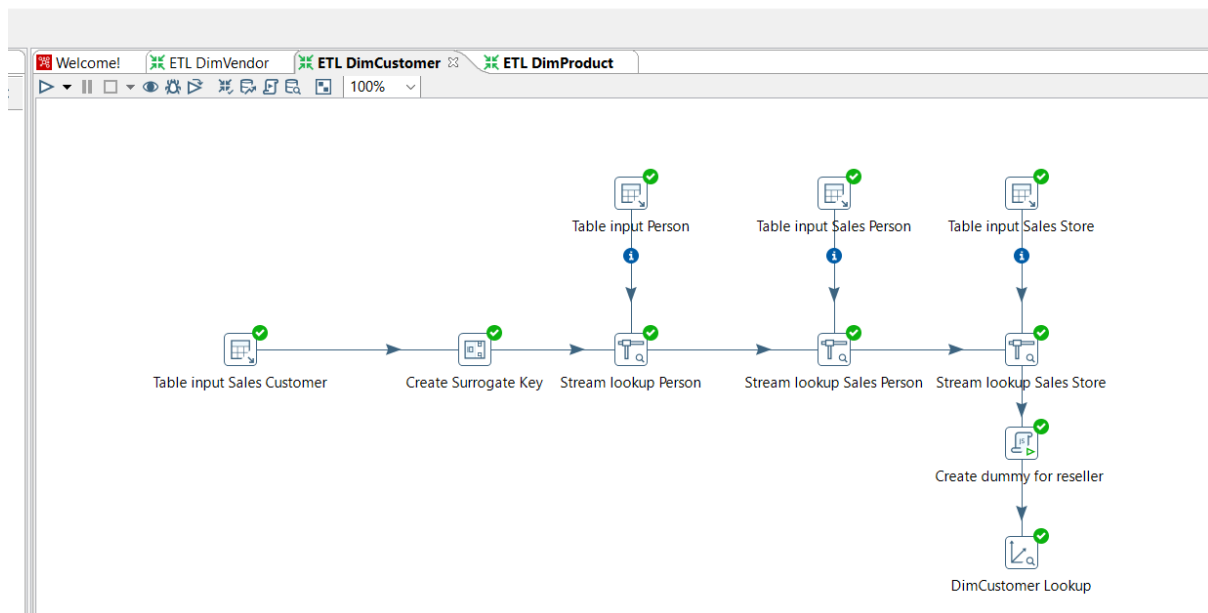
3. DimStore Table Output (Target : AdventureWorks2016olap)

- step name : dim_sales_territory
- target schema : AdventureWorks2016olap
- target table : DimStore
- Database fields :

Main options Database fields			
Fields to insert:			
#	Table field	Stream field	
1	sales_territory_name	Name	
2	country_region_code	CountryRegionCode	
3	territory_id	TerritoryID	
4	dim_sales_territory_id	Sales_territory_ID	
5	group_name	Group	

- Execute SQL

Proses DimCustomer



1. Input table Sales Customer

SELECT

CustomerID

, PersonID

, StoreID

, TerritoryID

FROM Sales.Customer

2. Create Surrogate Key

- name of value : dim_customer_id

3. Input Table Person

```

SELECT

    BusinessEntityID

, Title

, FirstName

, MiddleName

, LastName

, Suffix

FROM Person.Person

```

4. Stream Tables Person Lookup

- Lookup step : Person
- Get fields → Field : PersonID, LookupField : BusinessEntityID
- Get Lookup fields :

#	Field	New name	Default	Type
1	FirstName			None
2	LastName			None
3	MiddleName			None
4	Suffix			None
5	Title			None

5. Input Sales Person

```

SELECT

    BusinessEntityID

, TerritoryID

FROM Sales.SalesPerson

```

6. Stream Sales Person Lookup

- Lookup step : Sales Person
- Get fields → Field : TerritoryID, LookupField : TerritoryID
- Get Lookup fields → Field : BusinessEntityID. Type : None
- Input Sales Person

7. Input Sales Store

```

SELECT

    BusinessEntityID

, Name

```

FROM Sales.Store

8. Stream Sales Store Lookup

- Lookup step : Sales Store
- Get fields → Field : StoreID, LookupField : BusinessEntityID
- Get Lookup fields → Field : Name, Type : None

