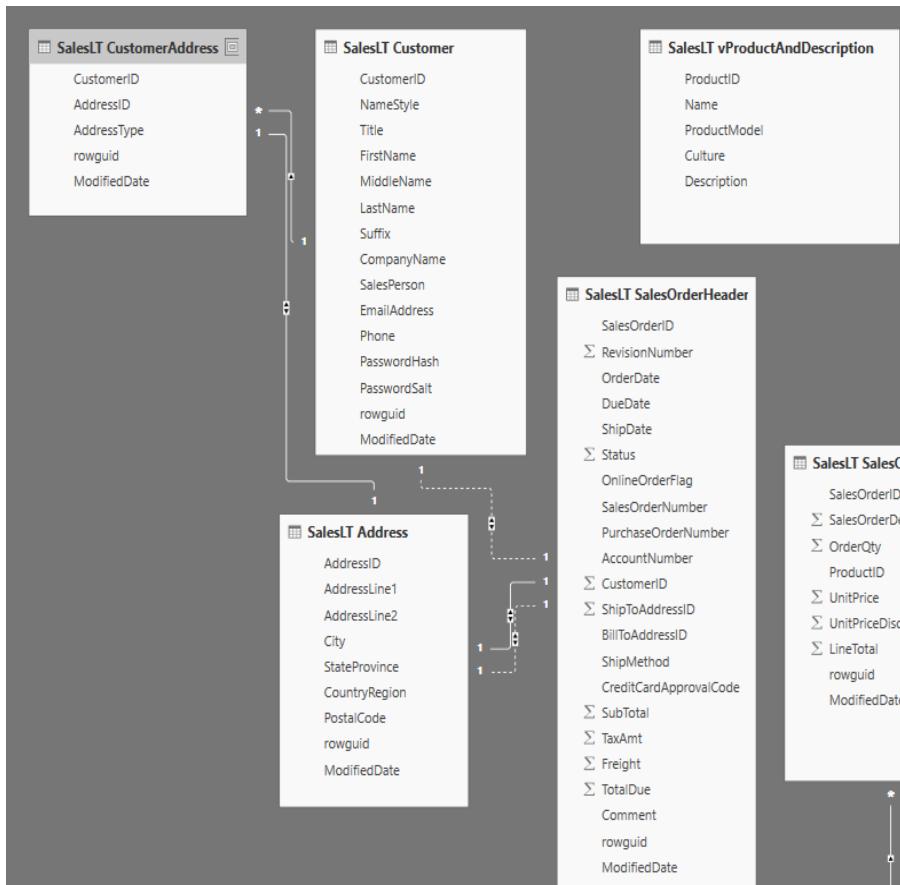


Data Model



These are some critical Sql queries which impact business rules and processes:

- BI Queries & results (SQL queries is fine, can use BI tools but NOT graded on BI)
 - Ranked order of Vendors by purchase amount \$

SQL Queries

```
select RANK() over
(order by
SUM(FP.TotalDue)
desc) as Ranking,
DV.VendorName as
'Vendor Name',
CONCAT('$',SUM(FP.Tot
alDue)) as 'Purchase
Amount'
from FactPurchases as
FP
Inner join DimVendors
as DV
```

```

on FP.VendorID = DV.BusinessEntityID
group by DV.VendorName;



- o Ranked order of products by amount $
  - By category



```

select RANK() over (order by SUM(FP.LineTotal)
desc) as Ranking, DP.ProductCategoryName as
'Product Category', CONCAT('$',SUM(FP.LineTotal))
as 'Purchase Amount'
from FactPurchases as FP
inner join DimProducts_Purchased as DP
on FP.ProductPurchasedSK = DP.ProductPurchasedSK
group by DP.ProductCategoryName;

```



- By subcategory



```

select RANK() over (order by SUM(FP.LineTotal)
desc) as Ranking, DP.ProductSubcategoryName as
'Product Subcategory',
CONCAT('$',SUM(FP.LineTotal)) as 'Purchase Amount'
from FactPurchases as FP
inner join DimProducts_Purchased as DP
on FP.ProductPurchasedSK = DP.ProductPurchasedSK
group by DP.ProductSubcategoryName;

```



- By product model (top 20)



```

select Top 20 RANK() over (order by
SUM(FP.LineTotal) desc) as Ranking, DP.ModelName as
'Product Model', CONCAT('$',SUM(FP.LineTotal)) as
'Purchase Amount'
from FactPurchases as FP
inner join DimProducts_Purchased as DP
on FP.ProductPurchasedSK = DP.ProductPurchasedSK
group by DP.ModelName;

```



- By product (top 20)



```

select Top 20 RANK() over (order by
SUM(FP.LineTotal) desc) as Ranking, DP.ProductName
as 'Product Name', CONCAT('$',SUM(FP.LineTotal)) as
'Purchase Amount'
from FactPurchases as FP
inner join DimProducts_Purchased as DP
on FP.ProductPurchasedSK = DP.ProductPurchasedSK
group by DP.ProductName;

```



- o Ranked list of employees purchasing products
amount $



```

select RANK() over (order by SUM(FP.TotalDue) desc)
as Ranking, (DE.FirstName + ' ' + DE.MiddleName + ' ' +
DE.LastName)as
'Employee Full
Name',
CONCAT('$',SUM(FP.
TotalDue)) as
'Purchase Amount'
from FactPurchases
as FP
Inner join
DimEmployee as DE
on FP.EmployeeID
=
DE.BusinessEntityI
D
group by
DE.FirstName,
DE.MiddleName,
DE.LastName;

```



- o List of employees who purchase d products with phone, email & address



```

select
(DE.FirstName +
'+ DE.MiddleName +
'+ DE.LastName)as
'Employee Full
Name',
DE.Phone,
DE.EmailAddress,
DE.AddressLine1,
DE.AddressLine2,
DG.City,
DG.StateProvinceNa
me as 'State',
DG.PostalCode
from FactPurchases
as FP
Inner join
DimEmployee as DE
on FP.EmployeeID
=
DE.BusinessEntityI
D
Inner join
DimGeography as DG
on DE.GeoSK =
DG.GeoSK;

```


```

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

```
Inner join
DimGeography as DG
on DV.GeoSK =
DG.GeoSK;
```

```
select (DVC.FirstName + ' ' + DVC.MiddleName+' '+
DVC.LastName)as 'Vendor Full Name', DVC.PhoneNumber,
DVC.EmailAddress, DV.AddressLine1, DV.AddressLine2,
DG.City, DG.StateProvinceName as 'State', DG.PostalCode
from FactPurchases as FP
Inner join DimVendorContacts as DVC
on FP.VendorID = DVC.Vendor_BusinessEntityID
Inner join DimVendors as DV
on DVC.VendorSK = DV.VendorSK
```

Microsoft Azure

```
select sum (TotalDue) AS TotalSales from SalesLT.salesOrderHeader;
```

- Input tables: SalesLT.SalesOrderHeader
- Input columns: TotalDue
- Output columns: Sum(TotalDue) AS ‘Total Sales’
- Result rowcount: 1

The screenshot shows the Microsoft Azure Data Explorer interface. The left sidebar lists various Azure services like App Services, Function Apps, and SQL databases. The main area is titled "MyAdventureSampleDatabase - Data explorer (preview)" and shows a "Data explorer (preview)" section with a tree view of tables, views, and stored procedures. A query editor window titled "Query 1" contains the following T-SQL code:

```
1 select * from SalesLT.salesOrderHeader;
2 select sum (TotalDue) AS TotalSales from SalesLT.salesOrderHeader;
3
```

The results pane below shows the output of the second query:

TOTALSALES
956303.5949

A status bar at the bottom indicates "Query succeeded | 1s".

a2. Using SalesOrderDetail

```
select sum(LineTotal) AS TotalSales from SalesLT.salesOrderDetail;
```

- Input tables: SalesLT.SalesOrderDetail
- Input columns: LineTotal
- Output columns: Sum(LineTotal) AS 'Total Sales'
- Result rowcount: 1

The screenshot shows the Microsoft Azure portal interface with the 'Data explorer (preview)' section selected. The left sidebar lists various Azure services like Storage accounts, Virtual machines, and SQL databases. The main area displays a query editor window titled 'MyAdventureSampleDatabase - Data explorer (preview)'. The query is:

```
1 select * from SalesLT.salesOrderHeader;
2
3 select sum(TotalDue) AS TotalSales from SalesLT.salesOrderHeader;
4 select * From SalesLT.salesOrderDetail;
5 select sum(LineTotal) AS TotalSales from SalesLT.salesOrderDetail;
6 select sum(TotalDue) AS TotalSales, a.CountryRegion from SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
7 select sum(TotalDue) AS TotalSales, a.CountryRegion,a.City from SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
8 select sum(TotalDue) AS TotalSales, c.CustomerID from SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON h.CustomerID=c.CustomerID group by
9
```

The results pane shows the output of the query:

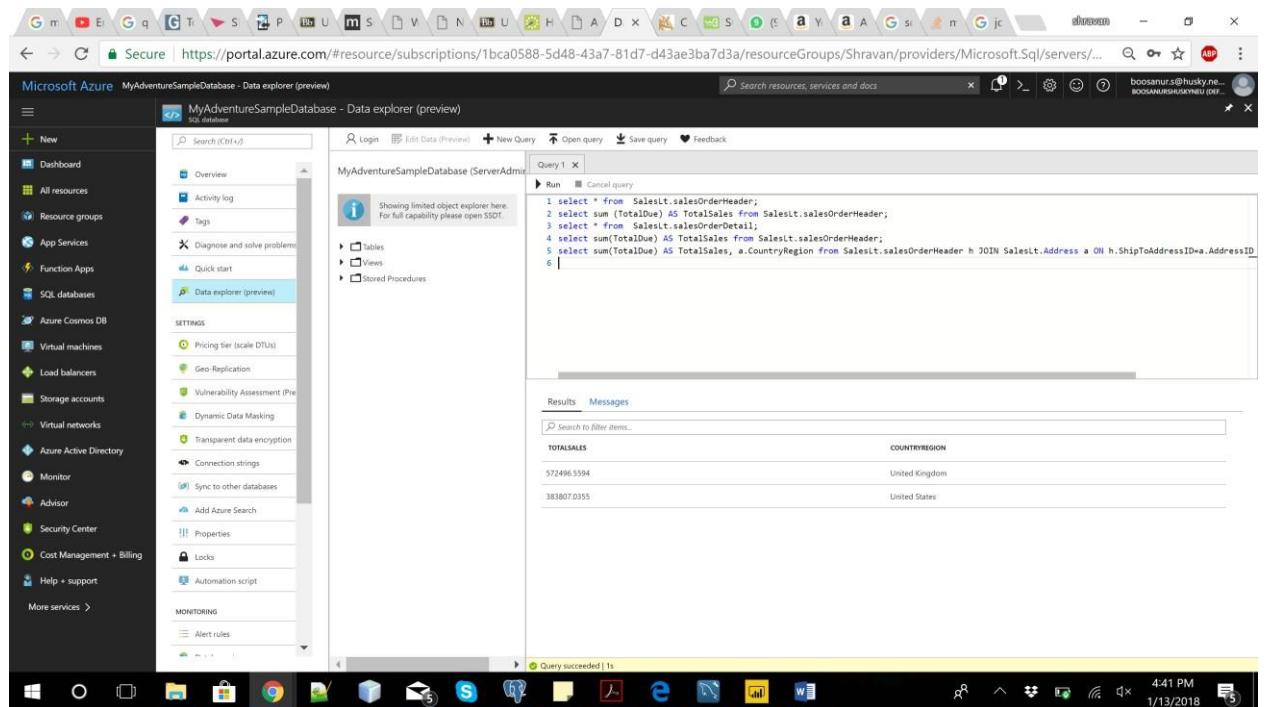
TOTALSALES
708690.153058

The status bar at the bottom indicates 'Query succeeded | 1s'.

