Extract , Transform and Load (ETL)

Assignment 5 GAP

Momodou Lamin Keita

This report is presented as part of the requirements for assignment 5

Department: Information Technology

College: Nova Scotia Community College

Country: Halifax,Canada

Abstract

The term ”big data” is used to describe data sets so large that traditional do processing processes are inadequate.As technology becomes move seamlessly integrated into our daily lives, larger and larger amounts of data are being collected.This data comes in the form of customer transaction histories of a loyalty program for a major grocery retailer over the past decade or all application that have submitted to a university, whether it is a successful application or not.

Processing through these sets of data is a what some might refer to as business intelligence. With predictive analysis,reporting and or visualization tools these data sets can be a wealth of information to business. In the case major grocery retailer mention earlier an analysis method called clustering can be used to classify customers who live in the same city by shopping habits based on locations in the city. The university could also used to its data set to define its requirement policies and focus on location, ethnicity's and gender/race to improve diversity

The world of big data is vast and solution a built bases to a very strict set of business requirements/rules to ensure the accuracy and the information being produce.

Introduction

This Assignment will attempt to demonstrate a complete Extract, Transform and Load (E.T.L) process. E.T.L is a very important step in the building of data-warehouses. Databases in a data-warehouse that are a results of the E.T.L process may then be used for analysis and/or reporting. The results of the case shown in this assignment is being built for the purposes of being used as a model for: Data analysis and data reporting.

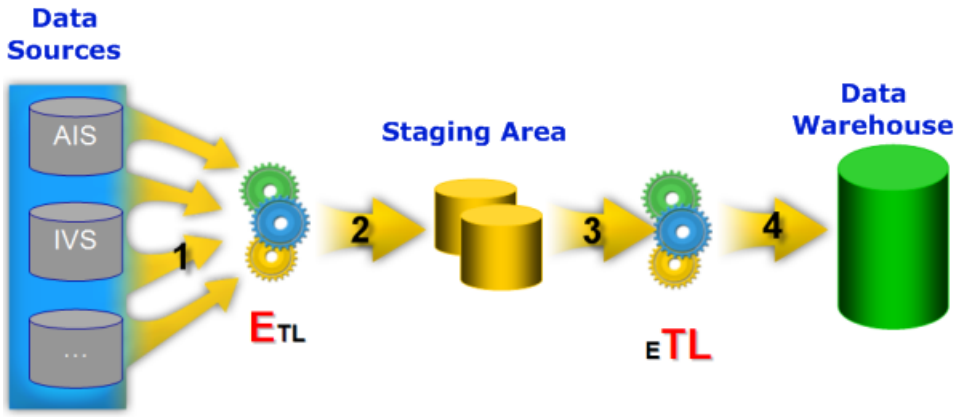


Figure 1: ETL Process

What is Extract Transform and Load(E.T.L)?

E.T.L stands for extraction, transformation and loading. It is a process in data-warehousing responsible for pulling data out of the source system(s) and placing it into a data-warehouse. All three(3) steps in the process work in unison as one tool.

- Extract : Extracting data from a source SAP, E.R.P, operational systems or archive systems which are the primary source of data for the data warehouse.
- Transform : which may involve joining together data from multiple sources, cleaning, filtering, validating, and applying business rules
- Load: Loading the extracted and transformed data into the data-warehouse

Extract Transform and Load(E.T.L) Tools

Below is a list of tools that can used to perform E.T.L operations. These are the most popular and the most widely used:

- IBM Websphere DataStage (Formerly known as Ascential DataStage and Ardent DataStage)
- Informatica PowerCenter
- Oracle ETL
- Ab Initio
- Pentaho Data Integration - Kettle Project (open source ETL)
- SAS ETL studio
- Cognos Decisionstream
- Business Objects Data Integrator (BODI)
- Microsoft SQL Server Integration Services (SSIS)

Business Requirements

Design and build a data-warehouse that will house facts and dimensions of a log server. The end product will be used for reporting and analysis services. Dimensions for Date,Time,Person,Report and the Main fact table are required. The data source is a S.Q.L Server database and the destination warehouse is also expected to be a S.Q.L server database.

The client is looking to use the end product to generate the following reports:

- Determine which reports are used the most and which are not being used over time
- Identify year over year usage trends
- Determine the time of day and/or day of week that the system is used the most
- Determine how much each user runs reports and when they run them

Tools

For the purposes of this project the extraction and transformation query building will be done with S.Q.L Server management Studio. Combining all three(3) steps will be handled using the S.Q.L Server Data Tools' S.Q.L Server Integration Services(S.S.I.S Packages).

[0]The tools selected for this project are project specific. The data source runs a S.Q.L server, the destination will be a S.Q.L Server so these specific tools have been selected.They are also not the only tools that can be used for the same task

Extract

Data Sources

The Data source for this project is a database called `C10_audit`. This table holds all the logging information of the reports being run on a business intelligence server.

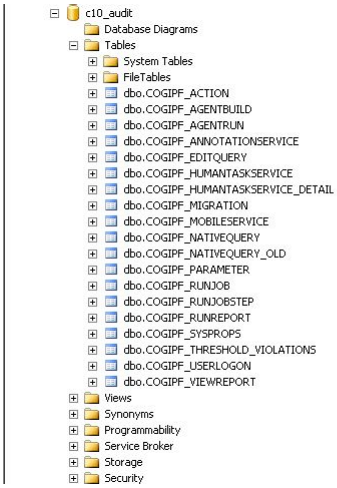
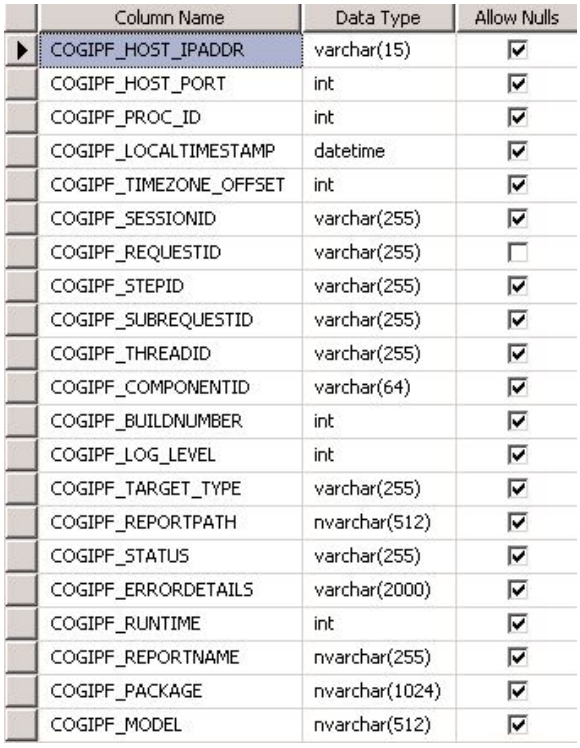


Figure 2: Source Database

The tables required to meet the business requirements have already between identified. `dbo.COGIPF_RUNREPORT` is the driving table with most of the information we require. The tables `dbo.COGIPF_PARAMETER` and `dbo.COGIPF_USERLOGON` will provide additional data.

Note

The extract stage of the E.T.L process can vary in size. Depending on what the business requirements are and what is needed to build the specified data-warehouse.The extractions and transformation steps will be at the same time and are not two separate steps.



Column Name	Data Type	Allow Nulls
COGIPF_HOST_IPADDR	varchar(15)	☑
COGIPF_HOST_PORT	int	☑
COGIPF_PROC_ID	int	☑
COGIPF_LOCALTIMESTAMP	datetime	☑
COGIPF_TIMEZONE_OFFSET	int	☑
COGIPF_SESSIONID	varchar(255)	☑
COGIPF_REQUESTID	varchar(255)	☐
COGIPF_STEPID	varchar(255)	☑
COGIPF_SUBREQUESTID	varchar(255)	☑
COGIPF_THREADID	varchar(255)	☑
COGIPF_COMPONENTID	varchar(64)	☑
COGIPF_BUILDNUMBER	int	☑
COGIPF_LOG_LEVEL	int	☑
COGIPF_TARGET_TYPE	varchar(255)	☑
COGIPF_REPORTPATH	nvarchar(512)	☑
COGIPF_STATUS	varchar(255)	☑
COGIPF_ERRORDETAILS	varchar(2000)	☑
COGIPF_RUNTIME	int	☑
COGIPF_REPORTNAME	nvarchar(255)	☑
COGIPF_PACKAGE	nvarchar(1024)	☑
COGIPF_MODEL	nvarchar(512)	☑

Figure 3: RunReport table definition. we require the columns `COGIPF_LOCALTIMESTAMP` : for time and date. `COGIPF_REPORTPATH`: can be tranformed into the report subject and the path to the report. `COGIPF_REPORTNAME`: Name of the report. `COGIPF_STATUS`: did the report run successfully or not. `COGIPF_RUNTIME`: how long it took to run the report.