9. Create dummy for reseller pake Modified javascript value

```
var calculated_is_reseller
if(StoreID != null)
    calculated_is_reseller = 1
else
    calculated_is_reseller = 0;
```

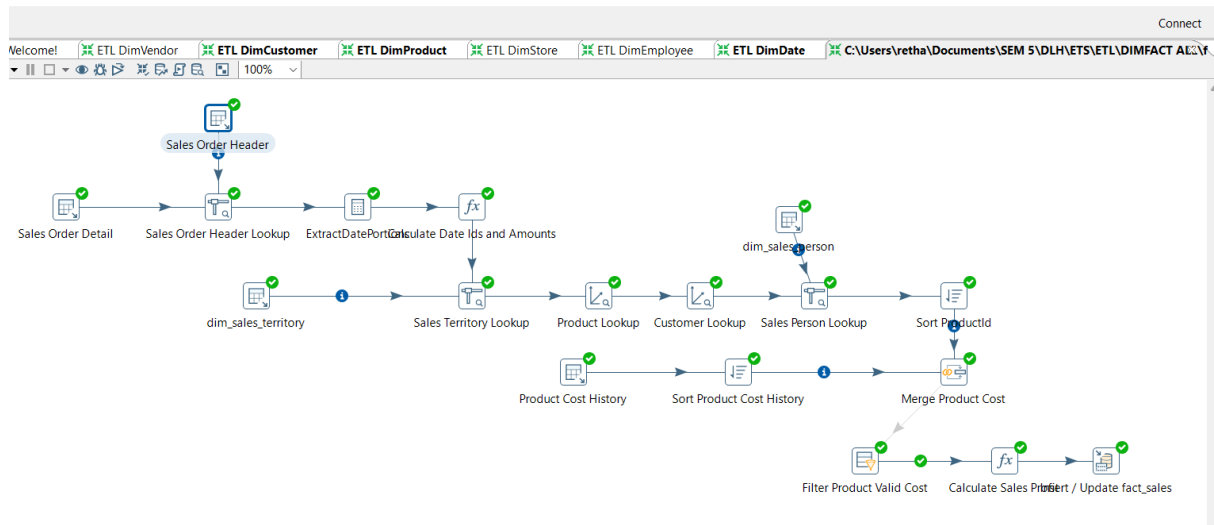
10. DimCustomer Lookup

- connection : AdventureWorks2016olap
- target schema : dbo
- target table : DimCustomer
- technical key field : dim_customer_id
- tab KEY, get fields → dimension field : customer_id, Field : CustomerID
- tab FIELDS, get fields :

Dimension field	Stream field to compare with	Type of dimension update
reseller_store_id	StoreID	Insert
reseller_store_name	Name	Insert
is_reseller	calculated_is_reseller	Insert
person_title	Title	Insert
person_first_name	FirstName	Insert
person_middle_name	MiddleName	Insert
person_last_name	LastName	Insert
person_suffix	Suffix	Insert

- Execute SQL
“Di sini menggunakan SCD type 2”

Proses FactSales



1. Sales Order Detail Input :

SELECT

SalesOrderID

, SalesOrderDetailID

, CarrierTrackingNumber

, OrderQty

, ProductID

, SpecialOfferID

, UnitPrice

, UnitPriceDiscount

, LineTotal

, rowguid

, ModifiedDate

FROM Sales.SalesOrderDetail

where SalesOrderID in

(

43659,

43660,

43661,

43662,

43663,

43664,

43665,

43666,

43667,

43668,

51092

)

2. Sales Order Header Input

SELECT

SalesOrderID

, RevisionNumber

, OrderDate

, DueDate

, ShipDate

, Status

, OnlineOrderFlag

, SalesOrderNumber

, PurchaseOrderNumber

, AccountNumber

, CustomerID

, SalesPersonID

```
, TerritoryID
, BillToAddressID
, ShipToAddressID
, ShipMethodID
, CreditCardID
, CreditCardApprovalCode
, CurrencyRateID
, SubTotal
, TaxAmt
, Freight
, TotalDue
, Comment
, rowguid
, ModifiedDate

FROM Sales.SalesOrderHeader
```

3. Sales Order Header Lookup
 - lookup step : Sales Order Header
 - Get fields dan Get Lookup fields

The key(s) to look up the value(s):