- b) Total sales by country – ranked/sorted (highest to lowest)

```
select sum(TotalDue) AS TotalSales, a.CountryRegion from
SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON
h.ShipToAddressID=a.AddressID group by CountryRegion order by TotalSales
DESC;
```

- Input tables: SalesLT.SalesOrderHeader, SalesLT.Address
- Input columns: From table SalesOrderHeader - Total due,
- From table SalesOrderDetail - CountryRegion
- Output columns: CountryRegion, Sum(TotalDue) AS ‘Total Sales’
- Result rowcount: 2



The screenshot shows the Microsoft Azure portal interface with the Data Explorer preview for the 'MyAdventureSampleDatabase'. The left sidebar lists various Azure services like Dashboard, Resource groups, App Services, Function Apps, SQL databases, and more. The main area shows the database structure with 'Tables', 'Views', and 'Stored Procedures' listed under 'MyAdventureSampleDatabase (ServerAdmin)'. A query window titled 'Query 1' contains the T-SQL code provided above. The results tab shows the output of the query:

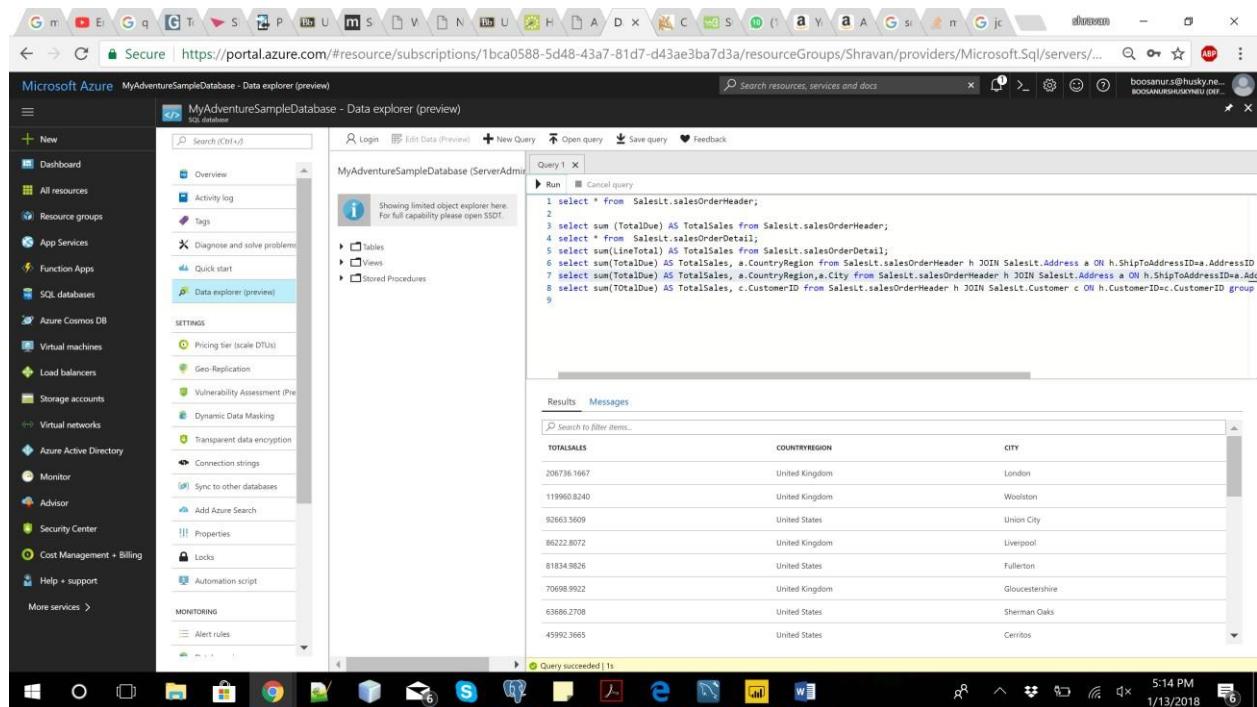
TOTALSALES	COUNTRYREGION
572465594	United Kingdom
3838070355	United States

The status bar at the bottom indicates 'Query succeeded | 1s'.

- c) Total sales by city & country – ranked/sorted (highest to lowest)

```
select sum(TotalDue) AS TotalSales, a.CountryRegion,a.City from
SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON
h.ShipToAddressID=a.AddressID group by CountryRegion,City order by
TotalSales DESC;
```

- Input tables: SalesLT.SalesOrderHeader, SalesLT.Address
- Input columns: From table SalesOrderHeader - Total due,
- From table Address - CountryRegion, City
- Output columns: Sum(TotalDue) AS ‘Total Sales’,CountryRegion, City,
- Result rowcount: 29



The screenshot shows the Microsoft Azure portal with the Data Explorer preview for the MyAdventureSampleDatabase. The left sidebar lists various Azure services like Resource groups, App Services, and SQL databases. The main area shows the query results for the provided T-SQL script. The results table has three columns: TOTALSALES, COUNTRYREGION, and CITY. The data is sorted by TOTALSALES in descending order.

TOTALSALES	COUNTRYREGION	CITY
206736.1667	United Kingdom	London
119960.8240	United Kingdom	Woolston
92063.5609	United States	Union City
86222.8072	United Kingdom	Liverpool
81834.9826	United States	Fullerton
70698.9922	United Kingdom	Gloucestershire
63886.2708	United States	Sherman Oaks
45992.3685	United States	Cerritos

- d) Total sales by customer (person) – ranked/sorted (highest to lowest)

```
select sum(TotalDue) AS TotalSales, c.CustomerID from
SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON
h.CustomerID=c.CustomerID group by c.CustomerID order by TotalSales DESC;
```

- Input tables: SalesLT.SalesOrderHeader, SalesLT.Customer
- Input columns: From table SalesOrderHeader - Total due,
- From table Customer - CustomerID,
- Output columns: ‘Total Sales’, CustomerID,
- Result rowcount: 32

The screenshot shows the Microsoft Azure portal interface with the Data Explorer preview open. The left sidebar lists various Azure services like Storage accounts, Virtual machines, and Active Directory. The main area displays a query window with the following content:

```
Query 1 | X
Run Cancel query
1 select * from SalesLT.salesOrderHeader;
2
3 select sum (TotalDue) AS TotalSales from SalesLT.salesOrderHeader;
4 select sum(TotalSales) AS TotalSales from SalesLT.salesOrderDetail;
5 select sum(TotalDue) AS TotalSales, a.CountryRegion from SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
6 select sum(TotalDue) AS TotalSales, a.CountryRegion,a.City from SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
7 select sum(TotalDue) AS TotalSales, c.CustomerID from SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON h.CustomerID=c.CustomerID group
8
9
```

The Results tab shows the output of the query:

TOTALSALES	CUSTOMERID
119960.8240	29736
108597.9356	30050
98138.2131	29546
92661.5609	29957
86222.8072	29796
81854.9826	29929
70698.9922	29932
63688.2708	29660

At the bottom of the screenshot, the taskbar shows the date and time as 5:11 PM on 1/13/2018.

- e) Total sales by customer (company) – ranked/sorted (highest to lowest)

```
select sum(TotalDue) AS TotalSales, c.CompanyName from
SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON
h.CustomerID=c.CustomerID group by c.CompanyName order by TotalSales DESC;
```

- Input tables: SalesLT.SalesOrderHeader, SalesLT.Customer
- Input columns: From table SalesOrderHeader - Total due,
- From table Customer - CompanyName
- Output columns: CompanyName, ‘Total Sales’
- Result rowcount: 32

The screenshot shows the Microsoft Azure portal interface with the 'Data explorer (preview)' section open. The left sidebar lists various Azure services like Storage accounts, Virtual machines, and SQL databases. The main area displays a query editor with the following SQL code:

```
1 select * from SalesLT.salesOrderHeader;
2
3 select sum (TotalDue) AS TotalSales from SalesLT.salesOrderHeader;
4 select * from SalesLT.salesOrderDetail;
5 select sum(LineTotal) AS TotalSales from SalesLT.salesOrderDetail;
6 select sum(TotalDue) AS TotalSales, a.CountryRegion from SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
7 select sum(TotalDue) AS TotalSales, a.CountryRegion,a.City from SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
8 select sum(TotalDue) AS TotalSales, c.CustomerID from SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON h.CustomerID=c.CustomerID group
9 select sum(TotalDue) AS TotalSales, c.CompanyName from SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON h.CustomerID=c.CustomerID group
10
```

The results pane shows the output of the query:

TOTALSALES	COMPANYNAME
119960.8240	Action Bicycle Specialists
108597.9536	Metropolitan Bicycle Supply
98138.2131	Bulk Discount Store
92665.5609	Eastside Department Store
86222.8072	Riding Cycles
81834.9826	Many Bikes Store
70696.9922	Instruments and Parts Company
63686.2708	Extreme Riding Supplies

At the bottom of the results pane, it says 'Query succeeded | 1s'.

- f) Sales by product category – ranked/sorted (highest to lowest)

```
select sum(totaldue) as total,pc.ProductCategoryID,pc.Name from
SalesLT.Product p join SalesLT.ProductCategory pc on
p.ProductCategoryID=pc.ProductCategoryID join SalesLT.SalesOrderDetail sod
on sod.ProductID=p.ProductID join SalesLT.SalesOrderHeader soh on
soh.SalesOrderID=sod.SalesOrderID group by pc.ProductCategoryID,pc.Name
order by total desc;
```

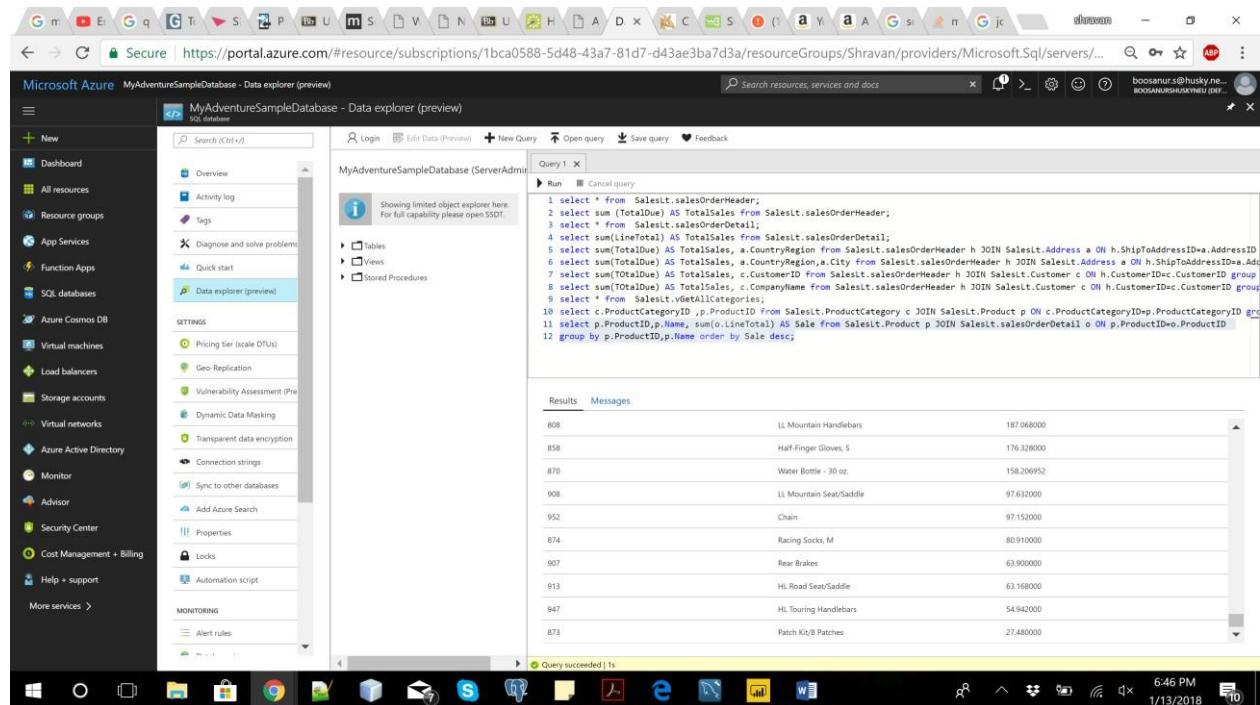
- Input tables: SalesOrderDetail, salesLT.Product, salesLT.ProductCategory.
- Input columns: From table SalesOrderDetail - LineTotal,
- From table ProductCategory – ProductCategoryID, Name
- Output columns: ‘Total’, ProductCategoryID, Name
- Result rowcount: 26

TOTAL	PRODUCTCATEGORYID	NAME
4951047.8822	7	Touring Bikes
4307066.3148	5	Mountain Bikes
3860090.9106	6	Road Bikes
3020346.9345	25	Jerseys
2981257.3917	16	Mountain Frames
1680020.8122	35	Helmets
1678628.6952	17	Pedals
1545066.0863	18	Road Frames

g) Sales by product name – ranked/sorted (highest to lowest)

```
select p.ProductID,p.Name, sum(o.LineTotal) AS Sale from SalesLT.Product p
JOIN SalesLT.salesOrderDetail o ON p.ProductID=o.ProductID
group by p.ProductID,p.Name order by Sale desc;
```

- Input tables: SalesOrderDetail, salesLT.Product
- Input columns: From table SalesOrderDetail - LineTotal,
- From table Product- ProductID, Name
- Output columns: ProductID, Name, ‘Total’
- Result rowcount: 142



The screenshot shows the Microsoft Azure portal interface with the Data Explorer preview for the 'MyAdventureSampleDatabase'. The left sidebar lists various Azure services like App Services, Function Apps, and Storage accounts. The main area shows the database structure with 'Tables' and 'Views' expanded. A query editor window is open with the following SQL code:

```
1 select * from SalesLT.product
2 select sum(TotalDue) AS TotalSales from SalesLT.salesOrderHeader;
3 select * from SalesLT.salesOrderDetail;
4 select sum(LineTotal) AS TotalSales; a.CountryRegion From SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
5 select sum(TotalDue) AS TotalSales, a.CountryRegion From SalesLT.salesOrderHeader h JOIN SalesLT.Address a ON h.ShipToAddressID=a.AddressID
6 select sum(TotalDue) AS TotalSales, a.CustomerID From SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON h.CustomerID=c.CustomerID group
7 select sum(TotalDue) AS TotalSales, c.CompanyName From SalesLT.salesOrderHeader h JOIN SalesLT.Customer c ON h.CustomerID=c.CustomerID group
8 select * from SalesLT.viewAllCategories;
9 select c.ProductCategoryID ,p.ProductID From SalesLT.ProductCategory c JOIN SalesLT.Product p ON c.ProductCategoryID=p.ProductCategoryID
10 select p.ProductID,p.Name, sum(o.LineTotal) AS Sale from SalesLT.Product p JOIN SalesLT.salesOrderDetail o ON p.ProductID=o.ProductID
11 select p.ProductID,p.Name, sum(o.LineTotal) AS Sale from SalesLT.Product p JOIN SalesLT.salesOrderDetail o ON p.ProductID=o.ProductID
12 group by p.ProductID,p.Name order by Sale desc;
```

The results pane displays a table with 142 rows, showing the total sales for each product. The columns are ProductID, Name, and Sale. The top few rows are:

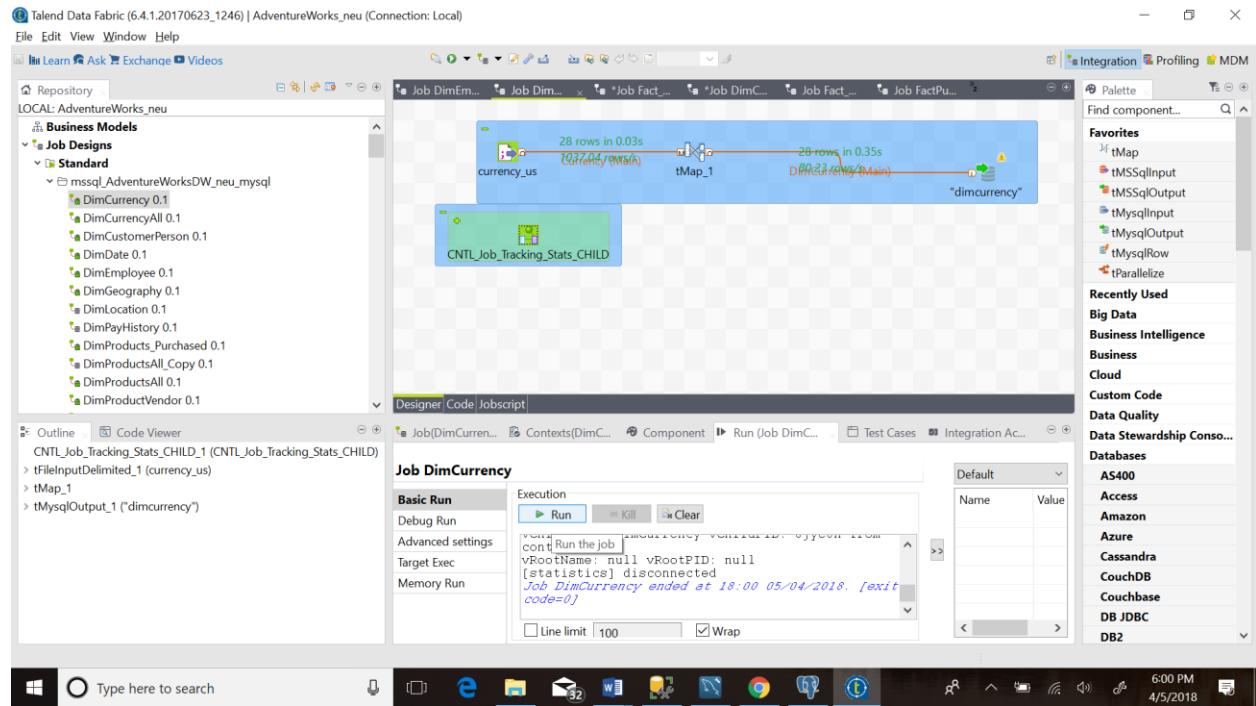
ProductID	Name	Sale
608	LL Mountain Handlebars	187.068000
858	Half-Finger Gloves, S	176.328000
870	Water Bottle - 30 oz.	158.209552
908	LL Mountain Seat/Saddle	97.632000
952	Chain	97.152000
874	Racing Socks, M	80.910000
907	Rear Brakes	63.900000
913	HL Road Seat/Saddle	63.168000
947	HL Touring Handlebars	54.942000
873	Patch Kit/B Patches	27.480000

```
*****
*****
*****
*
```

Data Integration

These are talend jobs for Extract, Transform and Load in Data Warehouse:

DimCurrency



MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

adventureworksdw_neu

Tables

- dimcurrency
- dimcustomerperson
- dimdate
- dimemployee
- dimgeography
- dimlocation
- dimpayhistory
- dimproducts_purchased
- dimproductsall
- dimproductvendor
- dimpromotion
- dimsalesterritory
- dimscreaperson
- dimshipmethod
- dimstore
- dimvendorcontacts
- dimvendors
- fact_workorder
- fact_workorder_rejects

Management Schemas

Information

Table: dimemployee

Columns:

EmployeeSK	int(11) AI PK
BusinessEntityID	int(11)
EmployeeNationalID	varchar(15)
ManagerEmployeeSK	int(11)
ManagerBusinessEntityID	int(11)

Object Info Session

Result Grid

Result 27 x

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
82	20:42:11	SELECT count(*) FROM adventureworksdw_neu.dimfactpurchases	1 row(s) returned	0.015 sec / 0.000 sec
83	20:42:11	SELECT count(*) FROM adventureworksdw_neu.dimfactpurchases_rejects	1 row(s) returned	0.000 sec / 0.000 sec
84	20:43:14	SELECT count(*) as dimcurrency FROM adventureworksdw_neu.dimcurren...	1 row(s) returned	0.000 sec / 0.000 sec

Result Grid

Form Editor

Context Help Snippets

8:43 PM 4/5/2018

Row count: 28

DimDate

Talend Data Fabric (6.4.1.20170623_1246) | AdventureWorks_neu (Connection: Local)

File Edit View Window Help

Learn Ask Exchange Videos

Job DimDate ... Job DimProduct... Job DimEmplo... *Job Fact_Wor... *Job DimCurre... Job Fact_Wor... Job FactPurch...

Integration Profiling MDM

Repository LOCAL: AdventureWorks_neu

Business Models

Job Designs

Standard

mssql_AdventureWorksDW_neu_mysql

- DimCurrency 0.1
- DimCurrencyAll 0.1
- DimCustomerPerson_copy 0.1
- DimDate 0.1
- DimEmployee 0.1
- DimGeography 0.1
- DimLocation 0.1
- DimPayHistory 0.1
- DimProducts_Purchased 0.1
- DimProductsAll_Copy 0.1
- DimProductsAll 0.1
- DimProductVendor 0.1

DimDate

3652 rows in 0.58s 6296.53 MB/s tMap_1 3652 rows in 1.25s 2900.96 MB/s "dimdate"

Designer Code Jobscrip[

Outline Code Viewer

Job DimDate

Basic Run

Execution

Run Kill Clear

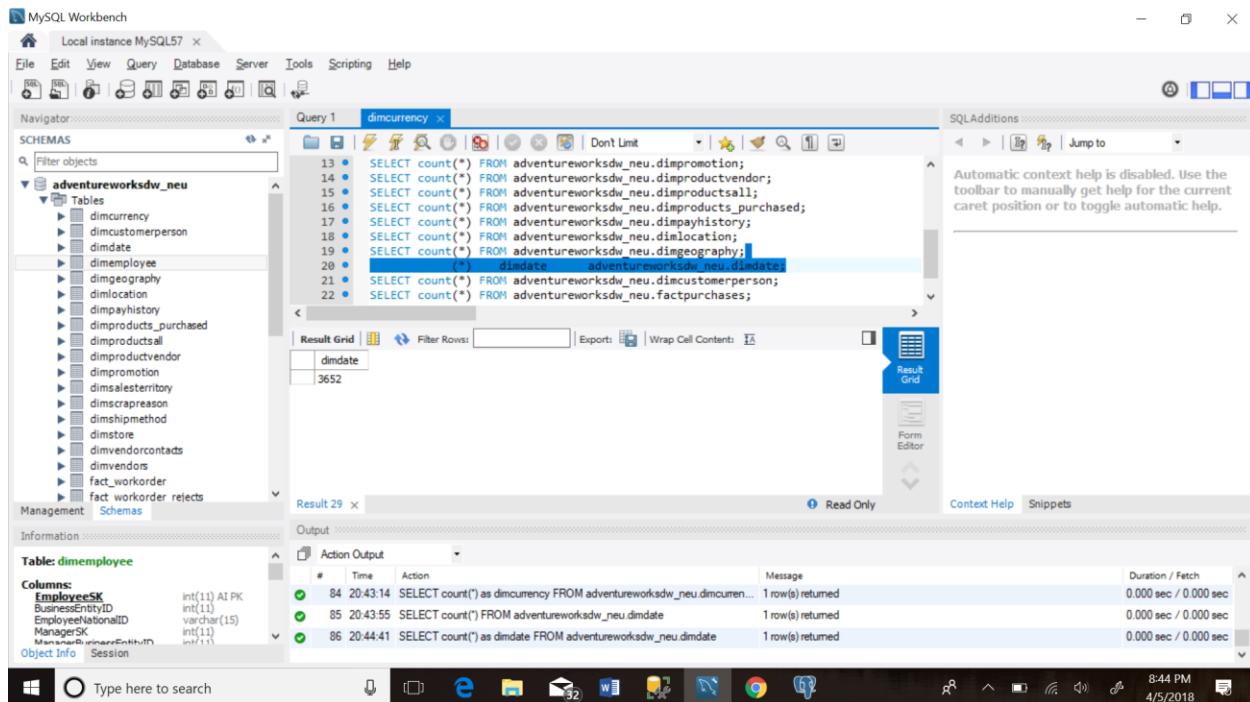
Starting job DimDate at 06:45 05/04/2018.

[statistics] connecting to socket on port 4069
[statistics] connected
[statistics] disconnected
Job DimDate ended at 06:45 05/04/2018. [exit code=0]

Default

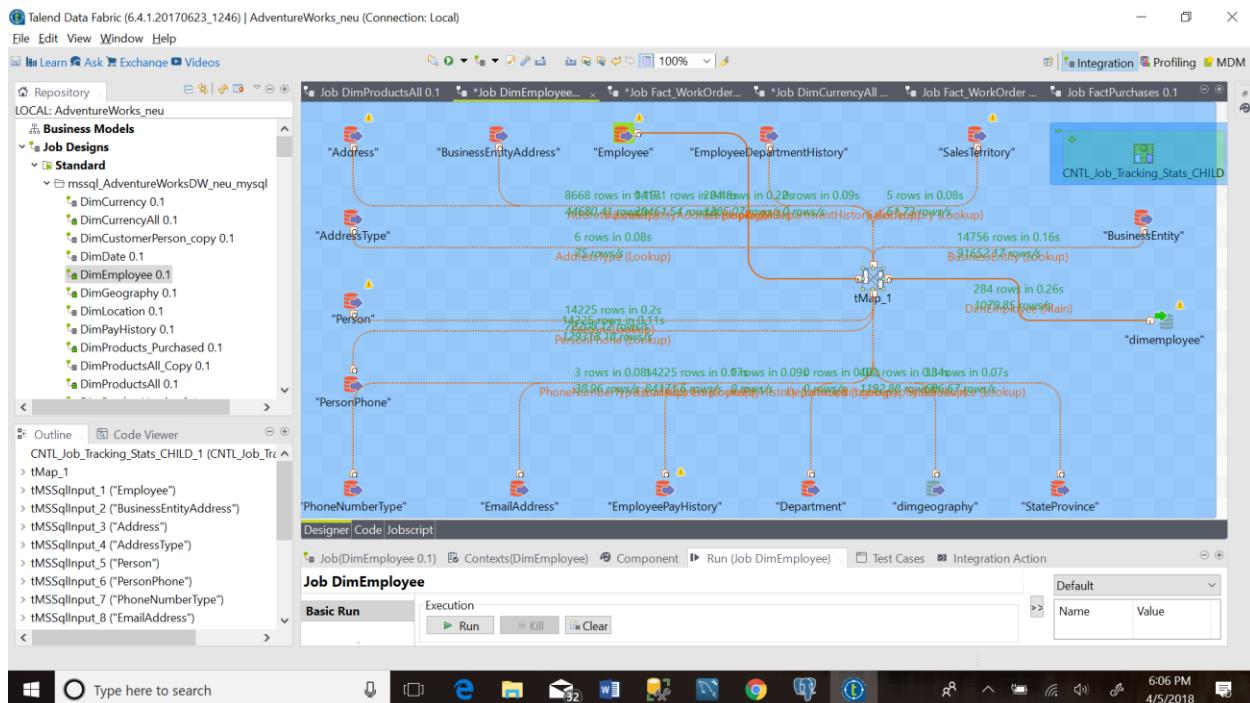
Name Value

6:03 PM 4/5/2018



Row Count 3652

DimEmployee



MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1 dimcurrency

```

1 SELECT count(*) as dimCurrency FROM adventureworksdw_neu.dimCurrency;
2
3 SELECT count(*) FROM adventureworksdw_neu.fact_workorderrouting;
4
5 SELECT count(*) FROM adventureworksdw_neu.fact_workorder_rejects;
6
7 SELECT count(*) FROM adventureworksdw_neu.dimCustomerPerson;
8
9 SELECT count(*) FROM adventureworksdw_neu.dimVendor;
10
11 SELECT count(*) FROM adventureworksdw_neu.dimVendorContacts;
12
13 SELECT count(*) FROM adventureworksdw_neu.dimStore;
14
15 SELECT count(*) FROM adventureworksdw_neu.dimShipMethod;

```

Result Grid | Filter Rows: Export | Wrap Cell Content: Result Grid

Result 31 | Read Only | Context Help | Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
86	20:44:41	SELECT count(*) as dimdate FROM adventureworksdw_neu.dimdate	1 row(s) returned	0.000 sec / 0.000 sec
87	20:45:19	SELECT count(*) FROM adventureworksdw_neu.dimEmployee	1 row(s) returned	0.000 sec / 0.000 sec
88	20:45:31	SELECT count(*) as dimEmployee FROM adventureworksdw_neu.dimempl...	1 row(s) returned	0.000 sec / 0.000 sec

Information

Table: dimemployee

Columns:

- EmployeeSK int(11) AI PK
- BusinessEntityID int(11)
- EmployeeNationalID varchar(15)
- Manager int(11)
- ManagerEmployeeSK int(11)

Object Info Session

Type here to search

8:45 PM 4/5/2018

Row Count 284

DimGeography

Talend Data Fabric (6.4.1.20170623_1246) | AdventureWorks_neu (Connection: Local)