Column Name	Data Type	Allow Nulls
COGIPF_HOST_IPADDR	varchar(128)	☑
COGIPF_HOST_PORT	int	☑
COGIPF_PROC_ID	int	☑
COGIPF_LOCALTIMESTAMP	datetime	☑
COGIPF_TIMEZONE_OFFSET	int	☑
COGIPF_SESSIONID	varchar(255)	☑
COGIPF_REQUESTID	nvarchar(255)	☐
COGIPF_STEPID	nvarchar(255)	☑
COGIPF_SUBREQUESTID	nvarchar(255)	☑
COGIPF_THREADID	varchar(255)	☑
COGIPF_COMPONENTID	varchar(64)	☑
COGIPF_BUILDNUMBER	int	☑
COGIPF_LOG_LEVEL	int	☑
COGIPF_STATUS	varchar(255)	☑
COGIPF_ERRORDETAILS	nvarchar(2000)	☑
COGIPF_LOGON_OPERATION	varchar(255)	☑
COGIPF_USERNAME	nvarchar(255)	☑
COGIPF_USERID	nvarchar(255)	☑
COGIPF_NAMESPACE	nvarchar(255)	☑
COGIPF_REMOTE_IPADDR	varchar(128)	☑
COGIPF_CAMID	nvarchar(512)	☑
COGIPF_TENANTID	nvarchar(255)	☑

Figure 4: UserLogon table definition. We require the columns `COGIPF_USERNAME`:holds the username of the user. `COGIPF_USERID`:holds the userid of the user.

Column Name	Data Type	Allow Nulls
COGIPF_LOCALTIMESTAMP	datetime	☑
COGIPF_SESSIONID	varchar(255)	☑
COGIPF_REQUESTID	nvarchar(255)	☐
COGIPF_SUBREQUESTID	nvarchar(255)	☑
COGIPF_STEPID	nvarchar(255)	☑
COGIPF_OPERATION	varchar(255)	☑
COGIPF_TARGET_TYPE	varchar(255)	☑
COGIPF_PARAMETER_NAME	varchar(255)	☑
COGIPF_PARAMETER_VALUE	nvarchar(512)	☑
COGIPF_PARAMETER_VALUE_BLOB	ntext	☑

Figure 5: `COGIPF_OPERATION`: holds the Operation ran by the user. e.g. execute

Destination

The Destination data-warehouse is where the data will be stored after it extracted and transformed. The data definition for the destination table columns will be the same as the source.In specific case the data will go through a transform that requires a destination table column of a different data definition.The following tables will be the main tables of our focus but will not be the only tables in the warehouse.Other tables will be added as a need arises.

Dimension Tables

A dimension is a table in a star schema of a data warehouse. A dimension table stores attributes, or dimensions, that describe the objects in a fact table. In data warehousing, a dimension is a collection of reference information about a measurable event. These events are known as facts and are stored in a fact table. Dimensions categorize and describe data warehouse facts and measures in ways that support meaningful answers to business questions. They form the very core of dimensional modeling.

Person Table

The Person dimension table will hold the measures for very person who has ever logged on to the system.The data table(s) for this dimension will be the `dbo.COGIF_USERLOGON`.

```
CREATE TABLE [dbo].[DimPerson](
  [PersonKey] [int] IDENTITY(1,1) NOT NULL,
  [PersonId] [varchar](255) NOT NULL,
  [Username] [varchar](255) NULL,
 CONSTRAINT [pk_person] PRIMARY KEY CLUSTERED
(
  [PersonKey] 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
```

Report Table

The Report table will hold the measures for reports. Its' data source is `dbo.COGIF_RUREPORT`.

```
CREATE TABLE [dbo].[DimReport](
  [ReportKey] [int] IDENTITY(1,1) NOT NULL,
  [ReportName] [varchar](200) NOT NULL,
  [ReportPath] [varchar](300) NOT NULL,
  [ReportSubject] [varchar](100) NOT NULL,
 CONSTRAINT [pk_report] PRIMARY KEY CLUSTERED
(
  [ReportKey] 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
```

Fact Tables

A fact table is the central table in a star schema of a data warehouse. A fact table stores quantitative information for analysis. A fact table works with dimension tables. A fact table holds the data to be analyzed, and a dimension table stores data about the ways in which the data in the fact table can be analyzed.

FactReportRun

This fact table will hold the Keys from and dimension tables required for analysis and reporting. This table will be the bases for all reporting and analysis required the business requirements, it is very important that the data inserted into this fact table is always accurate and update to date.

```
CREATE TABLE [dbo].[FactReportRun](
  [PersonKey] [int] NOT NULL,
  [ReportKey] [int] NOT NULL,
  [DateKey] [int] NOT NULL,
  [TimeKey] [int] NOT NULL,
  [Result] [varchar](255) NOT NULL,
  [Report_Runtime] [int] NOT NULL
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[FactReportRun]  WITH CHECK ADD  CONSTRAINT [fk_DimDate_DateKey]
    FOREIGN KEY([DateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
GO

ALTER TABLE [dbo].[FactReportRun] CHECK CONSTRAINT [fk_DimDate_DateKey]
```

```
GO

ALTER TABLE [dbo].[FactReportRun]  WITH CHECK ADD  CONSTRAINT [fk_DimPerson_PersonKey]
    FOREIGN KEY([PersonKey])
REFERENCES [dbo].[DimPerson] ([PersonKey])
GO

ALTER TABLE [dbo].[FactReportRun] CHECK CONSTRAINT [fk_DimPerson_PersonKey]
GO

ALTER TABLE [dbo].[FactReportRun]  WITH CHECK ADD  CONSTRAINT [fk_DimReport_ReportKey]
    FOREIGN KEY([ReportKey])
REFERENCES [dbo].[DimReport] ([ReportKey])
GO

ALTER TABLE [dbo].[FactReportRun] CHECK CONSTRAINT [fk_DimReport_ReportKey]
GO

ALTER TABLE [dbo].[FactReportRun]  WITH CHECK ADD  CONSTRAINT [fk_DimTime_TimeKey]
    FOREIGN KEY([TimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
GO

ALTER TABLE [dbo].[FactReportRun] CHECK CONSTRAINT [fk_DimTime_TimeKey]
GO
```

Transform

Person Table

```
SELECT DISTINCT
CAST(COGIPF_USERID as VARCHAR(255)) as UserID,
CAST(COGIPF_USERNAME as VarChar(255)) as Username
FROM [c10_audit].[dbo].[COGIPF_USERLOGON] t
WHERE COGIPF_LOCALTIMESTAMP =
(
    SELECT
    MAX(COGIPF_LOCALTIMESTAMP)
    FROM [c10_audit].[dbo].[COGIPF_USERLOGON] t1
    WHERE t1.COGIPF_USERID = t.COGIPF_USERID
)
AND [COGIPF_USERNAME] NOT IN ('','not available')
AND [COGIPF_USERID] NOT IN ('','null')
```

Person notes:

- `SELECT DISTINCT` : because the source table logs every time a user logs in.
- `COGIPF_LOCALTIMESTAMP` sub-select: even with "select distinct" a user can appear more that once, if their is a change in the user's last name or first name. The user table measures are built from the last successful login by using "max" `COGIPF_LOCALTIMESTAMP`.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT[COGIPF_LOCALTIMESTAMP]
      ,[COGIPF_USERNAME]
      ,[COGIPF_USERID]
  FROM [c10_audit].[dbo].[COGIPF_USERLOGON]
```

.00 % Results Messages

	COGIPF_LOCALTIMESTAMP	COGIPF_USERNAME	COGIPF_USERID
1	2014-05-15 10:43:01.820	Caudle,Paul	W0262887
2	2014-05-15 10:46:52.210	Gillis,Maureen	W0099170
3	2014-05-15 10:58:29.483	Pike,Susan	W0272328
4	2014-05-15 11:07:47.123	Creelman,Michelle	W0001353
5	2014-05-15 11:08:51.970	Drapeau,Suzanne	W0001313
6	2014-05-15 11:09:29.520	Chaulk,Dennis	W0106030
7	2014-05-16 11:33:40.370	not available	not available
8	2014-05-16 11:33:41.760	not available	not available
9	2014-05-16 11:33:46.747	not available	not available
10	2014-07-09 21:00:41.493	Cognos Admin	cognos_admin

Figure 6: sample date from `dbo.COGIF_USERLOGON`

	UserID	Username
1	SysOp.CD	Deveau,Christian (SysOp)
2	W0000207	Padovani,Karen
3	W0000372	Ballantyne,Shelley
4	W0000473	Moore,Andrew
5	W0000478	Wilms,Karla
6	W0000696	Butt,Arlene
7	W0000844	Driscoll,John
8	W0000849	MacDonald,George
9	W0001054	Bate,Jim
10	W0001106	Murray-Sellars,Sharon
11	W0001393	Foster,Monica
12	W0001566	Theriault,Lisa
13	W0001651	Bennett,Janice
14	W0001717	Brown,Steve

Figure 7: result from the execution of the script. it successfully returned back 462 rows of distinct& update-to-date records

Report Table

```
SELECT  DISTINCT
Cast(REPORTNAME as varchar(200)) as ReportName,
Cast
  (
   SUBSTRING
       (
       TEMPREPORTPATH,
       1,
       ((CHARINDEX(CHAR(39),TEMPREPORTPATH)-1))
       ) as Varchar(300)) AS REPORTPATH,
Cast
  (
   SUBSTRING
       (
       TEMPSUBJECT,
       CHARINDEX(CHAR(39),TEMPSUBJECT) +1,
       (charindex(char(93),TEMPSUBJECT,2))-(CHARINDEX(CHAR(39),TEMPSUBJECT) +2)
        ) as varchar(100)) AS [SUBJECT]
FROM
(
SELECT
          SUBSTRING (
          COGIPF_REPORTNAME,
          CHARINDEX(CHAR(39),COGIPF_REPORTNAME) +1,
          ((CHARINDEX(CHAR(93),COGIPF_REPORTNAME)-2) -(CHARINDEX(CHAR(39),COGIPF_REPORTNAME
    )))
    ) AS REPORTNAME,
          SUBSTRING(
          COGIPF_REPORTPATH,
          PATINDEX('%]/folder%',COGIPF_REPORTPATH),
          PATINDEX('%]/folder%',COGIPF_REPORTPATH)
    ) AS TEMPSUBJECT,
    SUBSTRING(
```