#	Field	LookupField
1	SalesOrderID	SalesOrderID

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	ShipDate			None
2	DueDate			None
3	OrderDate			None
4	SalesOrderNumber			None
5	CustomerID			None
6	Status			None
7	TerritoryID			None
8	SalesPersonID			None

4. Calculator

Step name

ExtractDatePortions

☒ Throw an error on non existing files

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	order_date_year	Year of date A	OrderDate			None			N
2	order_date_month	Month of date A	OrderDate			None			N
3	order_date_day	Day of month of date A	OrderDate			None			N
4	ship_date_year	Year of date A	ShipDate			None			N
5	ship_date_month	Month of date A	ShipDate			None			N
6	ship_date_day	Day of month of date A	ShipDate			None			N

5. Formula

Step name Calculate Date Ids and Amounts

Fields:

#	New field	Formula	Value type	Length
1	calculated_order_date_id	[order_date_year]*10000 + [order_date_month]*100 + [order_date_day]	Integer	
2	calculated_ship_date_id	[ship_date_year]*10000 + [ship_date_month]*100 + [ship_date_day]	Integer	
3	calculated_discount_amount	[UnitPrice] * [UnitPriceDiscount] * [OrderQty]	Number	

6. Input DimStore

SELECT

dim_sales_territory_id

, territory_id

, sales_territory_name

, country_region_code

, group_name

FROM eae_data_management_mmjja.dim_sales_territory

7. Stream lookup Sales Territory

Step name	Sales Territory Lookup			
Lookup step	dim_sales_territory			

The key(s) to look up the value(s):

#	Field	LookupField
1	TerritoryID	territory_id

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	dim_sales_territory_id	calculated_dim_sales_territory_id		Integer

8. Dimension lookup Product

- Target : DimProduct
- KEY :

Keys		Fields	
Key fields (to look up row in dimension):			
#	Dimension field	Field in stream	
1	product_id	ProductID	

Technical key field New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field

Stream Datefield

Date range start field Min. year

Use an alternative start date? ☐

Table date range end Max. year

- FIELDS kosong

9. Dimension Lookup Customer

- target : DimCustomer
- KEYS :

Cache size in rows (U = cache all) 5000

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	customer_id	CustomerID

Technical key field dim_customer_id New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field version_number

Stream Datefield OrderDate

Date range start field start_date Min. year 1900

Use an alternative start date? ☐ <Select Option>

Table date range end end_date Max. year 2199

- FIELDS : kosong

10. Input Sales Person Table

- step : dim_sales_person

SELECT

dim_sales_person_id,

sales_person_entity_id

FROM dim_sales_person

11. Stream Sales Person Lookup

- lookup step : dim_sales_person
- Get Fields & Get lookup fields :

The key(s) to look up the value(s):

#	Field	LookupField
1	SalesPersonID	sales_person_entity_id

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	dim_sales_person_id			None

12. Sort Rows

- get fields :

Sort rows

Step name: Sort ProductId

Sort directory: %%java.io.tmpdir%% [Browse...](#)

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☒

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	ProductID	Y	N	N	0	N

13. Product Cost History Input Table

SELECT

ProductID as ProductCostHistoryProductId,

StartDate,

isnull(EndDate, Convert(DateTime, '2100-12-31',102)) as EndDate,

StandardCost

FROM Production.ProductCostHistory

14. Sort Rows

- get fields :

Sort rows

Step name: Sort Product Cost History

Sort directory: %%java.io.tmpdir%% Browse...

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength
1	ProductCostHistoryProductId	Y	N	N	0

15. Merge Join

Merge join

Step name: Merge Product Cost

First Step: Sort ProductId

Second Step: Sort Product Cost History

Join Type: INNER

Keys for 1st step:

#	Key field
1	ProductId

Get key fields

Keys for 2nd step:

#	Key field
1	ProductCostHistoryProductId

Get key fields

Help OK Cancel

16.