File Edit View Window Help

Job DimGeography

Integration Profiling MDM

Repository

LOCAL: AdventureWorks_neu

- mssql_AdventureWorksDW_neu_mysql
 - DimCurrency 0.1
 - DimCurrencyAll 0.1
 - DimCustomerPerson_copy 0.1
 - DimDate 0.1
 - DimEmployee 0.1
 - DimGeography 0.1
 - DimLocation 0.1
 - DimPayHistory 0.1
 - DimProducts_Purchased 0.1
 - DimProductsAll_Copy 0.1
 - DimProductsAll 0.1
 - DimProductVendor 0.1
 - DimScrapReason 0.1
 - DimStore 0.1

tMap_1

tMSSqlInput_1 ("Address")

tMSSqlInput_2 ("CountryRegion")

tMSSqlInput_3 ("StateProvince")

tMySQLInput_1 ("dimSalesTerritory")

tMySQLOutput_1 ("dimgeography")

Execution

Run Kill Clear

[statistics] connected [statistics] disconnected Job_DimGeography ended at 18-17-05/04/2018 exit code=0

Advanced settings

Target Exec

Line limit 100

Wrap

6:17 PM 4/5/2018

MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1: dimcurrency

```

12 • SELECT count(*) FROM adventureworksdw_neu.dimterritory;
13 • SELECT count(*) FROM adventureworksdw_neu.dimpromotion;
14 • SELECT count(*) FROM adventureworksdw_neu.dimproductvendor;
15 • SELECT count(*) FROM adventureworksdw_neu.dimproductsall;
16 • SELECT count(*) FROM adventureworksdw_neu.dimproducts_purchased;
17 • SELECT count(*) FROM adventureworksdw_neu.dimpayhistory;
18 • SELECT count(*) FROM adventureworksdw_neu.dimlocation;
19 • SELECT count(*) as dimdate FROM adventureworksdw_neu.dimdate;
20 • SELECT count(*) FROM adventureworksdw_neu.dimcustomerperson;
21

```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

Result 32 x Read Only

Context Help Snippets

Information

Table: dimemployee

Columns:

- EmployeeSK int(11) AI PK
- BusinessEntityID int(11)
- EmployeeNationalID varchar(15)
- Manager int(11)
- ManagerEmployeeSK int(11)

Action Output

#	Time	Action	Message	Duration / Fetch
87	20:45:19	SELECT count(*) FROM adventureworksdw_neu.dimemployee	1 row(s) returned	0.000 sec / 0.000 sec
88	20:45:31	SELECT count(*) as dimEmployee FROM adventureworksdw_neu.dimempl...	1 row(s) returned	0.000 sec / 0.000 sec
89	20:46:02	SELECT count(*) as dimGeography FROM adventureworksdw_neu.dimgeo...	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

8:46 PM 4/5/2018

Row Count 402

DimLocation

Talend Data Fabric (6.4.1.20170623_1246) | AdventureWorks_neu (Connection: Local)

File Edit View Window Help

Job DimProdu... Job DimEmplo... Job Fact Wo... Job DimCurre... Job Fact Wor... Job FactPurc... Job DimGeogr... Job DimLoc...

Repository

LOCAL: AdventureWorks_neu

mssql_AdventureWorksDW_neu_mysql

- DimCurrency 0.1
- DimCurrencyAll 0.1
- DimCustomerPerson_copy 0.1
- DimDate 0.1
- DimEmployee 0.1
- DimGeography 0.1
- DimLocation 0.1
- DimPayHistory 0.1
- DimProducts_Purchased 0.1
- DimProductsAll_Copy 0.1
- DimProductsAll 0.1
- DimProductVendor 0.1
- DimScrapReason 0.1
- DimStore 0.1

Outline Code Viewer

CNTL_Job_Tracking_Stats_CHILD_1 (CNTL_Job_Tracki... tFileInputDelimited_1 (Location_us) tMap_1 tMySQLOutput_1 ("dimlocation")

Designer Code Jobscript

Job DimLocation

Basic Run

Execution

Run Kill Clear

Job DimLocation ended at 18:18 05/04/2018. [exit code=0]

Advanced settings Target Exec

Default

Name Value

Type here to search

6:18 PM 4/5/2018

MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1: dimcurrency

```

10 • SELECT count(*) FROM adventureworksdw_neu.dimshipmethod;
11 • SELECT count(*) FROM adventureworksdw_neu.dimcrareson;
12 • SELECT count(*) FROM adventureworksdw_neu.dimsalesterritory;
13 • SELECT count(*) FROM adventureworksdw_neu.dimpromotion;
14 • SELECT count(*) FROM adventureworksdw_neu.dimproductvendor;
15 • SELECT count(*) FROM adventureworksdw_neu.dimproductsall;
16 • SELECT count(*) FROM adventureworksdw_neu.dimproducts_purchased;
17 • SELECT count(*) FROM adventureworksdw_neu.dimpayhistory;
18 • (    ) dimlocation FROM adventureworksdw_neu.dimlocation;
19 • SELECT count(*) as dimGeography FROM adventureworksdw_neu.dimgeography;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

Result 33 | Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
88	20:45:31	SELECT count(*) as dimEmployee FROM adventureworksdw_neu.dimempl...	1 row(s) returned	0.000 sec / 0.000 sec
89	20:46:02	SELECT count(*) as dimGeography FROM adventureworksdw_neu.dimegeo...	1 row(s) returned	0.000 sec / 0.000 sec
90	20:46:57	SELECT count(*) as dimLocation FROM adventureworksdw_neu.dimlocation	1 row(s) returned	0.000 sec / 0.000 sec

Table: dimemployee

Columns:

- EmployeeSK int(11) AI PK
- BusinessEntityID int(11)
- EmployeeNationalID varchar(15)
- Manager int(11)
- ManagerEmployeeSK int(11)

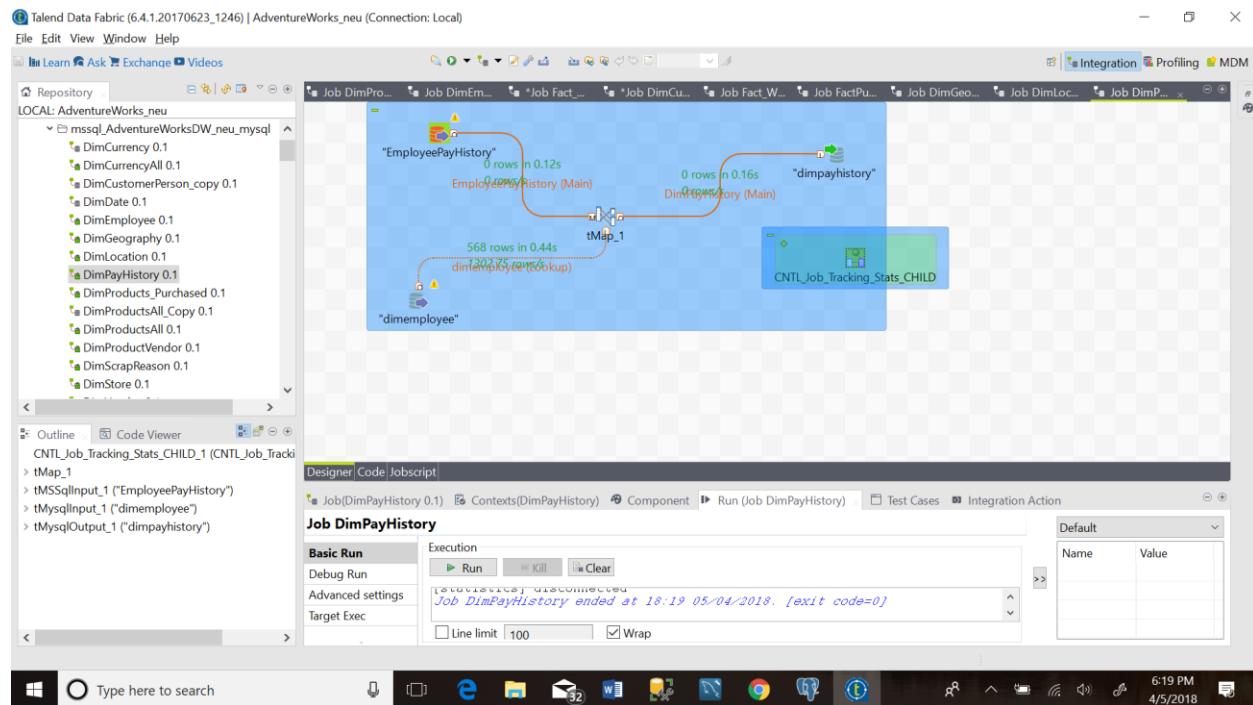
Object Info Session

Type here to search

8:47 PM 4/5/2018

Row Count 14

DimPayHistory



MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1: dimcurrency