```
        (Replace(COGIPF_REPORTPATH, ''']/folder[@name='', '->')),
        (24),
        Len(COGIPF_REPORTPATH)
    ) as TEMPREPORTPATH
FROM [c10_audit].[dbo].[COGIPF_RUNREPORT]

WHERE COGIPF_REPORTPATH LIKE '%Nova Scotia Community College%'
AND [COGIPF_REPORTNAME] NOT LIKE '%adHocReport%'
AND [COGIPF_REPORTNAME] NOT LIKE '%analysis%'
) a
```

```
/content/folder[@name='Nova Scotia Community College Reporting']/folder[@name='Finance']/package[@name='Reports - Budget (2014/15)']/report[@name='BRP 101 Income Statement Report Fiscal Year']
```

Figure 8: Sample data from the `COGIF_REPORTPATH` Column

Results | Messages

	COGIPF_REPORTPATH	COGIPF_STATUS	COGIPF_RUNTIME	COGIPF_REPORTNAME
1	/content/folder[@name='Nova Scotia Community C...	Success	28	report[@name='BRP 101 Income Statement Report Fi...
2	/content/folder[@name='Nova Scotia Community C...	Success	4165	report[@name='ENR 210 Tracking Admissions to First ...
3	/content/folder[@name='Nova Scotia Community C...	Success	385	report[@name='ENR 210 Tracking Admissions to First ...
4	/content/folder[@name='Nova Scotia Community C...	Success	8455	report[@name='ENR 310 Capacity and Target Enrolm...
5	/content/folder[@name='Nova Scotia Community C...	Success	1209	report[@name='Weekly Confirmations by School']
6	/content/folder[@name='Nova Scotia Community C...	Success	978	report[@name='Confirmations vs Y1 Target']
7	/content/folder[@name='Nova Scotia Community C...	Success	4194	report[@name='FIN 110 Income Statement']
8		Success	522	
9	/content/folder[@name='Nova Scotia Community C...	Success	244	report[@name='FIN 130 Detailed Trial Balance']
10	/content/folder[@name='Nova Scotia Community C...	Success	527	report[@name='FIN 130 Detailed Trial Balance']

Figure 9: Sample `COGIF_RUNREPORT` data

Report notes:

- This query will extract "ReportName" from the column `COGIPF_REPORTNAME`, "ReportPath" from `COGIPF_REPORTPATH` and "subject" from `COGIPF_REPORTPath` of `dbo.COGIF_RUNREPORT`.

Results | Messages

	ReportName	REPORTPATH	SUBJECT
1	FND 330 Endowment Donor Report	Nova Scotia Community College Reporting->Foundation->Development Reports (Need ...	Foundation
2	ENR 302 Milestone Enrolment Profile	Nova Scotia Community College Reporting->Enrolment	Enrolment
3	FND 300 Foundation Endowment Report	Nova Scotia Community College Reporting->Foundation->Development Reports (Need ...	Foundation
4	Confirmations and Confirmed Returning Students as a...	Nova Scotia Community College Reporting->Dashboards->Admissions->Individual Repo...	Dashboards
5	ENR 313 Tracking Graduates for Current Academic ...	Nova Scotia Community College Reporting->Enrolment->Production Release 2	Enrolment
6	Users w/ Enrolment Reporting Rights	Nova Scotia Community College Reporting->Finance->Private Reports	Finance
7	301 - Detailed Trial Balance	Nova Scotia Community College Reporting->Finance->Private Reports->New Combined...	Finance
8	FIN 136 Audit Detail	Nova Scotia Community College Reporting->Finance->Audit Reports	Finance
9	Budget Comparison (Year over Year)	Nova Scotia Community College Reporting->Finance	Finance
10	FIN - 110 Income Statement	Nova Scotia Community College Reporting->Finance->Phase 2 Development	Finance

Figure 10: Sample `COGIF_RUNREPORT` data

FactReportRun Table