17. Formula

Formula

Step name: Calculate Sales Profit

Fields:

#	New field	Formula	Value type	Length	Precision	Replace value
1	calculated_sales_profit	([UnitPrice] - [StandardCost]) * [OrderQty]	Number			

Help OK Cancel

18. Insert/Update factSales

- target : FactSales

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2		Get fields
1	sales_order_detail_id	=	SalesOrderDetailID			

Update fields:

#	Table field	Stream field	Update		Get update fields
1	sales_profit	calculated_sales_profit	Y		
2	ship_date_datetime	ShipDate	Y		Edit mapping
3	ship_date_id	calculated_ship_date_id	Y		
4	sales_order_number	SalesOrderNumber	Y		
5	sales_order_id	SalesOrderID	N		
6	sales_order_detail_id	SalesOrderDetailID	N		
7	total_discount	calculated_discount_amount	Y		
8	total_amount	LineTotal	Y		
9	product_unit_price_discount	UnitPriceDiscount	Y		
10	product_unit_price	UnitPrice	Y		
11	order_qty	OrderQty	Y		
12	dim_product_id	dim_product_id	Y		
13	dim_sales_person_id	dim_sales_person_id	Y		
14	dim_sales_territory_id	calculated_dim_sales_territory_id	Y		
15	dim_customer_id	dim_customer_id	Y		
16	order_date_id	calculated_order_date_id	Y		
17	order_date_datetime	OrderDate	Y		
18	product_standard_cost_on_day	StandardCost	Y		

Help

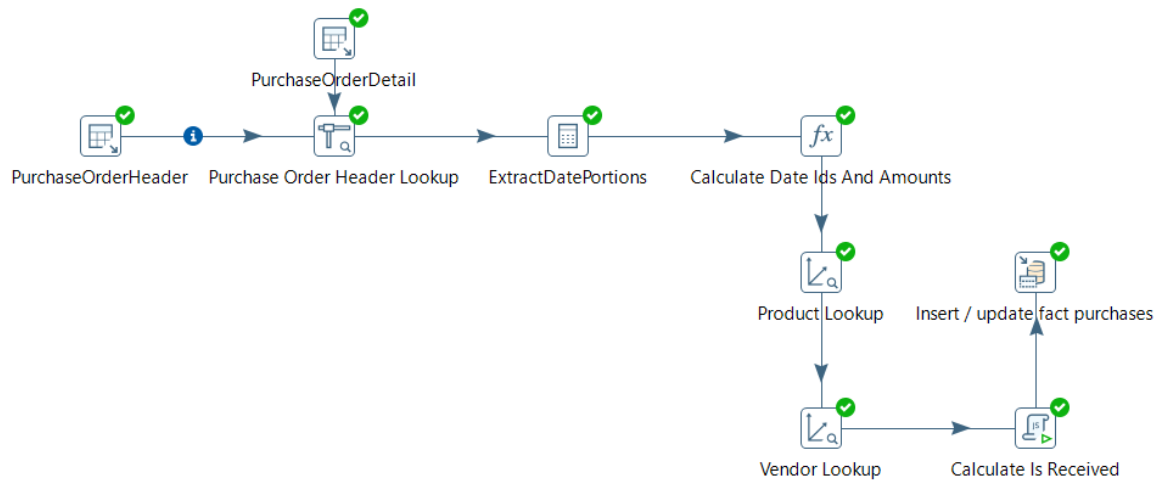
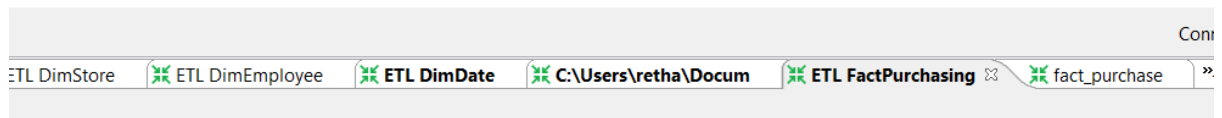
OK

Cancel

SQL

- Execute SQL

Proses FactPurchasing



1. Input Table PurchaseOrderDetail

```
SELECT  
  
PurchaseOrderID,  
  
PurchaseOrderDetailID,  
  
OrderQty,  
  
LineTotal,  
  
ProductID,  
  
UnitPrice  
  
, ReceivedQty  
  
, RejectedQty  
  
, StockedQty  
  
FROM Purchasing.PurchaseOrderDetail
```