```

12 • SELECT count(*) FROM adventureworksdw_neu.dimcustomerterritory;
13 • SELECT count(*) FROM adventureworksdw_neu.dimpromotion;
14 • SELECT count(*) FROM adventureworksdw_neu.dimproductvendor;
15 • SELECT count(*) FROM adventureworksdw_neu.dimproductsall;
16 • SELECT count(*) FROM adventureworksdw_neu.dimproducts_purchased;
17 • SELECT count(*) as dimPayHistory FROM adventureworksdw_neu.dimpayhistory;
18 • SELECT count(*) as dimLocation FROM adventureworksdw_neu.dimlocation;
19 • SELECT count(*) as dimGeography FROM adventureworksdw_neu.dimgeography;
20 • SELECT count(*) as dimdate FROM adventureworksdw_neu.dimdate;
21 • SELECT count(*) FROM adventureworksdw_neu.dimcustomerperson;

```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

Result 34 x

dimPayHistory

0

Action Output

#	Time	Action	Message	Duration / Fetch
89	20:46:02	SELECT count(*) as dimGeography FROM adventureworksdw_neu.dimgeo...	1 row(s) returned	0.000 sec / 0.000 sec
90	20:46:57	SELECT count(*) as dimLocation FROM adventureworksdw_neu.dimlocation	1 row(s) returned	0.000 sec / 0.000 sec
91	20:47:51	SELECT count(*) as dimPayHistory FROM adventureworksdw_neu.dimpay...	1 row(s) returned	0.000 sec / 0.000 sec

Result Grid | Form Editor | Context Help | Snippets

Information

Table: dimemployee

Columns:

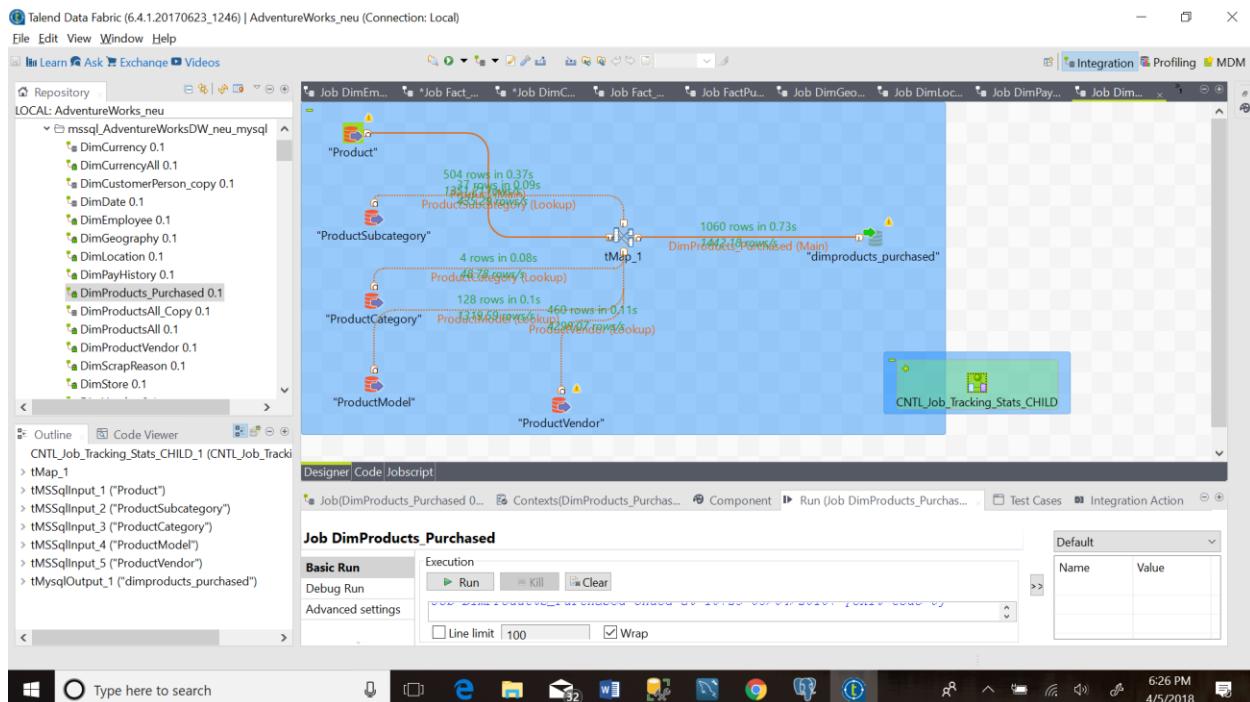
- EmployeeSK int(11) AI PK
- BusinessEntityID int(11)
- EmployeeNationalID varchar(15)
- Manager int(11)
- ManagerEmployeeSK int(11)

Object Info Session

Management Schemas

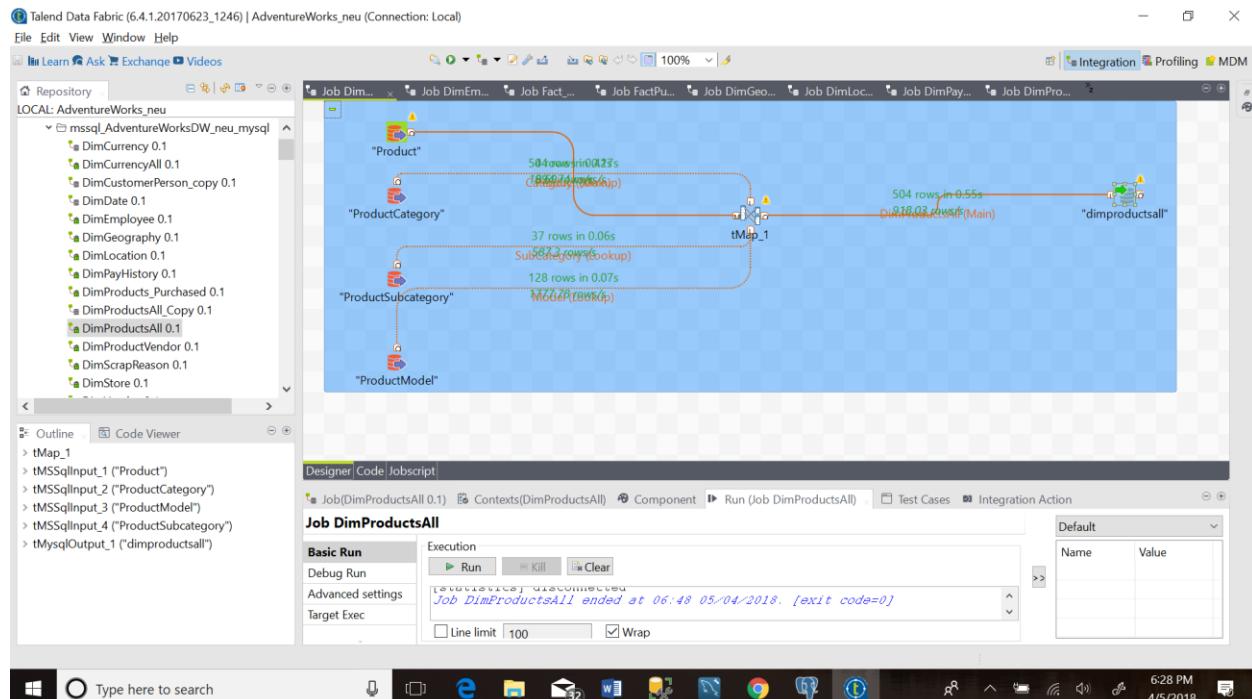
8:48 PM 4/5/2018

DimProduct_Purchased



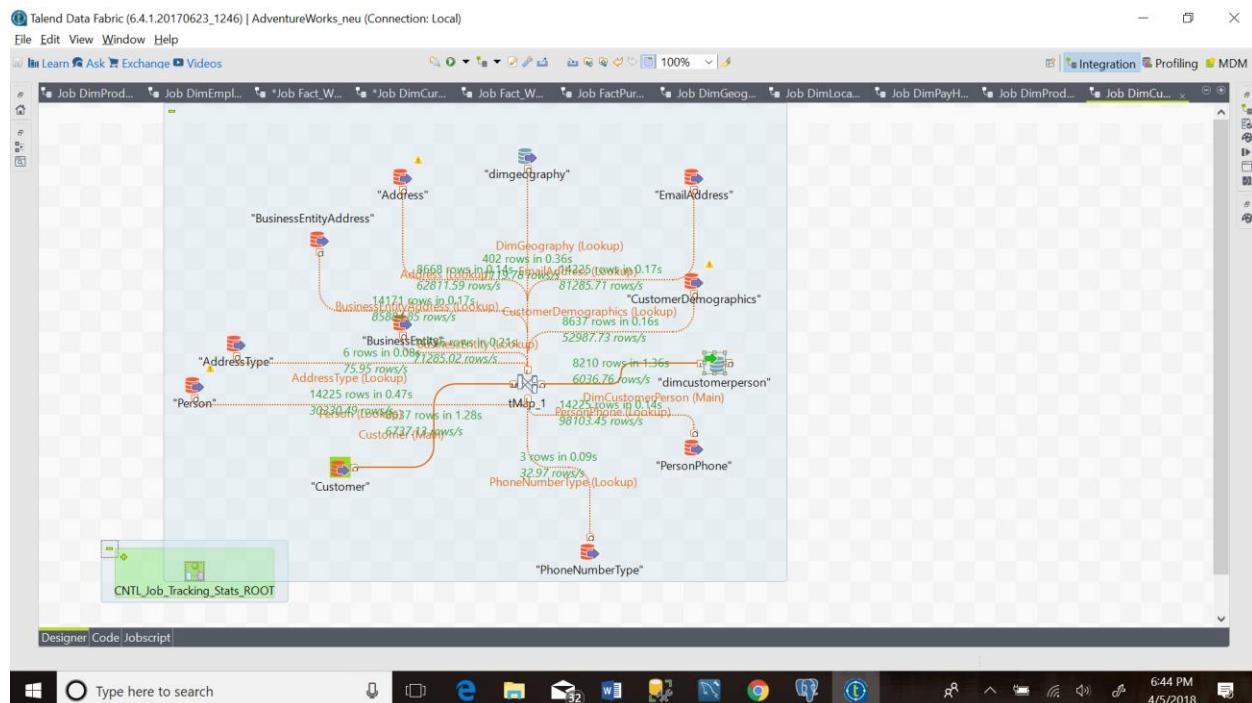
Row Count 1060

DimProductsAll



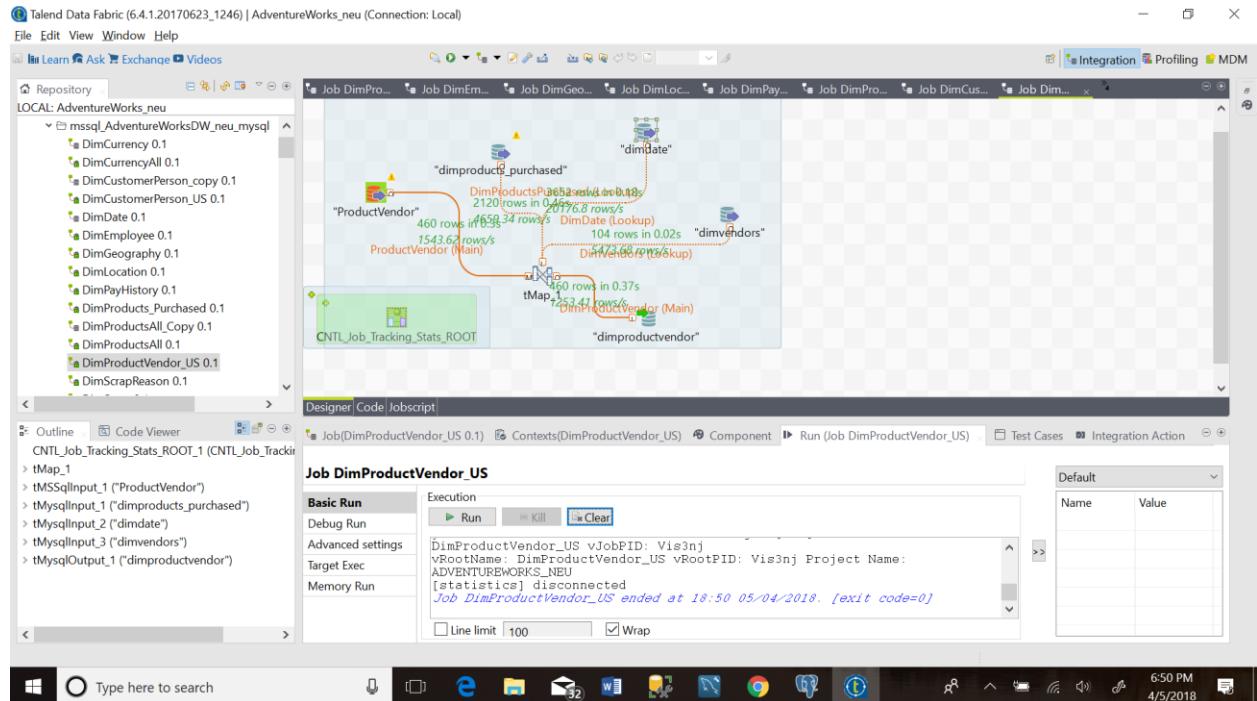
Row Count 504

DimCustomerPerson



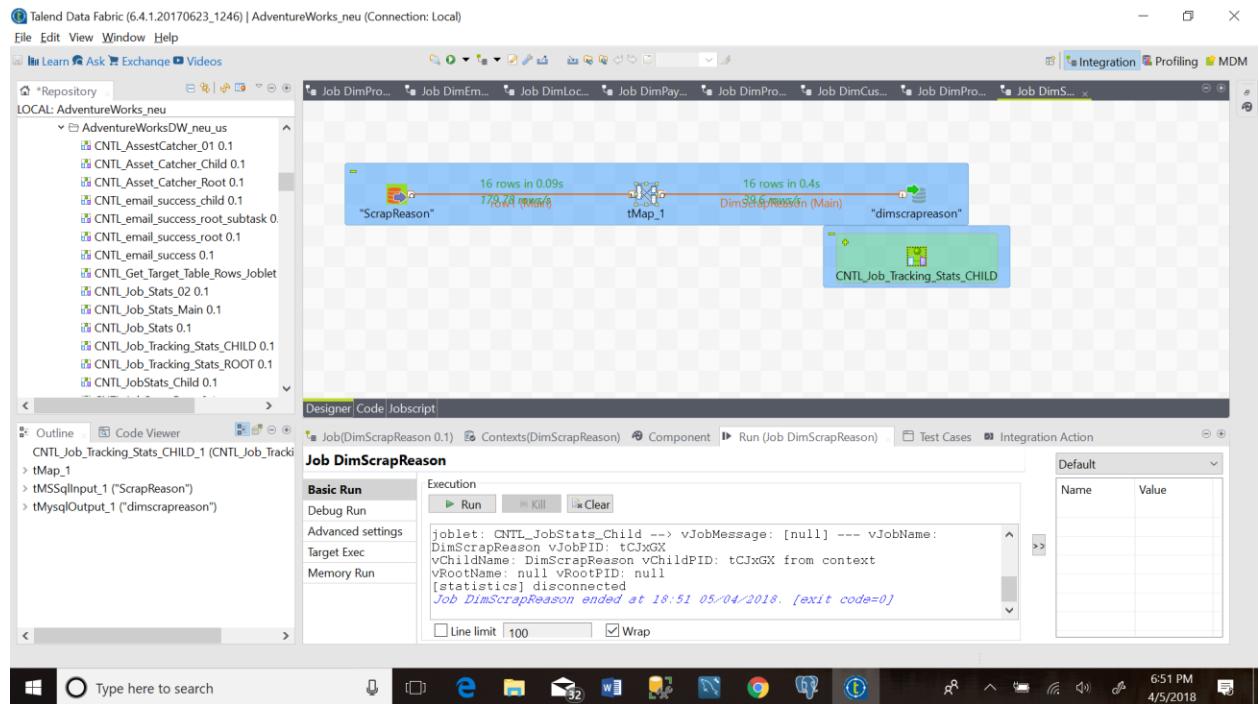
Row Count 8210

DimProductVedors



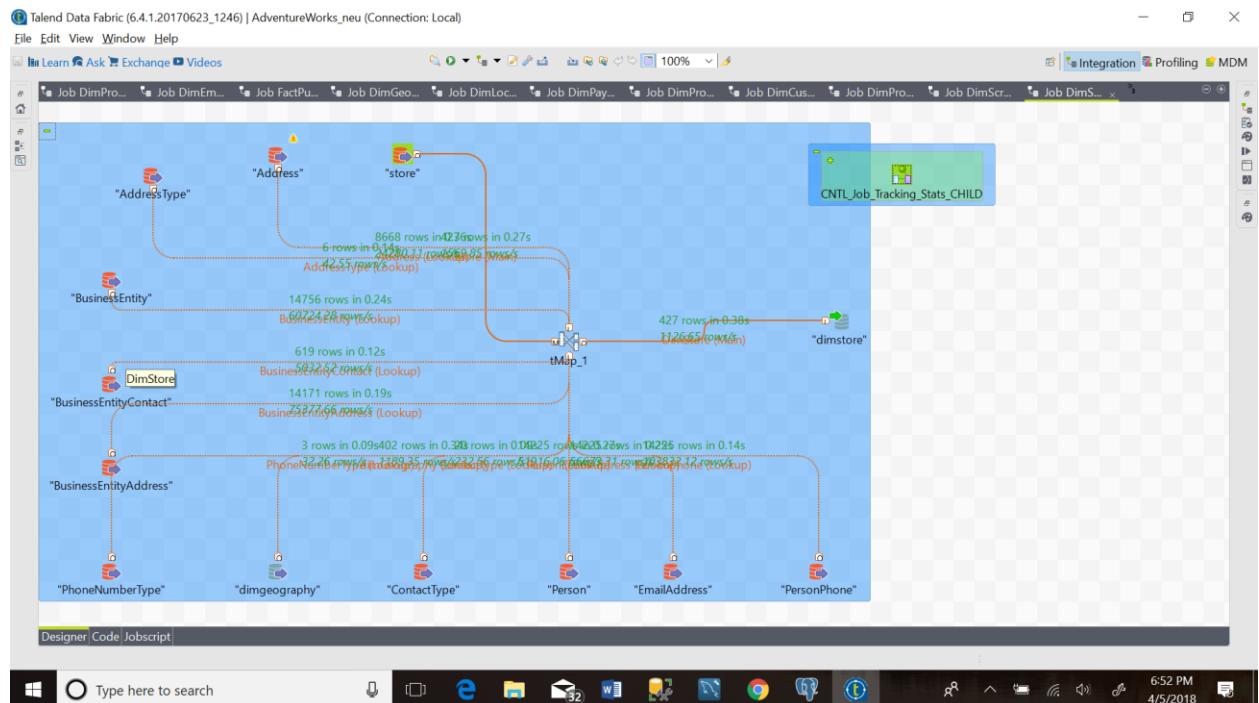
Row Count 104

DimScrapReason



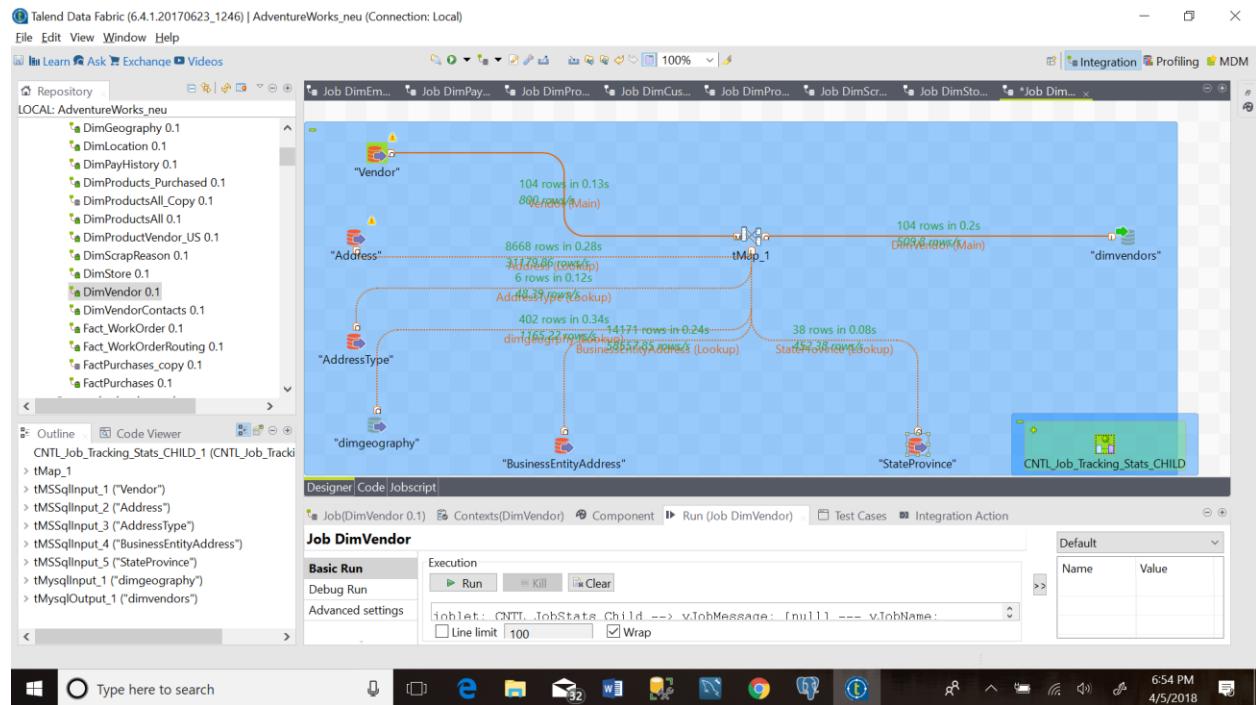
Row Count 16

DimStore



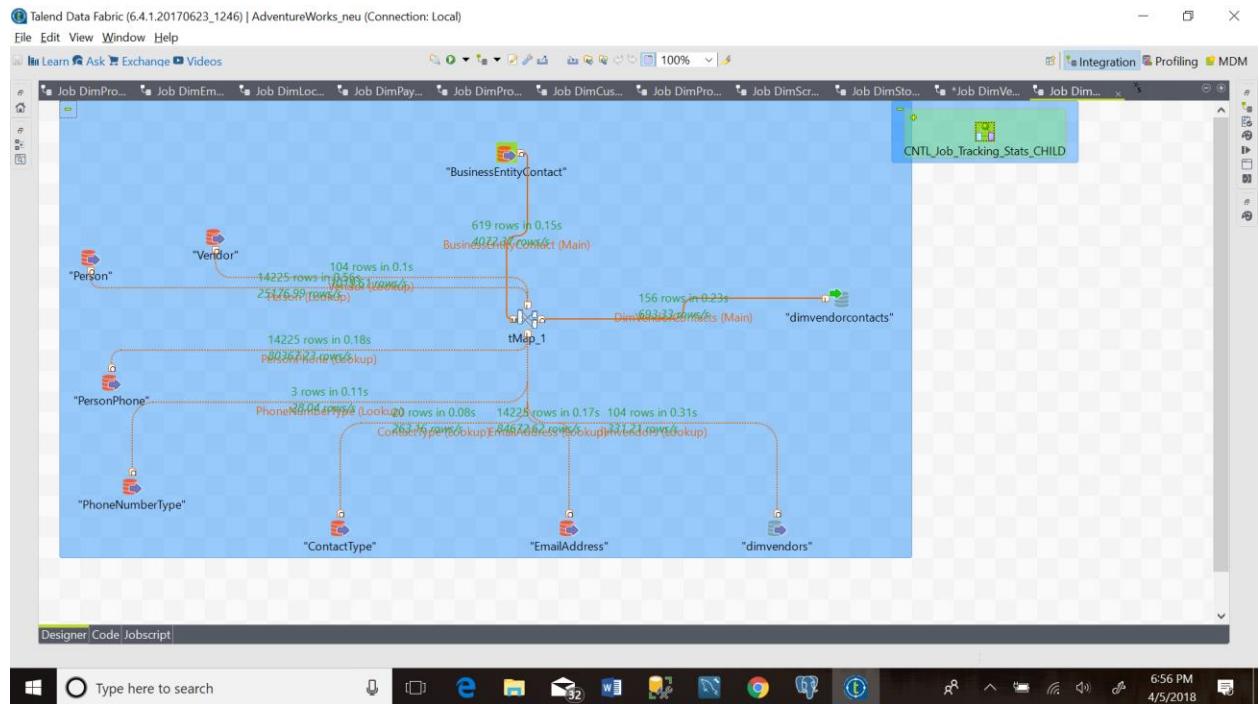
Row Count 427

DimVendors



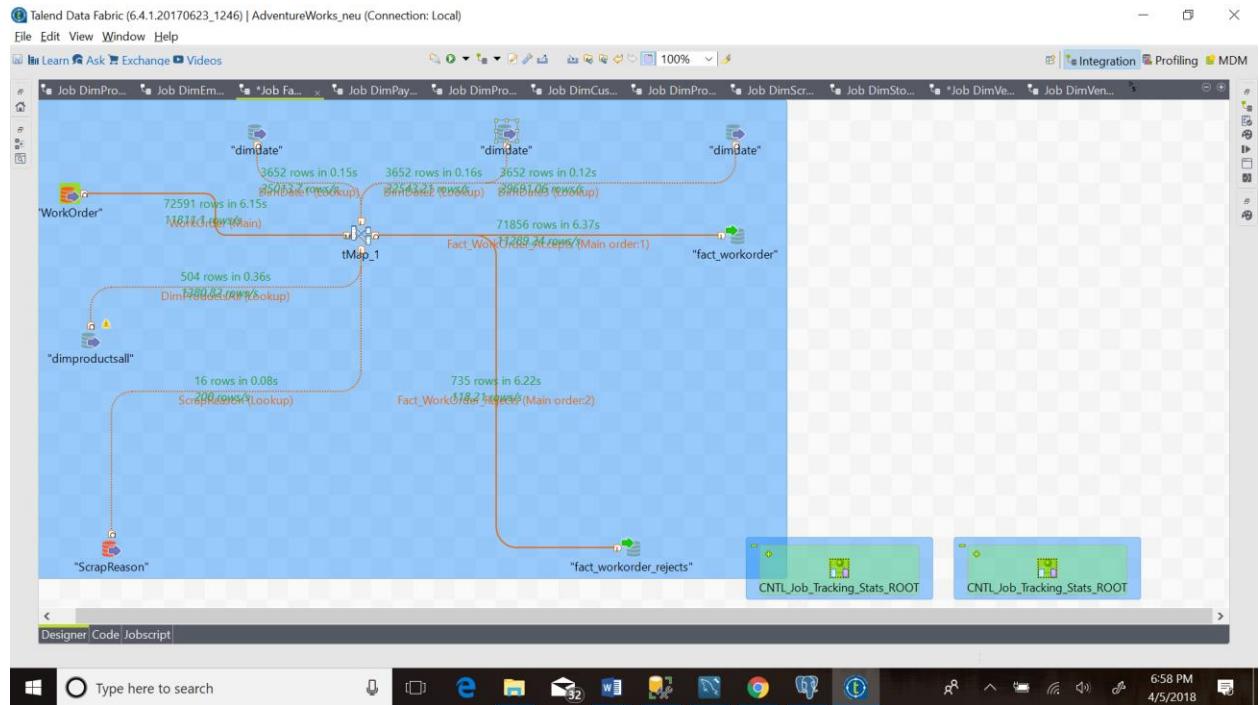
Row Count= 104

DimVendorContacts



Row Count = 156

Fact_WorkOrder



MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1: dimcurrency