```
SELECT
        CAST(UL.COGIPF_USERID AS VARCHAR(200)) AS RAN_BY_USERNAME,
        SUBSTRING(REPORT_NAME,CHARINDEX(CHAR(39),REPORT_NAME) +1,((CHARINDEX(CHAR(93),
      REPORT_NAME)-2) -(CHARINDEX(CHAR(39),REPORT_NAME)))) AS REPORTNAME,
        SUBSTRING(TEMPSUBJECT,CHARINDEX(CHAR(39),TEMPSUBJECT) +1,(charindex(char(93),
      TEMPSUBJECT,2))-(CHARINDEX(CHAR(39),TEMPSUBJECT) +2)) AS SUBJECT,
        SUBSTRING(TEMPREPORTPATH,1,(((CHARINDEX(CHAR(39),TEMPREPORTPATH)-1))) as REPORTPATH,
        [START_DATE],
        START_TIME,
        TASK,
        TASK_RUNTIME,
        RESULT
FROM
      (
       SELECT
       cast(RR.COGIPF_REPORTNAME as varchar(200)) AS REPORT_NAME,
       CAST(SUBSTRING(RR.COGIPF_REPORTPATH,PATINDEX('%]/folder%',COGIPF_REPORTPATH),
      PATINDEX('%]/folder%',COGIPF_REPORTPATH)) AS VARCHAR(100)) AS TEMPSUBJECT,
       cast(SUBSTRING((Replace(COGIPF_REPORTPATH,''']/folder[@name=''','->')),(24),Len(
      COGIPF_REPORTPATH))as varchar(300)) as TEMPREPORTPATH,
       P.COGIPF_OPERATION AS TASK,
```

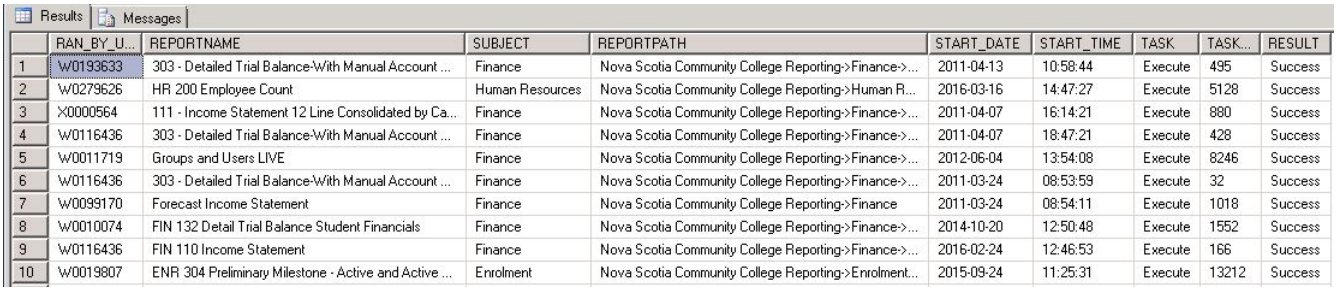
```
      cast(RR.COGIPF_LOCALTIMESTAMP as Date)  AS [START_DATE],
      cast(RR.COGIPF_LOCALTIMESTAMP as Time(0))  AS [START_TIME],
      RR.COGIPF_SESSIONID,
      RR.COGIPF_STATUS AS RESULT,
      RR.COGIPF_RUNTIME AS TASK_RUNTIME
      FROM c10_audit.dbo.COGIPF_RUNREPORT  AS RR
  inner join(
  select distinct  COGIPF_REQUESTID, COGIPF_OPERATION
  from COGIPF_PARAMETER
  where cogipf_operation = 'Execute'
  ) as p ON RR.COGIPF_REQUESTID = P.COGIPF_REQUESTID
       where
    p.COGIPF_OPERATION ='Execute'
      AND  p.COGIPF_OPERATION !=''
      AND
    [COGIPF_REPORTNAME] NOT LIKE'%adHocReport%'
      AND  [COGIPF_REPORTNAME] NOT  Like'%View%'
      AND  [COGIPF_REPORTNAME] NOT  Like '%Analysis%'
      AND COGIPF_REPORTPATH LIKE '%Nova Scotia Community College%'
     ) as R2015COG
(
SELECT distinct COGIPF_SESSIONID, COGIPF_USERID, COGIPF_USERNAME
  FROM [c10_audit].[dbo].[COGIPF_USERLOGON]
) as ul on R2015COG.COGIPF_SESSIONID = UL.COGIPF_SESSIONID
where  [COGIPF_USERNAME] NOT IN ('','not available')
  and [COGIPF_USERID] NOt IN ('','null')
```

FactReportRun notes:

- The values return by this script will not match the table definition of `Report_Audit.dbo.FactReportRun`.

  These have go through a lookup tranformation and are return as Keys.

Now that that destination tables are created.The scripts to extract and transform the data.The