2. Input PurchaseOrderHeader

```
SELECT
```

PurchaseOrderID,

RevisionNumber,

Status,

EmployeeID,

VendorID,

ShipMethodID,

OrderDate,

ShipDate,

SubTotal,

TaxAmt,

Freight,

TotalDue

FROM Purchasing.PurchaseOrderHeader

3. Stream Purchase Order Header Lookup

The key(s) to look up the value(s):

#	Field	LookupField
1	PurchaseOrderID	PurchaseOrderID

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	VendorID			None
2	OrderDate			None
3	ShipDate			None
4	Status			None

Preserve memory (costs CPU) ☒

Key and value are exactly one integer field ☐

Use sorted list (i.e.o. hashtable) ☐

4. Calculator

- step name : ExtractDate Portions :

Calculator

Step name: ExtractDatePortions

☒ Throw an error on non existing files

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove	Conversion mask	Decimal symbol	Grouping symbol
1	order_date_year	Year of date A	OrderDate			None			N			
2	order_date_month	Month of date A	OrderDate			None			N			
3	order_date_day	Day of month of date A	OrderDate			None			N			
4	ship_date_year	Year of date A	ShipDate			None			N			
5	ship_date_month	Month of date A	ShipDate			None			N			
6	ship_date_day	Day of month of date A	ShipDate			None			N			

5. Formula

Step name: Calculate Date Ids And Amounts

Fields:

#	New field	Formula	Value type	Length	Precision
1	calculated_order_date_id	[order_date_year]*10000 + [order_date_month]*100 + [order_date_day]	Integer		
2	calculated_ship_date_id	[ship_date_year]*10000 + [ship_date_month]*100 + [ship_date_day]	Integer		
3	calculated_rejected_amount	[RejectedQty] * [UnitPrice]	Number		
4	calculated_stocked_amount	[StockedQty] * [UnitPrice]	Number		
5	calculated_received_amount	[ReceivedQty] * [UnitPrice]	Number		

6. Dimension Product Lookup

- target : DimProduct
- KEYS :

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	product_id	ProductID

Technical key field: dim_product_id New name:

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field: version_number

Stream Datefield: OrderDate

Date range start field: start_date Min. year: 1900

Use an alternative start date? ☐ <Select Option>

Table date range end: end_date Max. year: 2199

- FIELDS : kosong

7. Dimension Vendor Lookup

- target : DimVendor
- KEYS :

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	vendor_id	VendorID

Technical key field New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field

Stream Datefield

Date range start field Min. year

Use an alternative start date? ☐

Table date range end Max. year

- FIELDS : kosong

8. Java Script :

Modified JavaScript value

Step name

Java script functions :

- > Transform Scripts
- > Transform Constants
- > Transform Functions
- > Input fields
 - Getting fields...please
- > Output fields
 - Getting fields...please

Java script :

Script 1

```
var calculated_received_flag = Status == 4;
```

Position: 2, 0

Compatibility mode? ☐ Optimization level

Fields

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	calculated_received_flag		Boolean			N

9. Insert/Update FactPurchasing :

- target : FactPurchasing
- get fields :

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2	
1	purchase_order_detail_id	=	PurchaseOrderDetailID		

Get fields

Update fields:

#	Table field	Stream field	Update	
1	purchase_order_id	PurchaseOrderID	N	
2	purchase_order_detail_id	PurchaseOrderDetailID	N	
3	received_flag	calculated_received_flag	Y	
4	stocked_amount	calculated_stocked_amount	Y	
5	received_amount	calculated_received_amount	Y	
6	rejected_amount	calculated_rejected_amount	Y	
7	stocked_qty	StockedQty	Y	
8	received_qty	ReceivedQty	Y	
9	rejected_qty	RejectedQty	Y	
10	order_amount	LineTotal	Y	
11	order_qty	OrderQty	Y	
12	product_unit_price	UnitPrice	Y	
13	dim_product_id	dim_product_id	Y	
14	dim_vendor_id	dim_vendor_id	Y	
15	ship_date_datetime	ShipDate	Y	
16	ship_date_id	calculated_ship_date_id	Y	
17	order_date_datetime	OrderDate	Y	
18	order_date_id	calculated_order_date_id	Y	

Get update fields

Edit mapping

Help OK Cancel SQL

- Execute SQL

e. ETL incremental (Teknologi ETL nya bebas)

- Tampilkan pada laporan, step by step proses pembuatannya selain hasil akhirnya
- Kemampuan pada level teknis sangat penting, termasuk comment dll

CONTOH PEMANFAATAN DARI DATA WAREHOUSE

a. Menjawab pertanyaan Who, Where, When, What

- Lakukan tapi dengan memperhatikan minimal 2 dimensi

Jawab :

Misalnya, kita ingin tahu siapa saja yang beli sepeda merek X di kota California pada musim kemarau.