```

1. SELECT count(*) as dimcurrency FROM adventureworksdw_neu.dimcurrency;
2. Execute the selected portion of the script or everything, if there is no selection: employee;
3. SELECT count(*) FROM adventureworksdw_neu.dimcustomerperson;
4. SELECT count(*) FROM adventureworksdw_neu.dimcustomerrejects;
5. SELECT count(*) FROM adventureworksdw_neu.dimcustomerrejects_rejects;
6. SELECT count(*) FROM adventureworksdw_neu.fact_workorderrejects;
7. SELECT count(*) FROM adventureworksdw_neu.dimvendordimensions;
8. SELECT count(*) FROM adventureworksdw_neu.dimvendortext;
9. SELECT count(*) FROM adventureworksdw_neu.dimstore;
10. SELECT count(*) FROM adventureworksdw_neu.dimshipmethod;

```

Result Grid: Fact_WorkOrder

71856

Result 35:

Action Output:

#	Time	Action	Message	Duration / Fetch
90	20:46:57	SELECT count(*) as dimLocation FROM adventureworksdw_neu.dimlocation	1 row(s) returned	0.000 sec / 0.000 sec
91	20:47:51	SELECT count(*) as dimPayHistory FROM adventureworksdw_neu.dimpay...	1 row(s) returned	0.000 sec / 0.000 sec
92	20:49:53	SELECT count(*) as Fact_WorkOrder FROM adventureworksdw_neu.fact_...	1 row(s) returned	0.032 sec / 0.000 sec

Information

Table: dimemployee

Columns:

- EmployeeSK int(11) AI PK
- BusinessEntityID int(11)
- EmployeeNationalID varchar(15)
- ManagerEmployeeSK int(11)
- ManagerBusinessEntityID int(11)

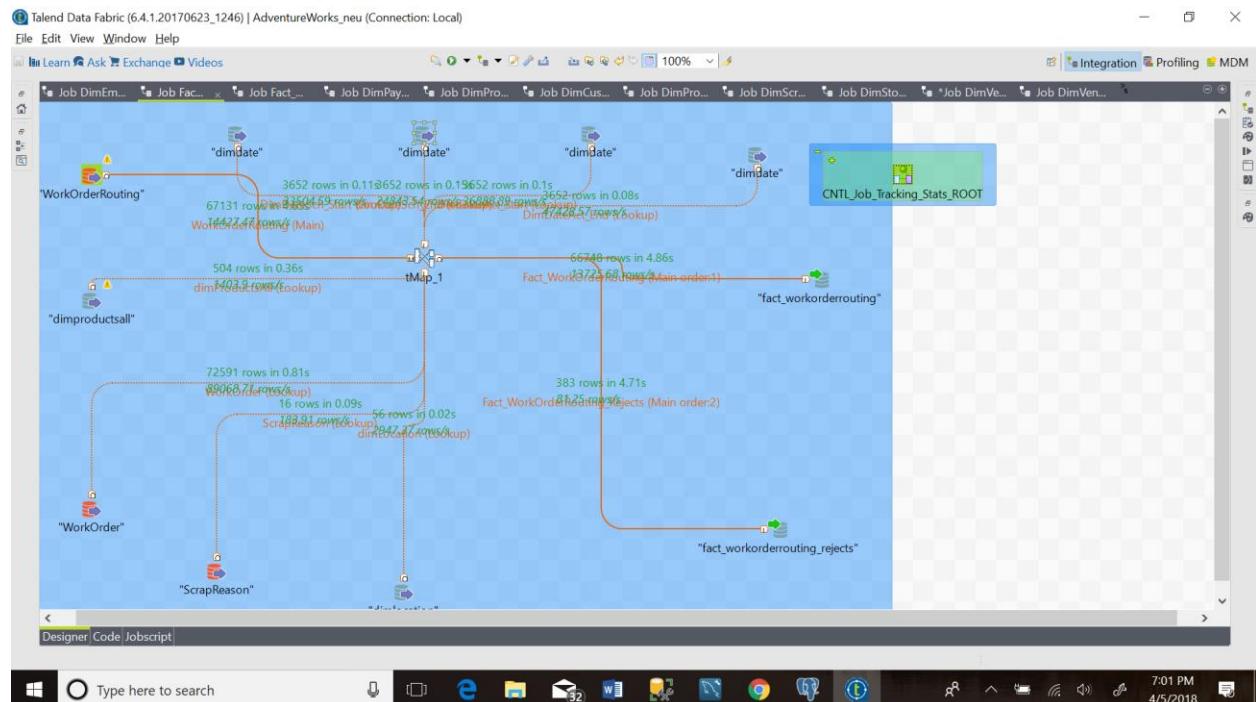
Object Info Session

Type here to search

8:49 PM 4/5/2018

Row Count = 71856, Rejects = 735

Fact_WorkOrderRouting



MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1: dimcurrency