Results | Messages

	RAN_BY_U...	REPORTNAME	SUBJECT	REPORTPATH	START_DATE	START_TIME	TASK	TASK...	RESULT
1	W0193633	303 - Detailed Trial Balance-With Manual Account ...	Finance	Nova Scotia Community College Reporting->Finance->...	2011-04-13	10:58:44	Execute	495	Success
2	W0279626	HR 200 Employee Count	Human Resources	Nova Scotia Community College Reporting->Human R...	2016-03-16	14:47:27	Execute	5128	Success
3	X0000564	111 - Income Statement 12 Line Consolidated by Ca...	Finance	Nova Scotia Community College Reporting->Finance->...	2011-04-07	16:14:21	Execute	880	Success
4	W0116436	303 - Detailed Trial Balance-With Manual Account ...	Finance	Nova Scotia Community College Reporting->Finance->...	2011-04-07	18:47:21	Execute	428	Success
5	W0011719	Groups and Users LIVE	Finance	Nova Scotia Community College Reporting->Finance->...	2012-06-04	13:54:08	Execute	8246	Success
6	W0116436	303 - Detailed Trial Balance-With Manual Account ...	Finance	Nova Scotia Community College Reporting->Finance->...	2011-03-24	08:53:59	Execute	32	Success
7	W0099170	Forecast Income Statement	Finance	Nova Scotia Community College Reporting->Finance	2011-03-24	08:54:11	Execute	1018	Success
8	W0010074	FIN 132 Detail Trial Balance Student Financials	Finance	Nova Scotia Community College Reporting->Finance->...	2014-10-20	12:50:48	Execute	1552	Success
9	W0116436	FIN 110 Income Statement	Finance	Nova Scotia Community College Reporting->Finance->...	2016-02-24	12:46:53	Execute	166	Success
10	W0019807	ENR 304 Preliminary Milestone - Active and Active ...	Enrolment	Nova Scotia Community College Reporting->Enrolment...	2015-09-24	11:25:31	Execute	13212	Success

Figure 11: result of the script ran on `dbo.COGIF_RUNREPORT`

next step will be to start working on combining all three(3) step into a S.Q.L Sever Analysis Service Solution.This process will be done in a different environment, `SQL Server Data Tools for Visual Studio`, more in the next chapter.

Load

SQL Server Data Tools (SSDT)

SSDT is a modern development tool that you can download for free to build SQL Server relational databases, Azure SQL databases, Integration Services packages, Analysis Services data models, and Reporting Services reports. With SSDT, you can design and deploy any SQL Server content type with the same ease as you would develop an application in Visual Studio.