Untuk menjawab pertanyaan ini, kita bisa menggunakan data dari tabel DimCustomer, DimProduct, dan FactInternetSales.

- DimCustomer menyediakan informasi tentang pelanggan, seperti nama, alamat, dan kota.
- DimProduct menyediakan informasi tentang produk, seperti merek, kategori, dan harga.
- FactInternetSales menyediakan informasi tentang penjualan, seperti tanggal, jumlah, dan nilai.

Berikut adalah query SQL untuk menjawab pertanyaan tersebut:

SQL

SELECT

c.FirstName,

c.LastName,

c.City,

p.ProductName,

f.OrderDate

FROM

DimCustomer c

JOIN

FactInternetSales f

ON

c.CustomerID = f.CustomerID

JOIN

DimProduct p

ON

f.ProductID = p.ProductID

WHERE

p.ProductName = 'Sepeda X'

AND c.City = 'California'

AND f.OrderDate BETWEEN '2023-06-01' AND '2023-08-31'

Query ini akan menampilkan daftar pelanggan yang membeli sepeda merek X di kota California pada musim kemarau.

b. Menjawab pertanyaan Why

- Misalnya, kenapa kok sepeda saya yang merk X tidak banyak terjual di kota Y pada musim Z.

Jawab :

Misalnya, kita ingin tahu kenapa sepeda merek X milik kita tidak laku di kota California pada musim kemarau.

Untuk menjawab pertanyaan ini, kita bisa menggunakan data dari tabel DimProduct, DimCustomer, FactInternetSales, dan DimSalesTerritory.

- DimProduct menyediakan informasi tentang produk, seperti merek, kategori, dan harga.
- DimCustomer menyediakan informasi tentang pelanggan, seperti nama, alamat, dan kota.
- FactInternetSales menyediakan informasi tentang penjualan, seperti tanggal, jumlah, dan nilai.
- DimSalesTerritory menyediakan informasi tentang wilayah penjualan, seperti kode wilayah, nama wilayah, dan manajer wilayah.

Berikut adalah beberapa faktor yang dapat mempengaruhi penjualan sepeda merek X di kota California pada musim kemarau:

- Harga: Apakah harga sepeda merek X terlalu mahal untuk pelanggan di kota California?
- Promosi: Apakah ada promosi untuk sepeda merek X di kota California pada musim kemarau?
- Musim: Apakah musim kemarau merupakan musim yang tidak tepat untuk penjualan sepeda?
- Kondisi ekonomi: Apakah kondisi ekonomi di kota California sedang buruk pada musim kemarau?
-

Untuk menjawab pertanyaan tersebut, kita bisa menggunakan query SQL untuk menganalisis data dari keempat tabel tersebut.

Misalnya, kita bisa menggunakan query SQL berikut untuk menganalisis faktor harga:

SQL

SELECT

f.SalesAmount,

p.ListPrice

FROM

FactInternetSales f

JOIN

DimProduct p

ON

f.ProductID = p.ProductID

WHERE

p.ProductName = 'Sepeda X'

AND c.City = 'California'

AND f.OrderDate BETWEEN '2023-06-01' AND '2023-08-31'

Query ini akan mengembalikan daftar nilai penjualan dan harga sepeda merek X di kota California pada musim kemarau.

Dengan menganalisis data ini, kita bisa menentukan apakah harga sepeda merek X terlalu mahal untuk pelanggan di kota California.

Berikut adalah beberapa contoh visualisasi yang bisa digunakan untuk menjawab pertanyaan di atas:

- Diagram batang: Diagram batang dapat digunakan untuk membandingkan penjualan sepeda merek X di kota California pada musim kemarau dengan penjualan di kota lain atau musim lain.
- Diagram pie: Diagram pie dapat digunakan untuk menunjukkan persentase penjualan sepeda merek X di kota California berdasarkan faktor-faktor tertentu, seperti harga, promosi, musim, atau kondisi ekonomi.