```

1.   COUNT(*) AS dimcurrency FROM adventureworksdw_neu.dimcurrency;
2.   COUNT(*) AS dimEmployee FROM adventureworksdw_neu.dimemployee;
3.   COUNT(*) AS Fact_WorkOrderRejects FROM adventureworksdw_neu.dimfact_workorderrejects;
4.   COUNT(*) AS Fact_WorkOrder FROM adventureworksdw_neu.dimfact_workorder;
5.   COUNT(*) AS Fact_WorkOrderRoutingRejects FROM adventureworksdw_neu.dimfact_workorderrouting_rejects;
6.   COUNT(*) FROM adventureworksdw_neu.dimvendors;
7.   COUNT(*) FROM adventureworksdw_neu.dimstore;
8.   COUNT(*) FROM adventureworksdw_neu.dimproduct;
9.   COUNT(*) FROM adventureworksdw_neu.dimshipmethod;
10.  COUNT(*) FROM adventureworksdw_neu.dimterritory;

```

Result Grid: Fact_WorkOrderRouting

Row Count: 66748

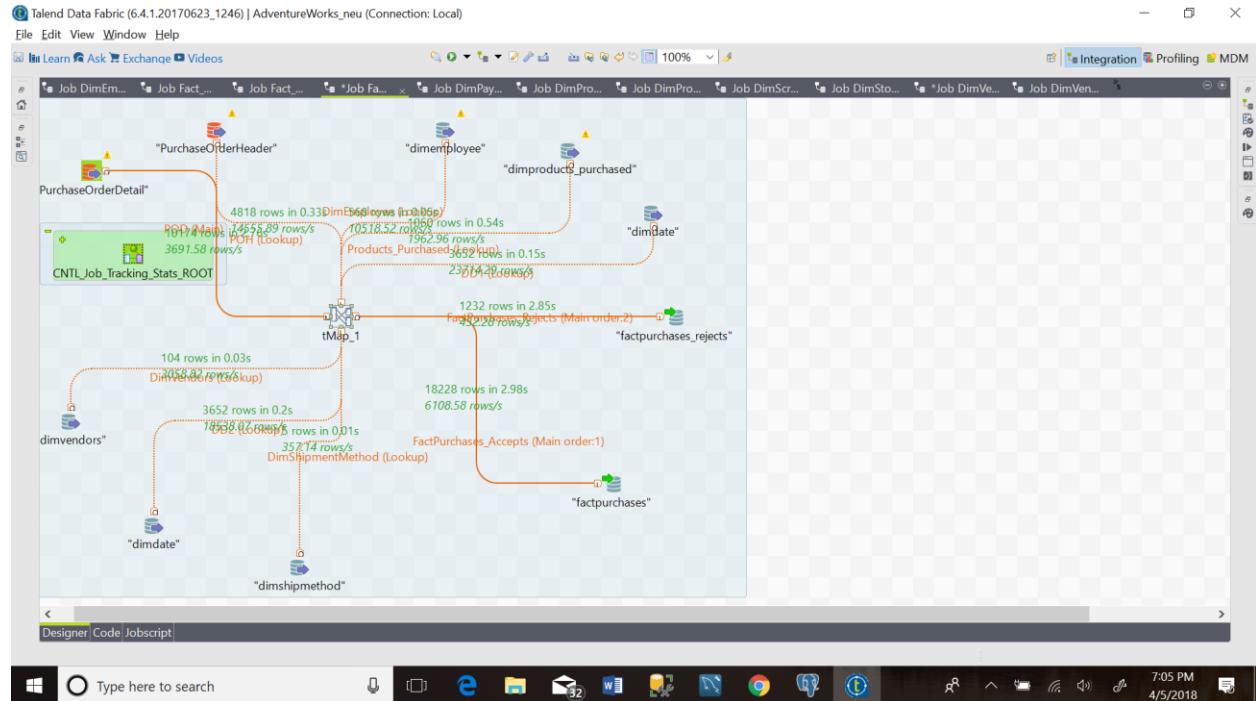
Result 36:

Action Output:

#	Time	Action	Message	Duration / Fetch
91	20:47:51	SELECT count(*) as dimPayHistory FROM adventureworksdw_neu.dimpay...	1 row(s) returned	0.000 sec / 0.000 sec
92	20:49:53	SELECT count(*) as Fact_WorkOrder FROM adventureworksdw_neu.dimfact...	1 row(s) returned	0.032 sec / 0.000 sec
93	20:50:48	SELECT count(*) as Fact_WorkOrderRouting FROM adventureworksdw_n...	1 row(s) returned	0.109 sec / 0.000 sec

Row Count = 66748, Rejects = 383

FactPurchases



```

12   . . . . . COUNT(*) FROM adventureworksdw_neu.dimSalesterritory;
13   . . . . . Execute the selected portion of the script or everything, if there is no selection
14   . . . . .
15   . . . . . COUNT(*) FROM adventureworksdw_neu.dimproductsall;
16   . . . . . COUNT(*) FROM adventureworksdw_neu.dimproducts_purchased;
17   . . . . . COUNT(*) AS dimPayHistory FROM adventureworksdw_neu.dimPayHistory;
18   . . . . . COUNT(*) AS dimLocation FROM adventureworksdw_neu.dimLocation;
19   . . . . . COUNT(*) AS dimGeography FROM adventureworksdw_neu.dimGeography;
20   . . . . . COUNT(*) AS dimdate FROM adventureworksdw_neu.dimdate;
21   . . . . . COUNT(*) FROM adventureworksdw_neu.dimCustomerPerson;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

FactPurchase
9114

Result 37 x Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
92	20:49:53	SELECT count(*) as Fact_WorkOrder FROM adventureworksdw_neu.fact_...	1 row(s) returned	0.032 sec / 0.000 sec
93	20:50:48	SELECT count(*) as Fact_WorkOrderRouting FROM adventureworksdw_n...	1 row(s) returned	0.109 sec / 0.000 sec
94	20:51:22	SELECT count(*) as FactPurchase FROM adventureworksdw_neu.factpur...	1 row(s) returned	0.031 sec / 0.000 sec

Row Count = 18228, Rejects = 1232

Some critical Project Standards:

Each table (dimensions & facts) need to record:

- *SOR_ID*
- *DI_Job_ID*
- *DI_Create_Date*
- *DI_Modified_Date*

Dimension or Fact Table

Column	Data Type	Key
table_id		PK
attributes....		
DI_Job_ID	Integer	FK
SOR_ID	Integer	FK
DI_Create_Date	DateTime	
DI_Modified_Date	DateTime	

(from BI Guidebook - Figure 12-16: Data Integration Table - Job Audit Columns)

- *SOR_ID* – This is the SOR identifier that will tie this row to a particular SOR. Use this when the table the row is sourced from multiple systems of records and enables the row to be tied to the specific SOR.
- *DI_Job_ID* – the data integration job identifier is be the job id that the data integration tool generated. This identifier is a foreign key to the data integration tool's processing metadata. If that metadata is available, then this link provides a powerful mechanism to analyze data integration processing and performance down to the level of a table's row. Note: The datatype for this column is specific to the data integration product used.
- *DI_Create_Date* – This is the date and time that this row was originally created in this table. Often a database trigger is used to insert the current time, but the data integration job could also insert the current time directly.
- *DI_Modified_Date* - This is the most recent date and time that this row has been modified in this table. Often a database trigger is used to insert the current time, but the data integration job could also insert the current time directly. It is often a standard practice to populate this column with an initial value far in the future such as "9999-12-31" rather than leaving it a NULL to avoid queries with NULLs when analyzing this column.

For SQL Server the SQL syntax for these columns is:

- DI_Job_ID nvarchar(20) NULL
- DI_Create_Date datetime NOT NULL DEFAULT (getdate())
- DI_Modified_Date datetime NOT NULL DEFAULT (getdate())

Every Talend job needs to tract job stats & errors

The steps to completing this:

1. Insert Joblet (supplied) that tracks job stats & errors (below)
2. Create columns as specified above
3. Insert values into those columns (below)

Joblets

- Every job created must have a Joblet that tracks stats via tStatCatcher, tLogCatcher & tAssetCatcher.
- For this project the Joblets provided:

- CNTL_Job_Tracking_Stats_ROOT – placed in top level job also referred to as the Root job
- CNTL_Job_Tracking_Stats_CHILD – Placed in every non-root job

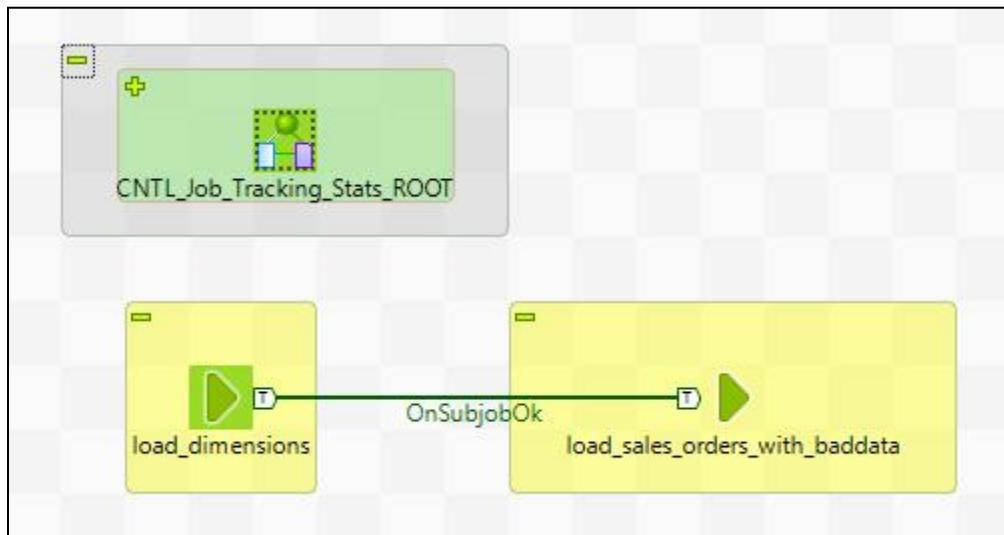


Figure 1: Joblet to track stats & errors in Root job

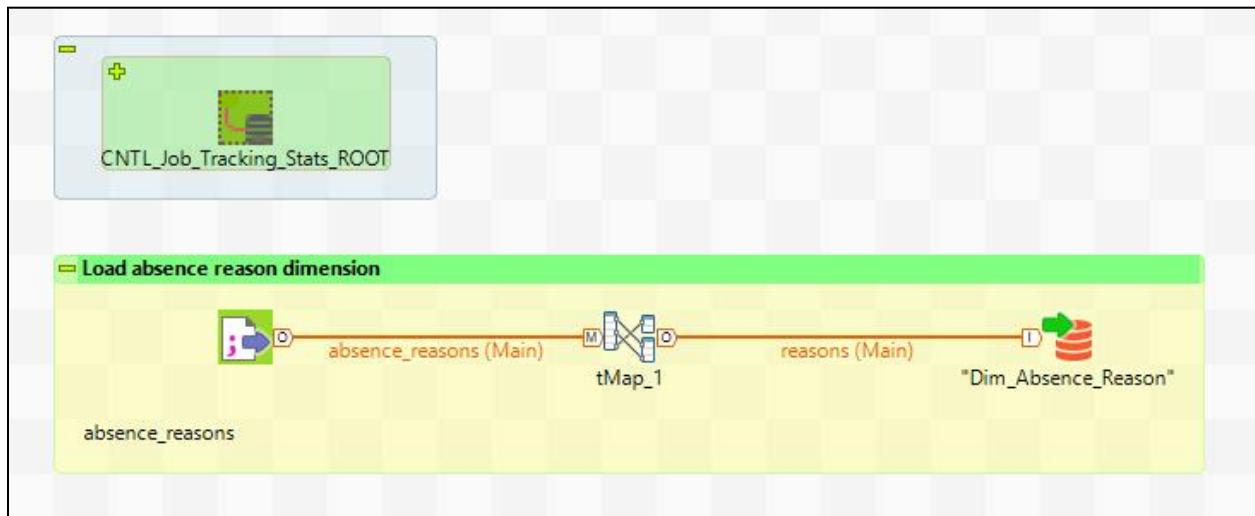


Figure 2: Joblet to track stats & errors in Child job

Insert values into standard columns

- DI_Job_ID - context.getProperty("vRootPID")
- DI_Create_Date - TalendDate.getCurrentDate()
- DI_Modified_Date - TalendDate.getCurrentDate()

context.getProperty("vRootPID")	DI_Job_ID
TalendDate.getCurrentDate()	DI_Create_Date
TalendDate.getCurrentDate()	DI_Modified_Date

Figure 3: tMap inserting values into target tables

Converting a date to a date surrogate key:

In the target table use the following function:

```
Integer.parseInt(routines.TalendDate.formatDate("yyyyMMdd",row1.TheSourceDate))
```