Creating a SSIS Package

- `File`$\rightarrow$ *New* $\rightarrow$ *Project* $\rightarrow$ *IntegrationServiceProject* $\rightarrow$ *OK*

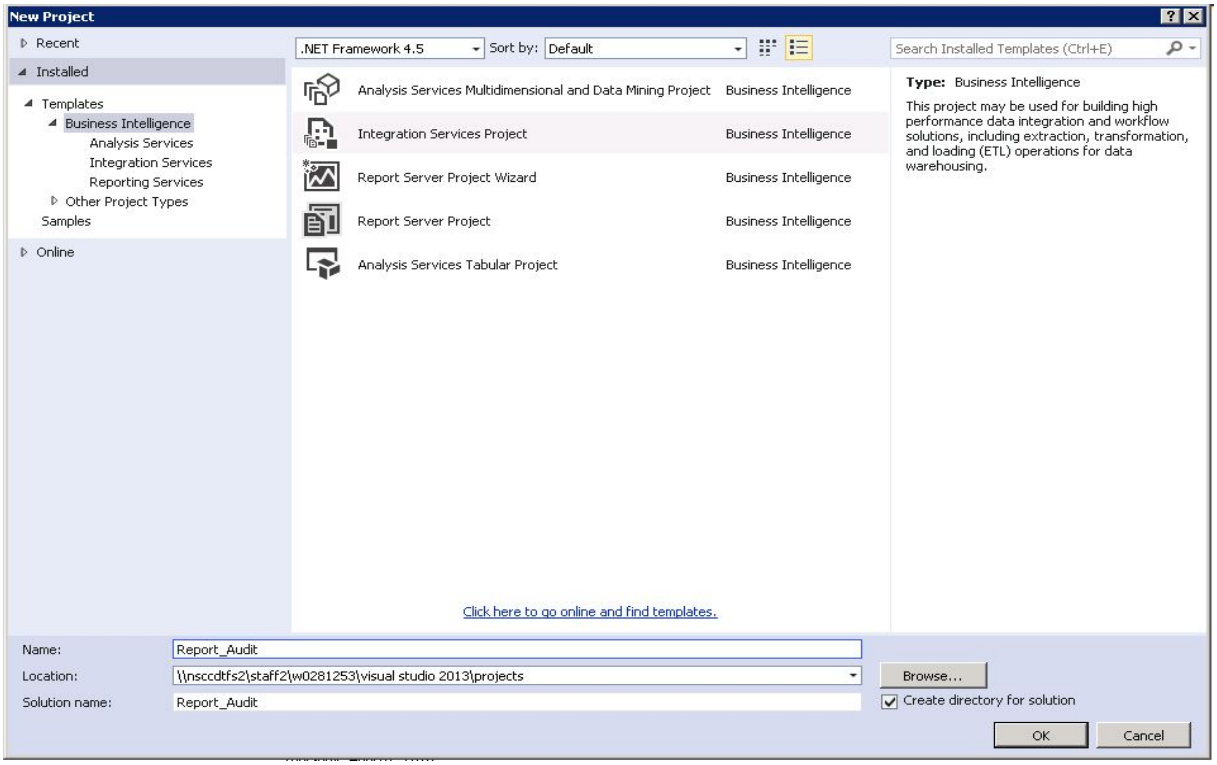


Figure 12: Creating a new Integration Service Project called `Report_Audit`

- Once the project has been created, a default package is created called `Package.dtsx`.You can either choose to rename the package by right clicking on the package and selecting rename or just Delete `Package.dtsx`. If you choose to rename the package skip the next step.

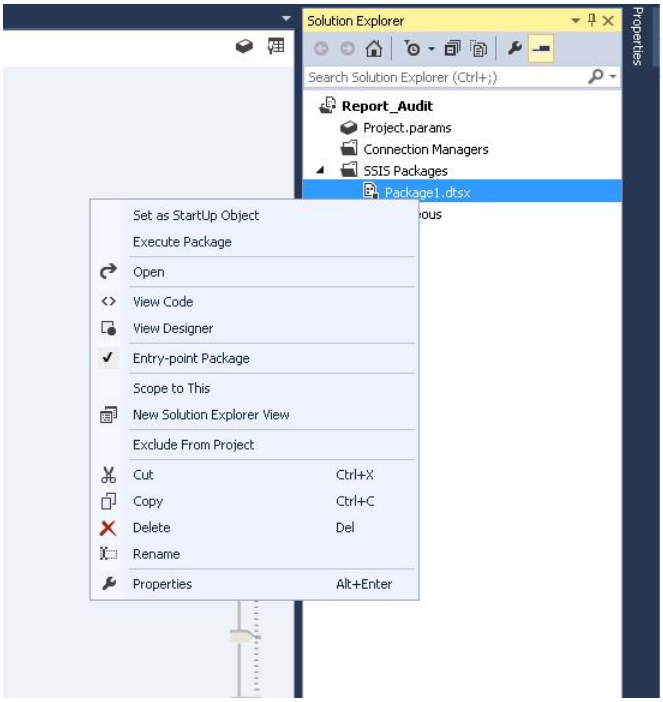


Figure 13: Rename Package to `DimPerson.dtsx`

- Right-Click on the `SSIS Packages` in the Solution Explorer Tab → `New SSIS Package`

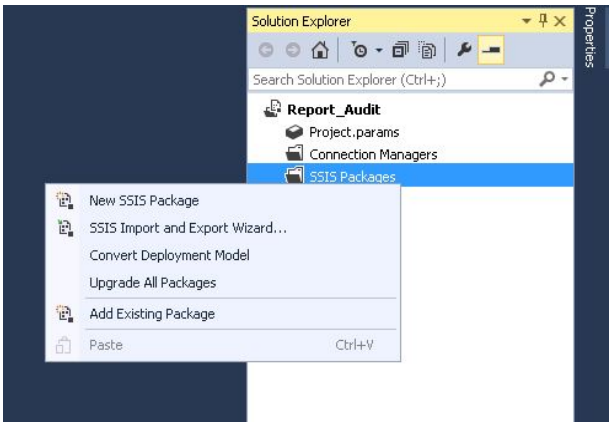


Figure 14: Creating a New SSIS Package called `DimPerson.dtsx`

Setting up Connections

The DimPerson package and all other packages form here on require connections to send and receive data.To begin setting up the connections for DimPerosn:

- right click on the connection manager tab and click on New OLE DB Connection

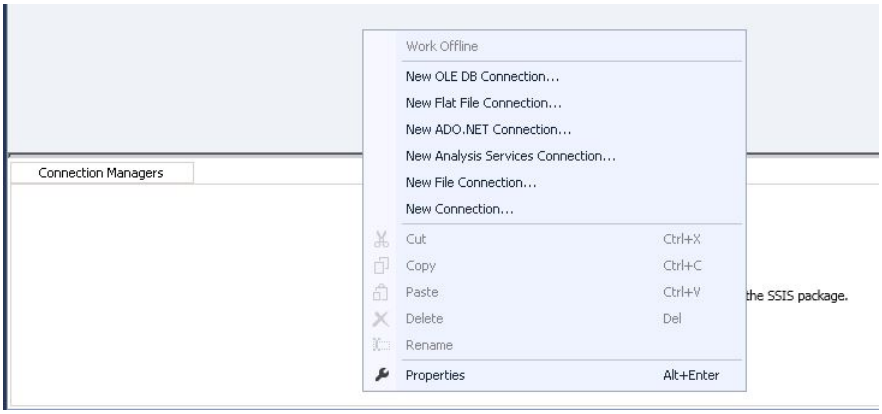


Figure 15: Creating a connection

- The OLE DB Connection Manger window opens. Click on New

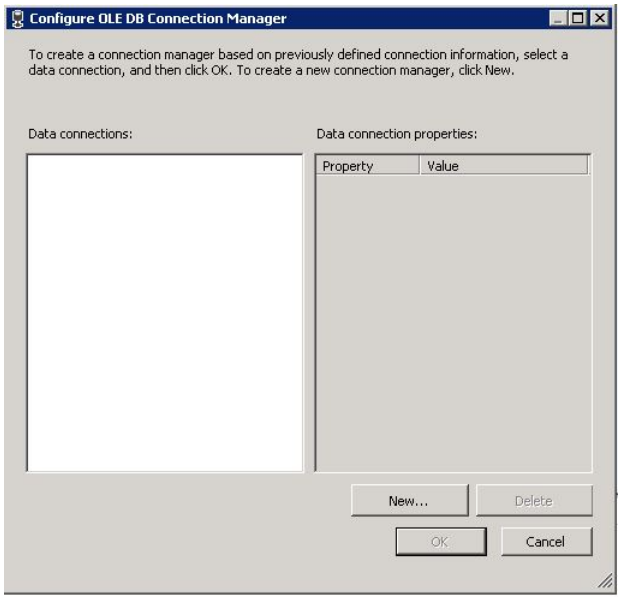


Figure 16: OLE DB Connection Manager

- The next Window will ask for a server name and which database on the server is the connection to.

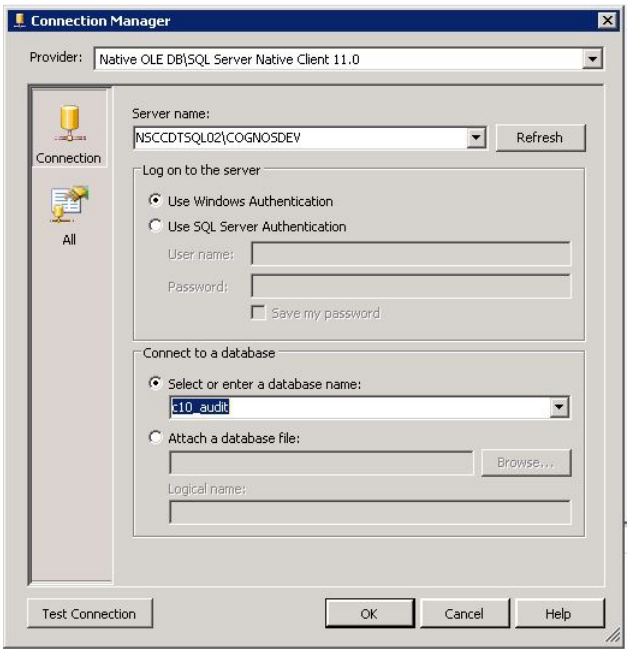


Figure 17: Connection Manager

- After both are entered in click on test connection

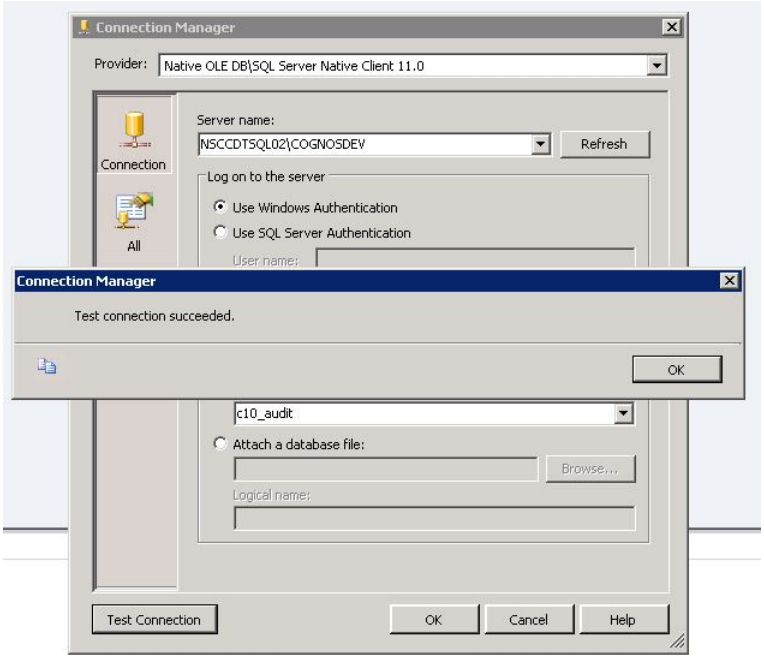


Figure 18: Connection Manager

- It is good practice to rename the connection, giving it a suitable name. To do this right-click on the new connection you created and click on rename

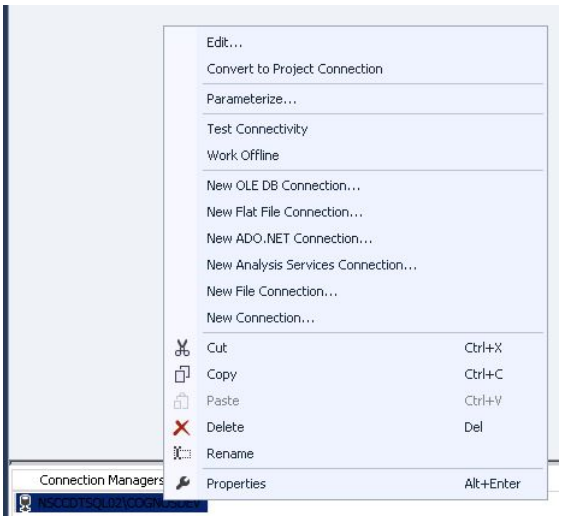


Figure 19: Renaming a connection

- The same process is repeated to set up a connection for the destination. resulting in a connection manager tab with 2 connections. a connection each for the data source and destination.

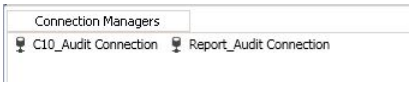


Figure 20: Data Source Connection Destination Connection

DimPerson Package

- In this packages Control Flow Tab, a date flow task is added simply by dragging and dropping a `Data Flow Task` from the toolbox and renaming it to a desired name.

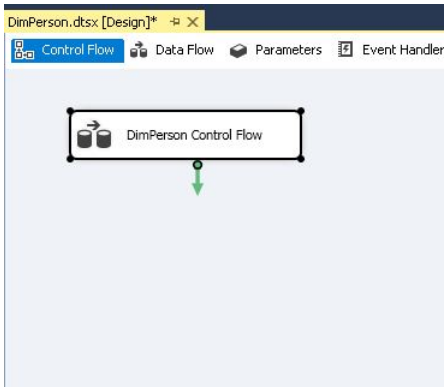


Figure 21: DimPerson Control Flow Tab