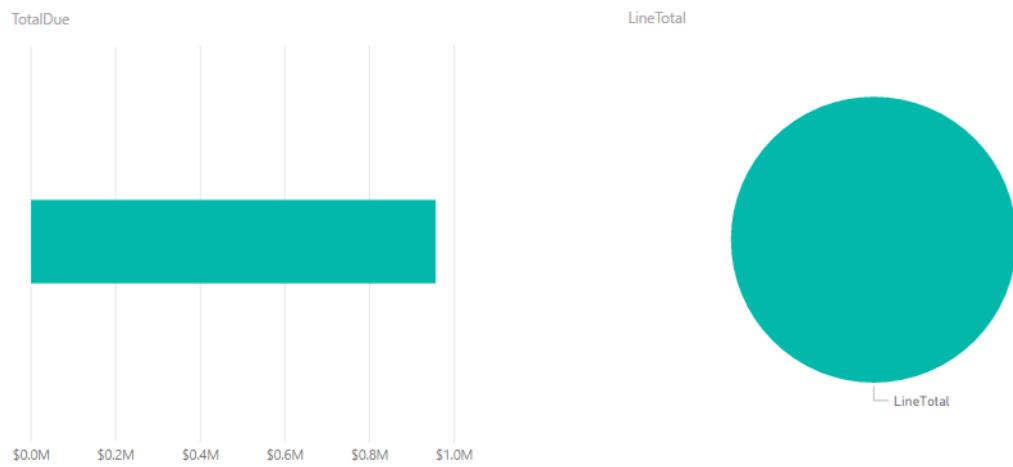
Figure 4: Converting a date to a date surrogate key

The example:

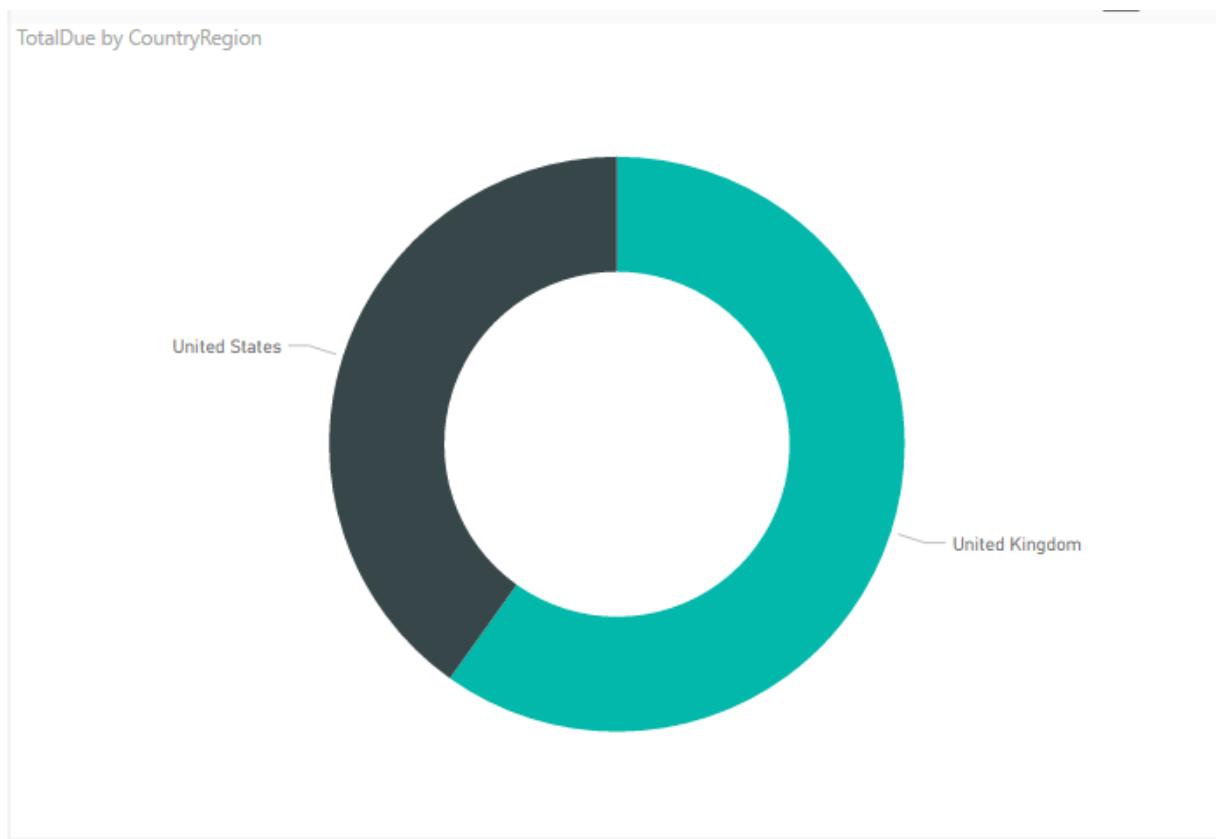
- Source column: header.OrderDate (date or datetime)
- Target column: internet_good.OrderDate_SK (integer)
- Expression:
 - `Integer.parseInt(routines.TalendDate.formatDate("yyyyMMdd",header.OrderDate))`

Data Visualization

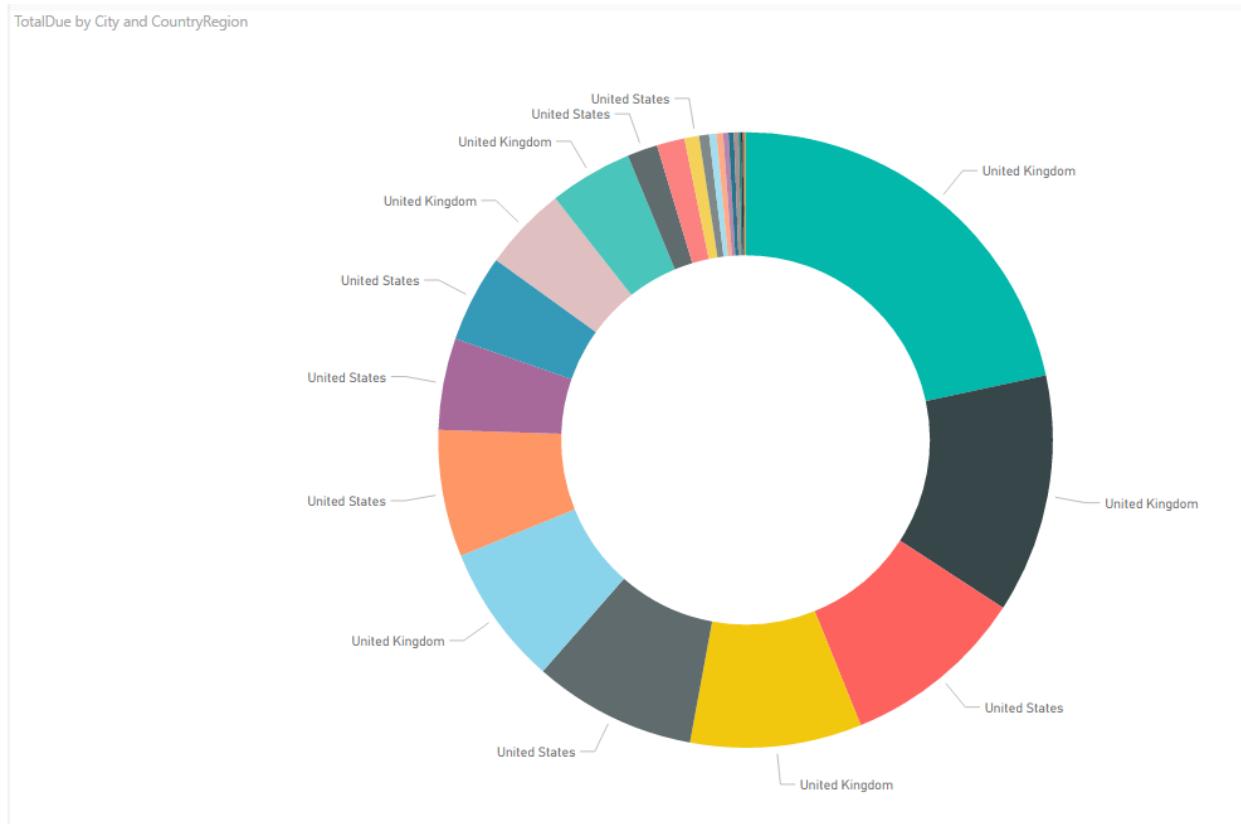
- a) Total sales a. Using SalesOrderHeader b. Using SalesOrderDetail



b) Total sales by country – ranked/sorted (highest to lowest)



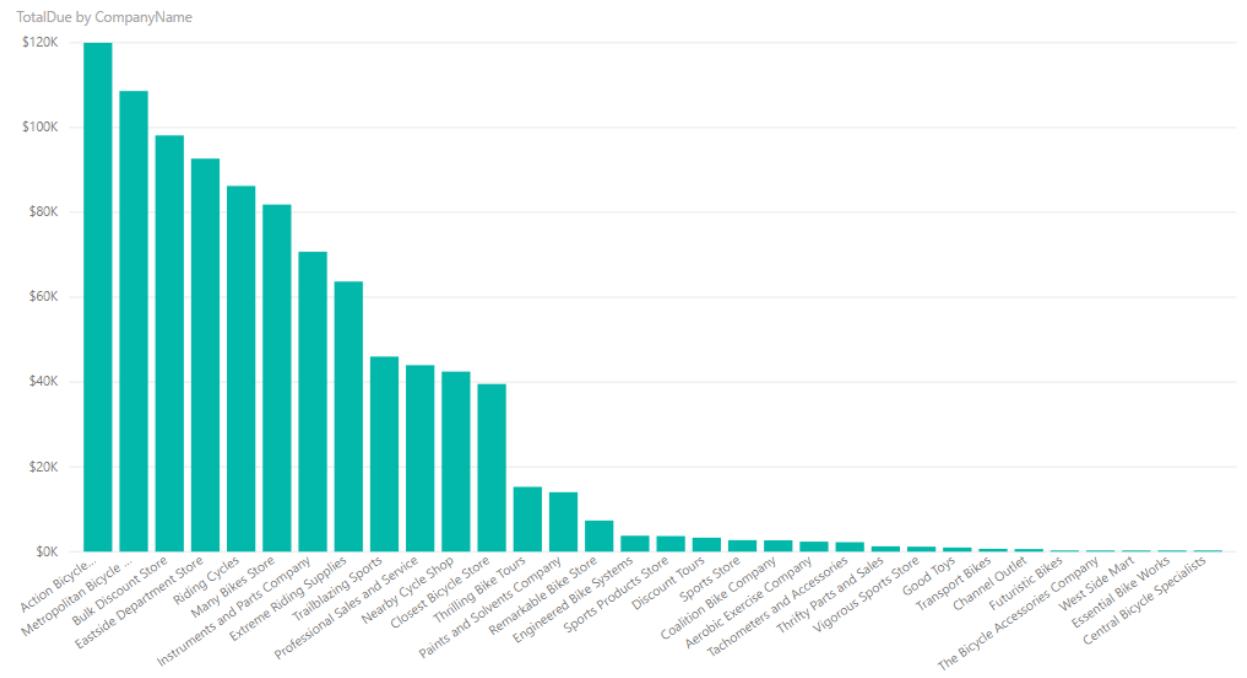
c) Total sales by city & country – ranked/sorted (highest to lowest)



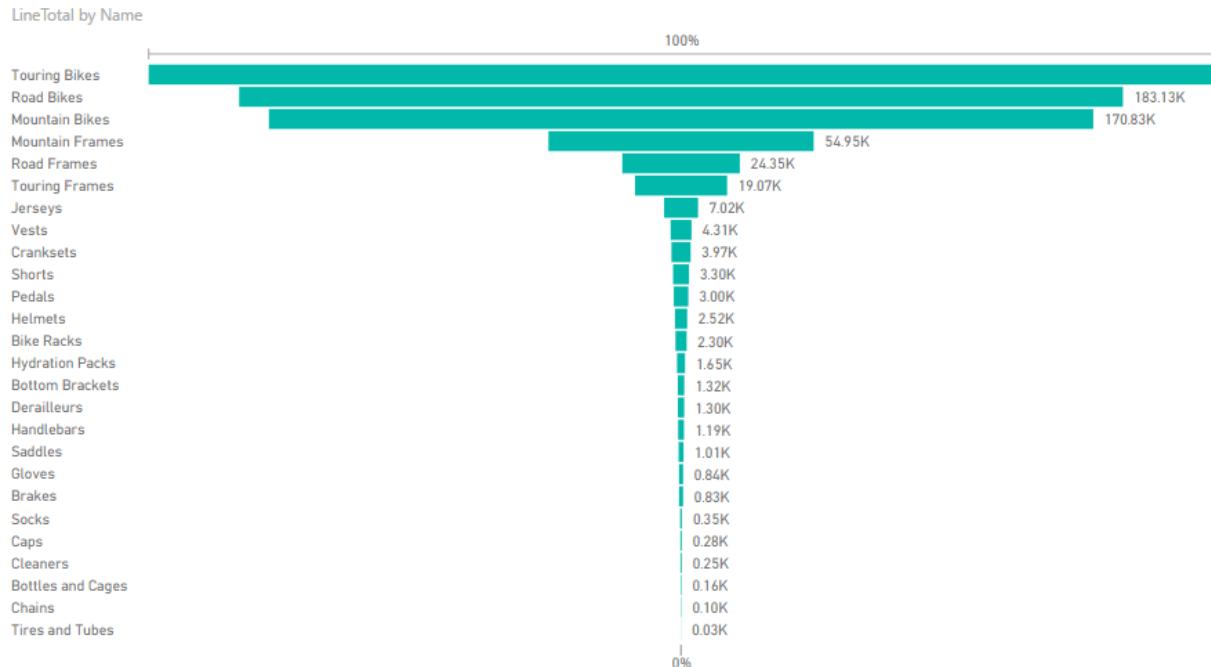
d) Total sales by customer (person) – ranked/sorted (highest to lowest)



e) Total sales by customer (company) – ranked/sorted (highest to lowest)



f) Sales by product category – ranked/sorted (highest to lowest)



g) Sales by product name – ranked/sorted (highest to lowest)