- Double clicking on the data flow task automatically takes you to the Data Flow Tab

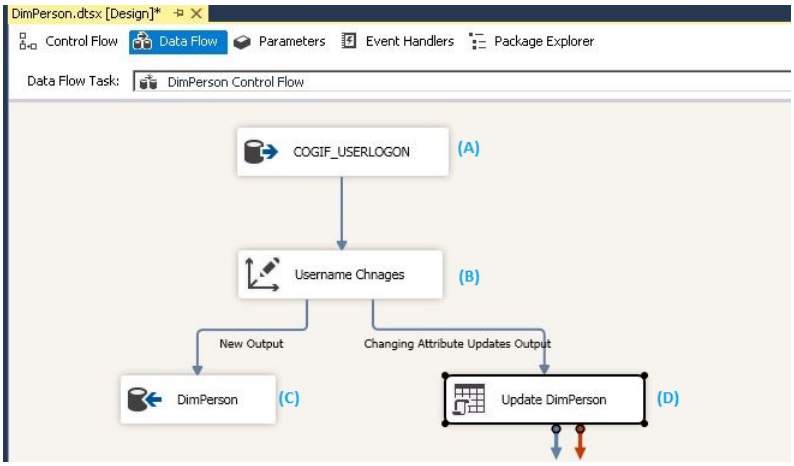


Figure 22: DimPerson Data Flow. The Flow of data from source 'A' to Destination 'C'

- 'A' is an `OLE DB Source` and has been renamed to match the table name from the source

warehouse. Double Clicking on it reveals an `OLE DB Source Editor` 'A' is set the connection

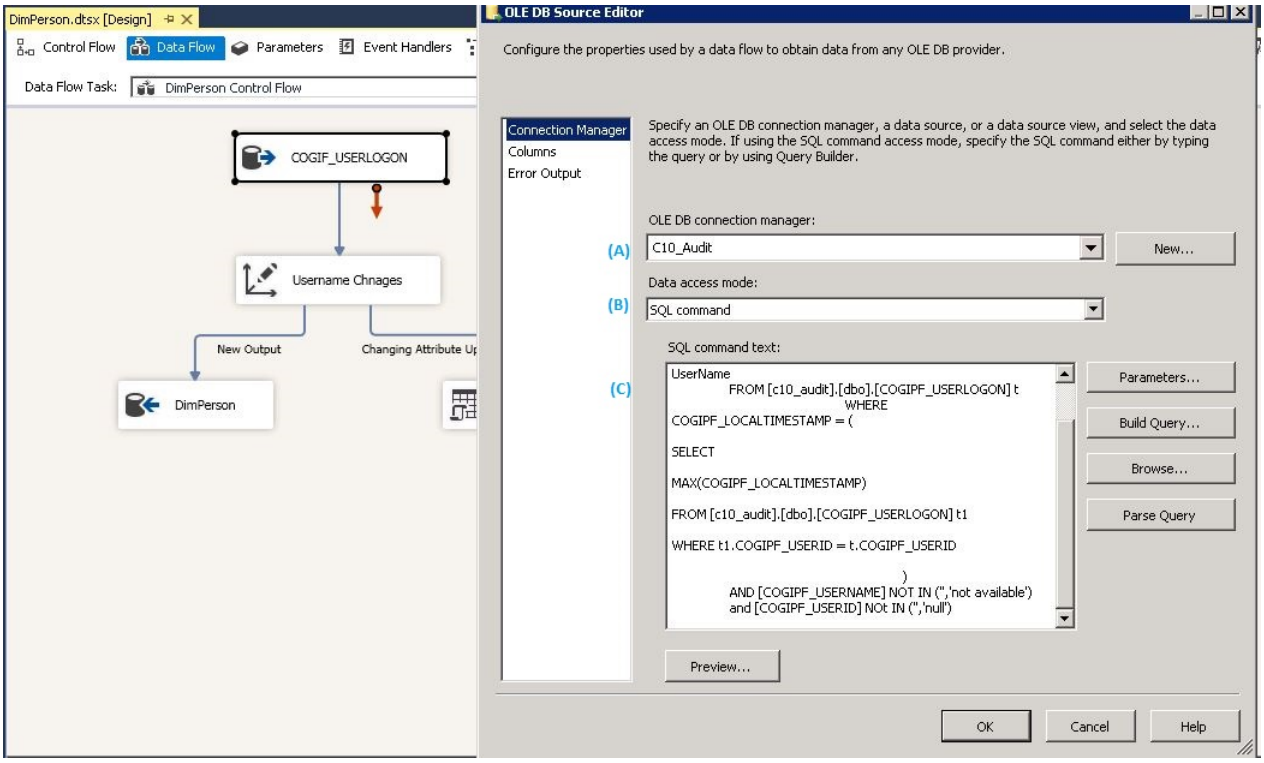


Figure 23: OLE DB Source

we created for our source in the connection manager tab. 'B' is Set to 'SQL Command' because we are using the sql command we wrote in the transform stage of the project for DimPerson, which is place in 'C'.Click OK

- Back to Figure 16 where 'B' is a `Slowing changing Dimension called Username Changes`.The username Column in our DimPerson Dimension table can change with an update to the users account. This change is not frequent and it is also unpredictable when a change like that will happen. The slowly changing dimension 'B' is set up to so : if a new user logs on to the system for the first time and the package is running the new user record will flow down on

the data flow to `New Output` 'C'. If an update happens to the users' Username then it flows down to D where and sql task executes an update statement in the Destionation table for that user.

DimReport Package

- The Report Package has a Data Flow Task created and renamed DimReport. Double clicking on the data flow task will open the data flow tab.

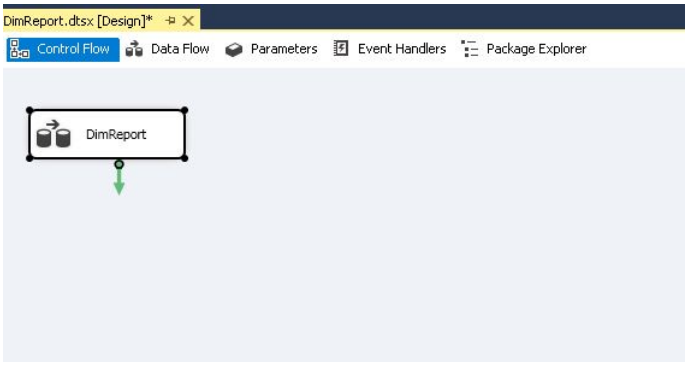


Figure 24: DimReport Control Flow

- The data flow for DimReport is Relatively simple we have a source and destination on the top and bottom respectively with a report `LookUp`

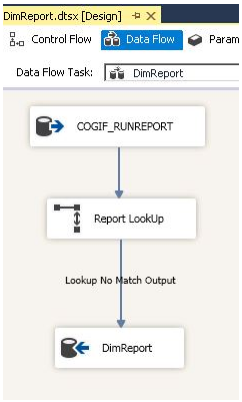


Figure 25: DimReport Data Flow

- The Data source for DimReport is set up the same way we did on DimPerson. We use the

connection to the data source, set data access mode to sql command with the SQl query we build in our extract and transform phase for DimReport.

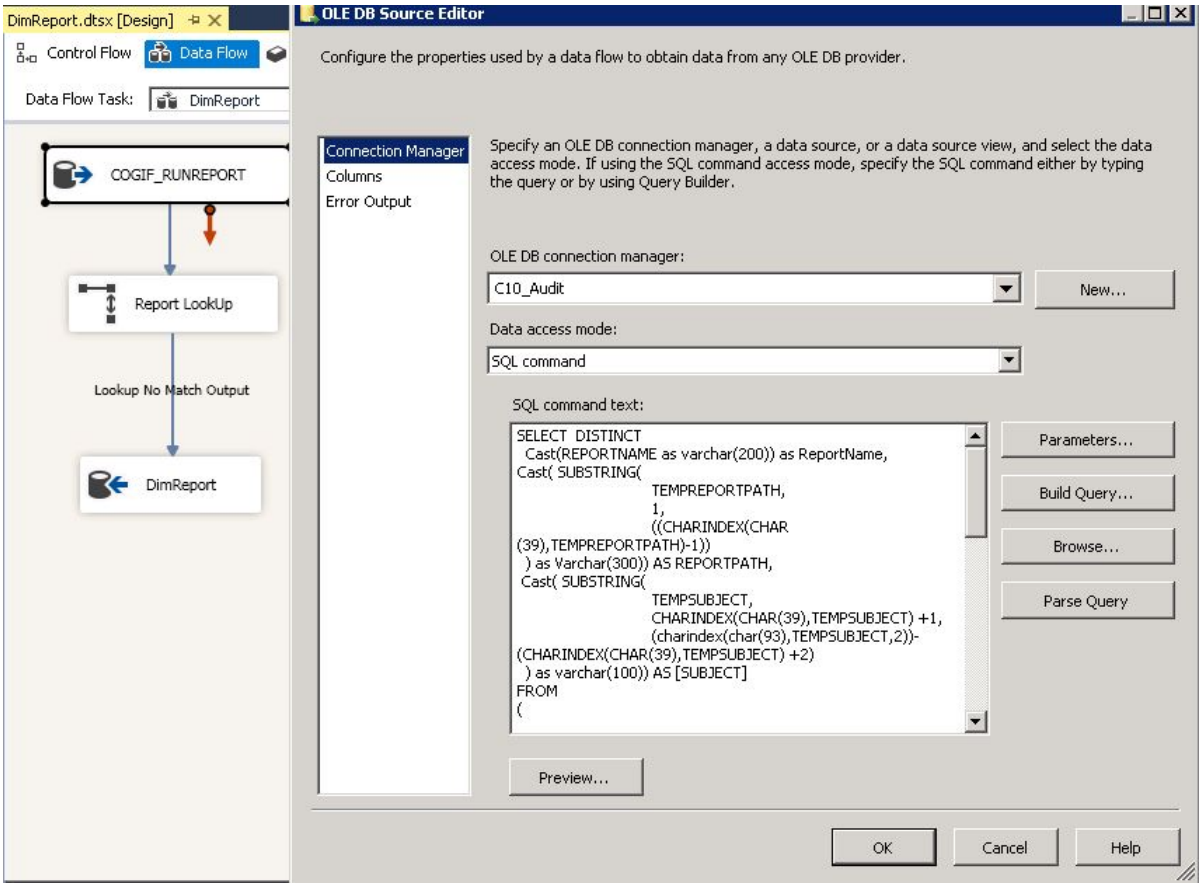


Figure 26: DimReport Data Source Configuration

- The Report LookUp is a sort of data validation or matching to ensure data integrity.This look up will match the data source rows with the data in destination rows. If a row in the data source does not exist in the destination, it is inserted into the destination.

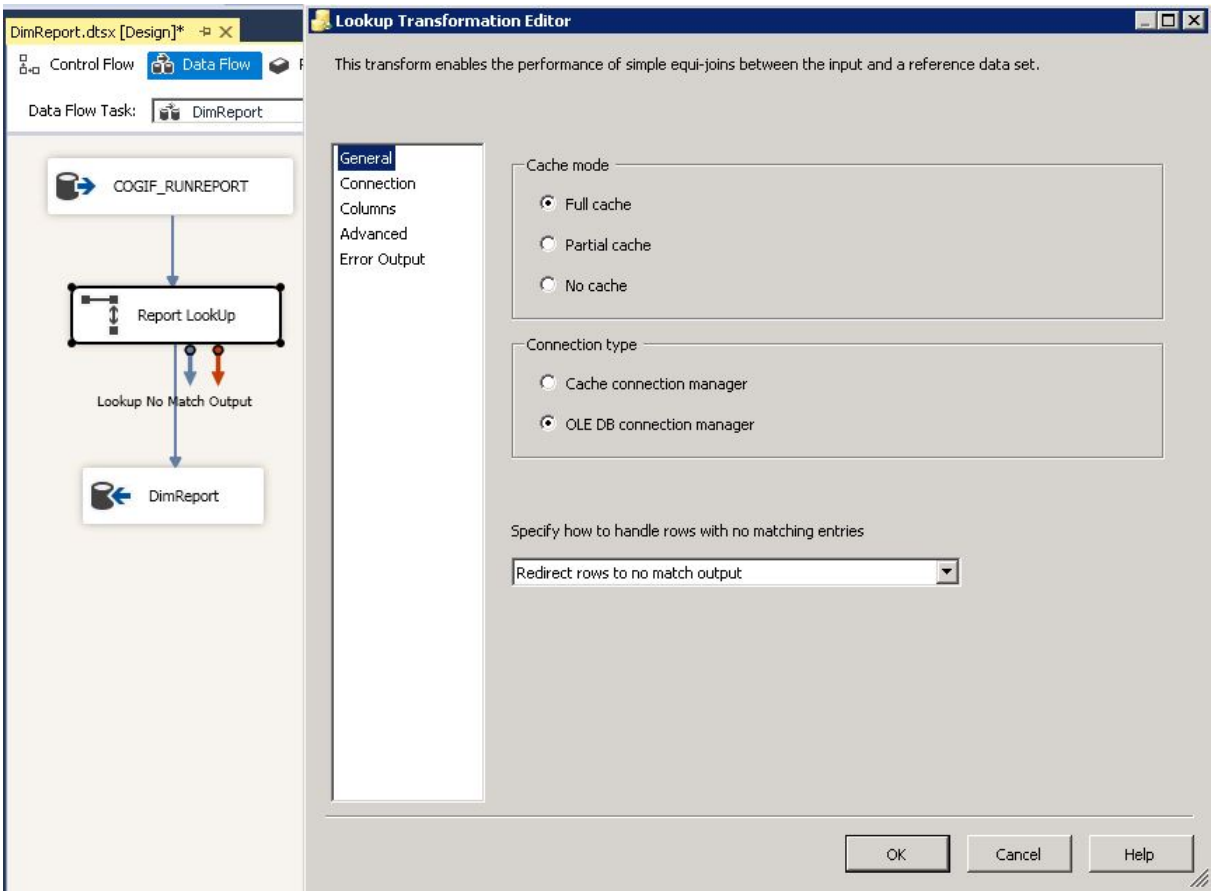


Figure 27: Lookup Transformation Editor Configuration

The "Specify how to handle rows with not matching entries" combobox allows you to select what happen row that don't match. In this case we select "Redirect rows to no match output" so all entries that don't match are sent down the data flow to the destination as new entries.To continue click on the connection link on the left panel.

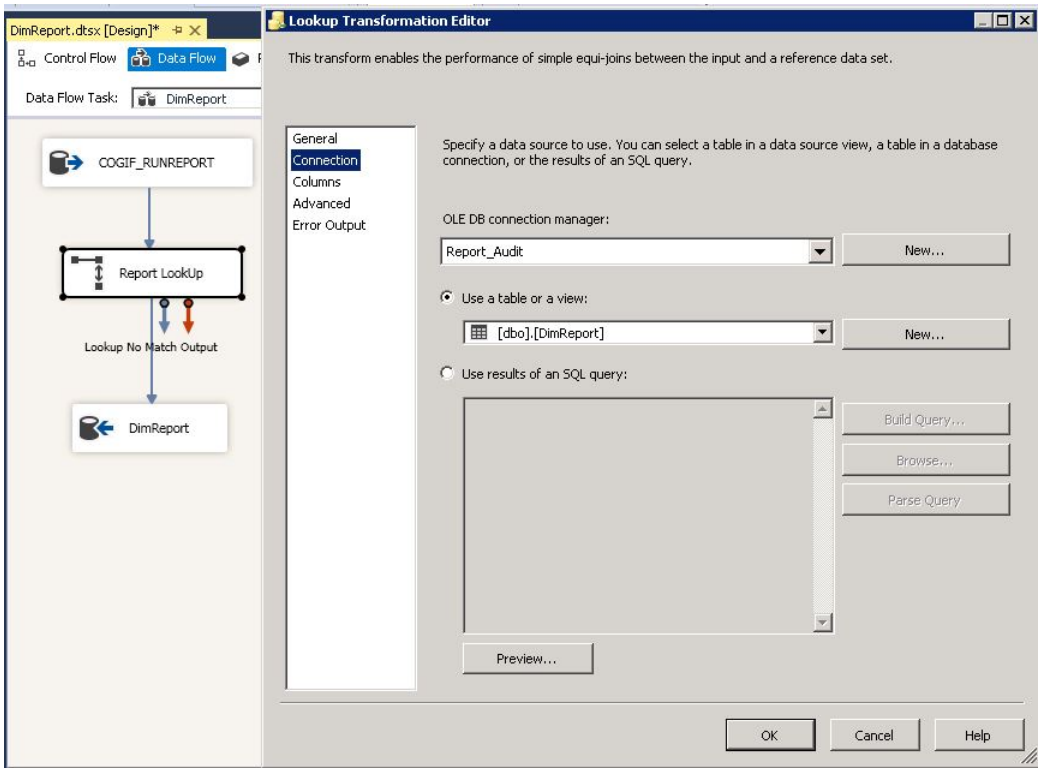


Figure 28: Lookup Transformation Editor Configuration , Connection

Next click on the mapping link in the left panel.

- Click and drag the corresponding field from the "Avaibable Input Column" to the "Available Destination Columns" windows. Click on okay and the Set up for our lookup tranformation is complete.

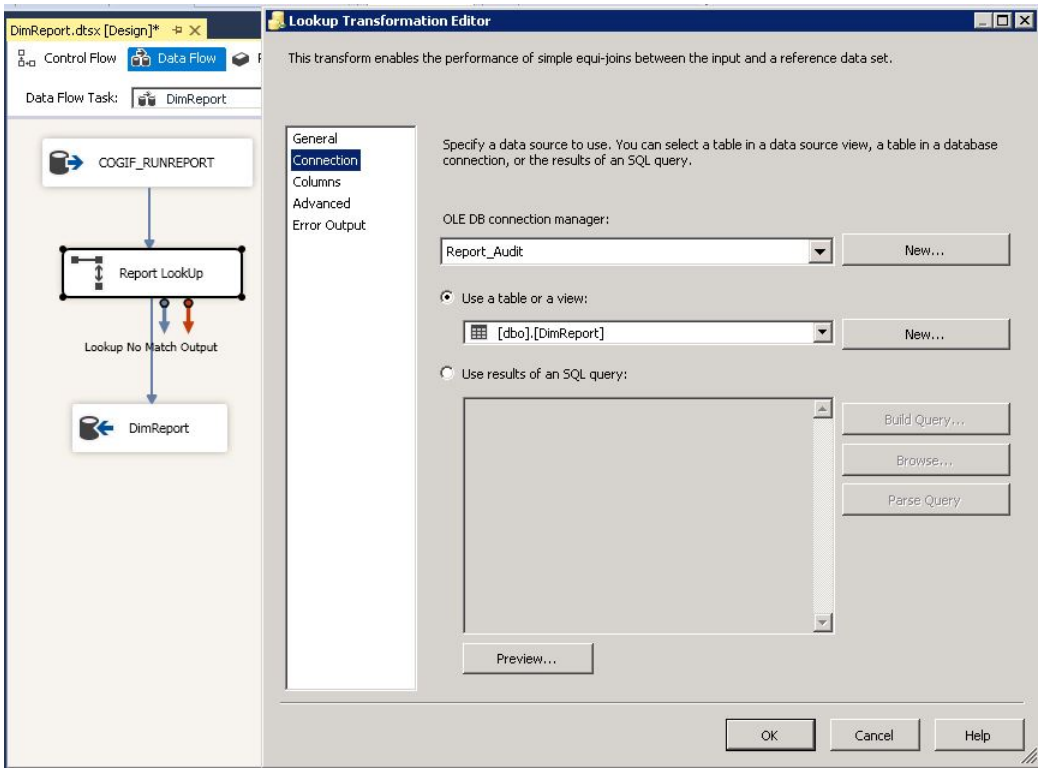


Figure 29: Lookup Transformation Editor Configuration , Mapping

drag the flow link to drop it on the destination DimReport and a window pops up. Select "Lookup NO Match Ouput"

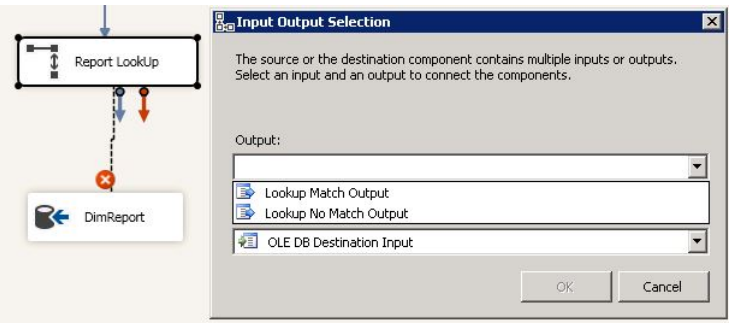


Figure 30: Input Output Selection

- In the final step we set up the destination and make sure our data mappings are accurate.

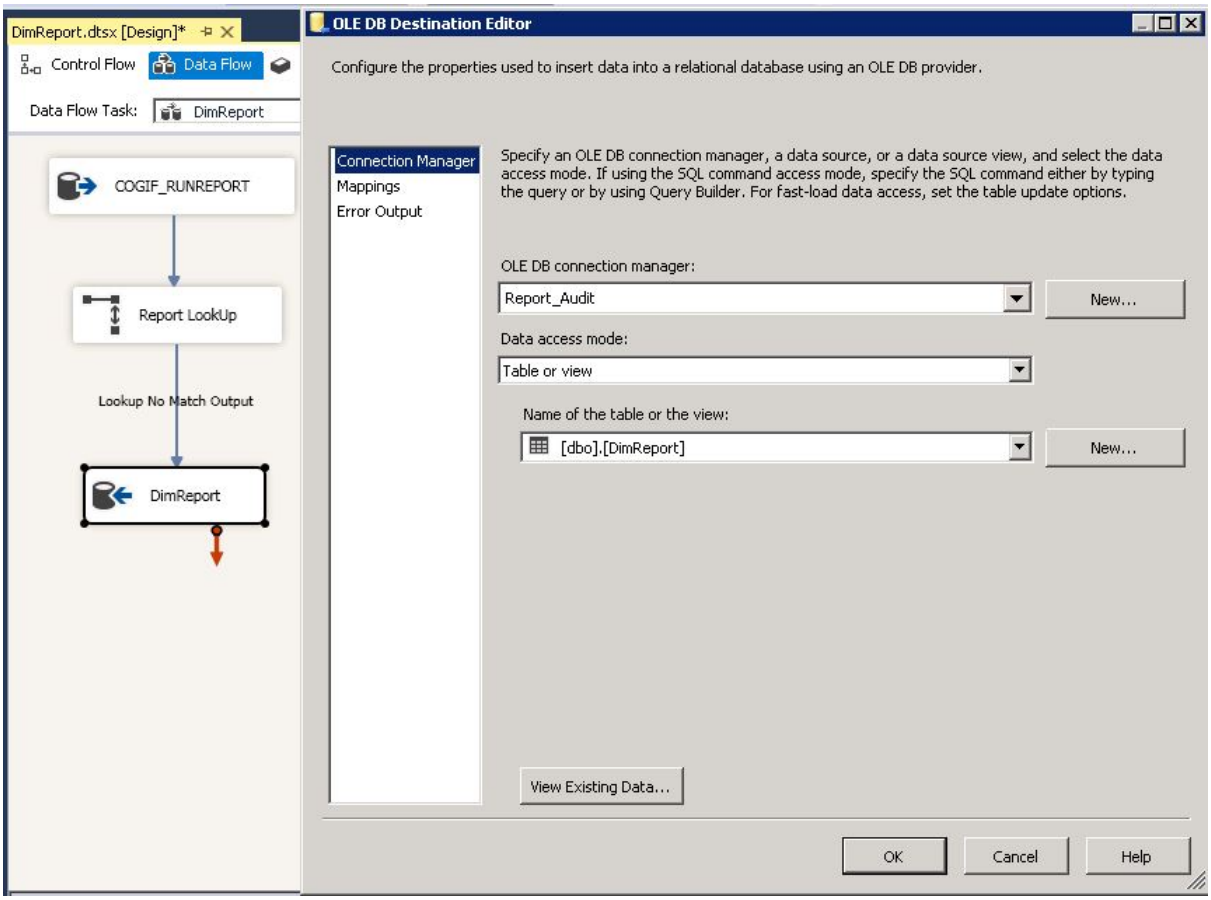


Figure 31: DimReport Destination Connection

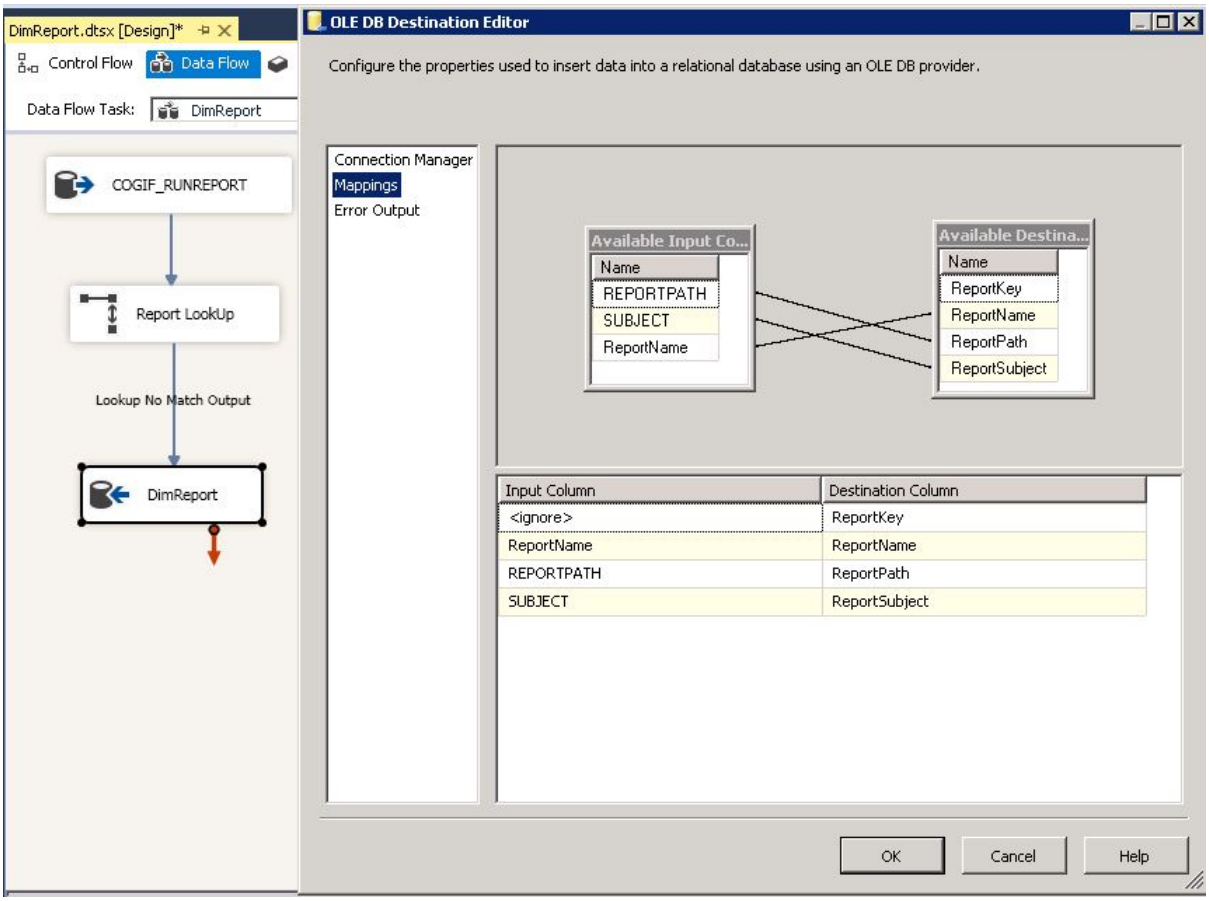


Figure 32: DimReport Destination Mapping

FactReportRun

FactRunReport Control Flow

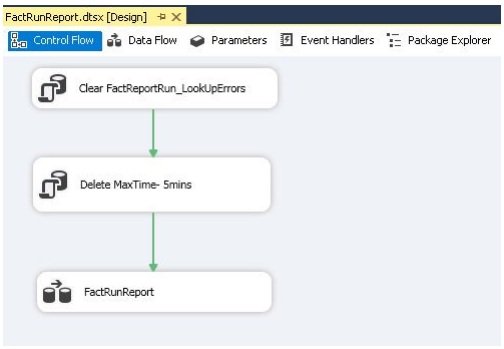


Figure 33: FactRunReport Contol Flow

The Control flow for FactRunReport Package is set up to first clear the warehouse table that stored the errors from the last run, then run the data flow task for FactRunReport shown below. The Data Flow task for FactRunReport is as follows:

- Extract and transform the data from the source
- Does a Lookup against DimPerson using USERID ,return back the PersonKey for each Person. If the userid does not exist the row show error out to `FactRunReport_LookUpErrors`
- Does a Lookup against DimReport using the report name, report subject and report path. it returns back the ReportKey. Rows with measures of the report that don't match error out.
- The Same is done for the DimDate and DimTime Dimensions.
- the Final is is to have all the matched output results to flow down to the destination and the destination is populated.

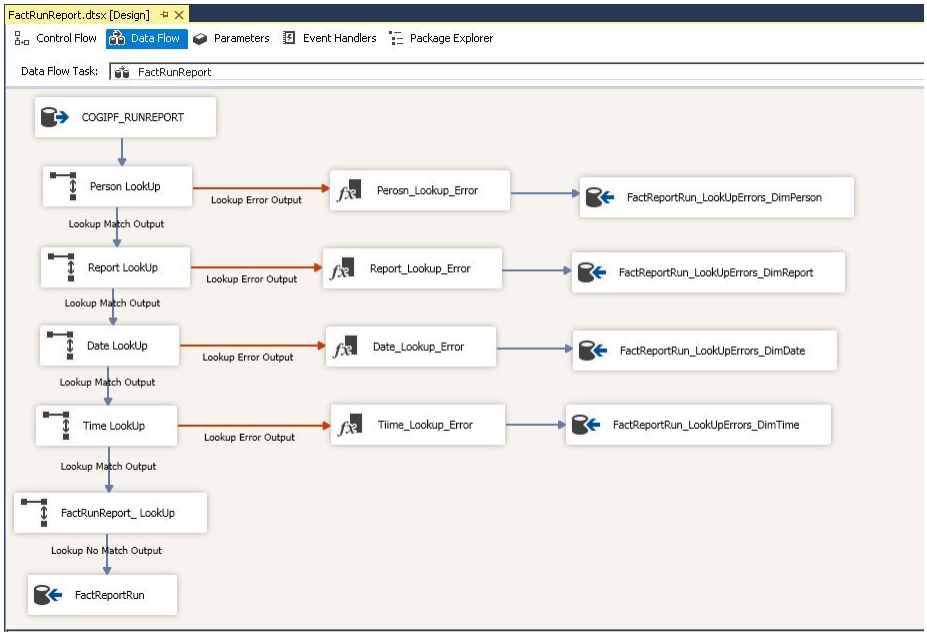


Figure 34: FactRunReport Data Flow