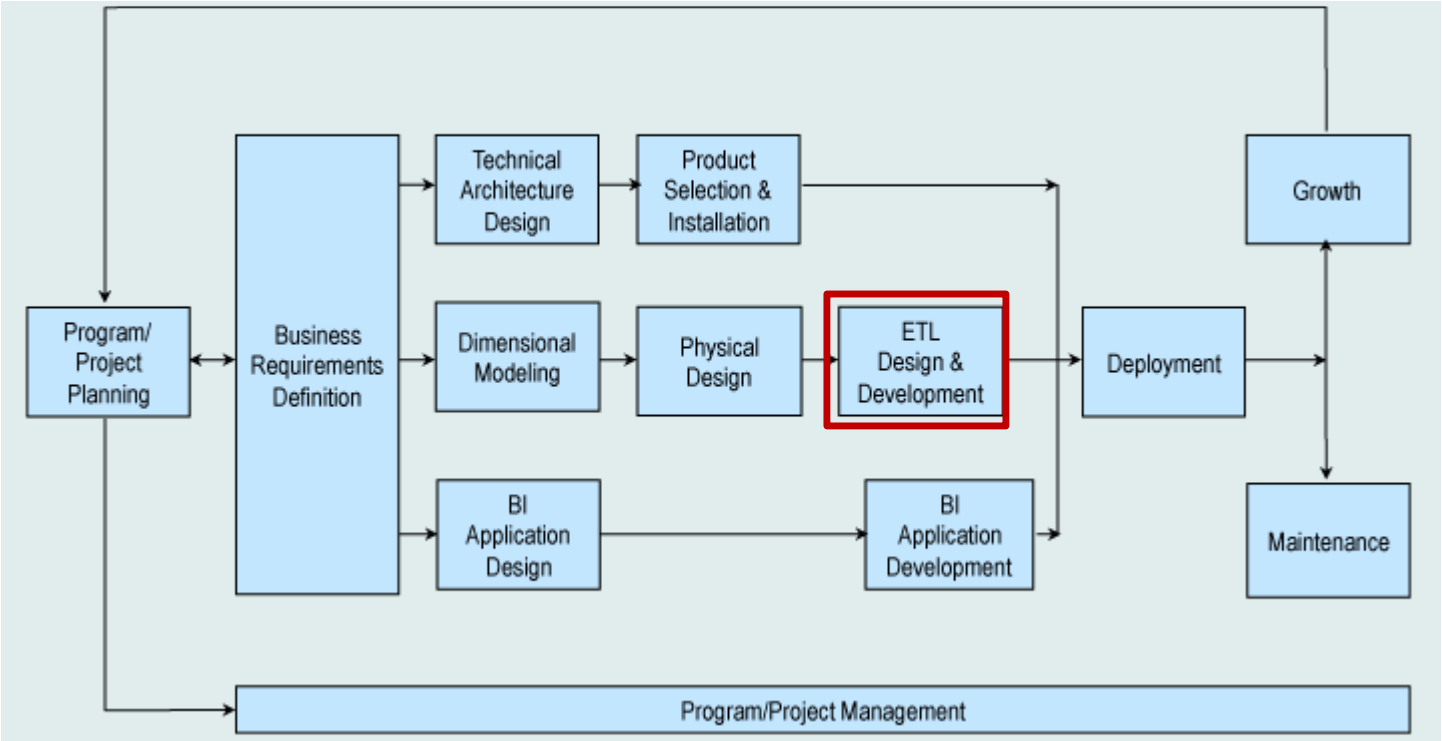
Introduction

School of Information Studies
Syracuse University

Agenda

- Explain the ETL development process
- Demonstrate an ETL tool: SQL Server Integration Services
- Walk through a demonstration:
  - Fudgemart employee timesheets
  - Source to stage
  - Loading dimensions
  - Loading the fact table
  - Putting it all together

Recall: Kimball Lifecycle



ETL Tooling

School of Information Studies
Syracuse University

ETL Again

- ETL stands for extract, transform, load.
- It's the process of:
  - Retrieving data from the OLTP sources,
  - Transforming it, then
  - Placing it into the data warehouse.
- According to Kimball, ETL is a time-consuming process, consuming up to 70% of your data warehousing effort.
- ETL is code but is not typically written in code. We use tooling to write the code for us.

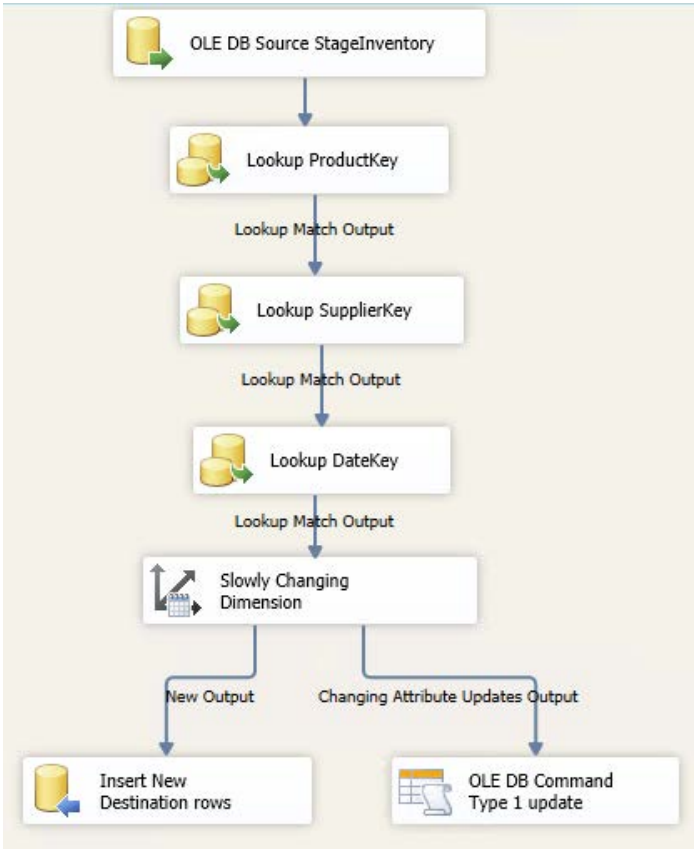
ETL Tools

- You could code ETL in Python or Java, but why?
- ETL tooling is a popular choice today.
- All the DBMS vendors offer tools.
- Tooling not required but aids the process greatly.
- Tooling is visual and self-documenting.
- It's SCM friendly.
- Useable by nonprogrammer.
- The only downside is cost!

Products

- Informatica DI
- IBM DataStage
- Oracle Data Integrator
- SAP Data Services
- Microsoft SSIS
- Pentaho Kettle
- SnapLogic
- AWS Glue

ETL Tool vs. Programming



Which of these is easier to understand?

Which is self-documenting?

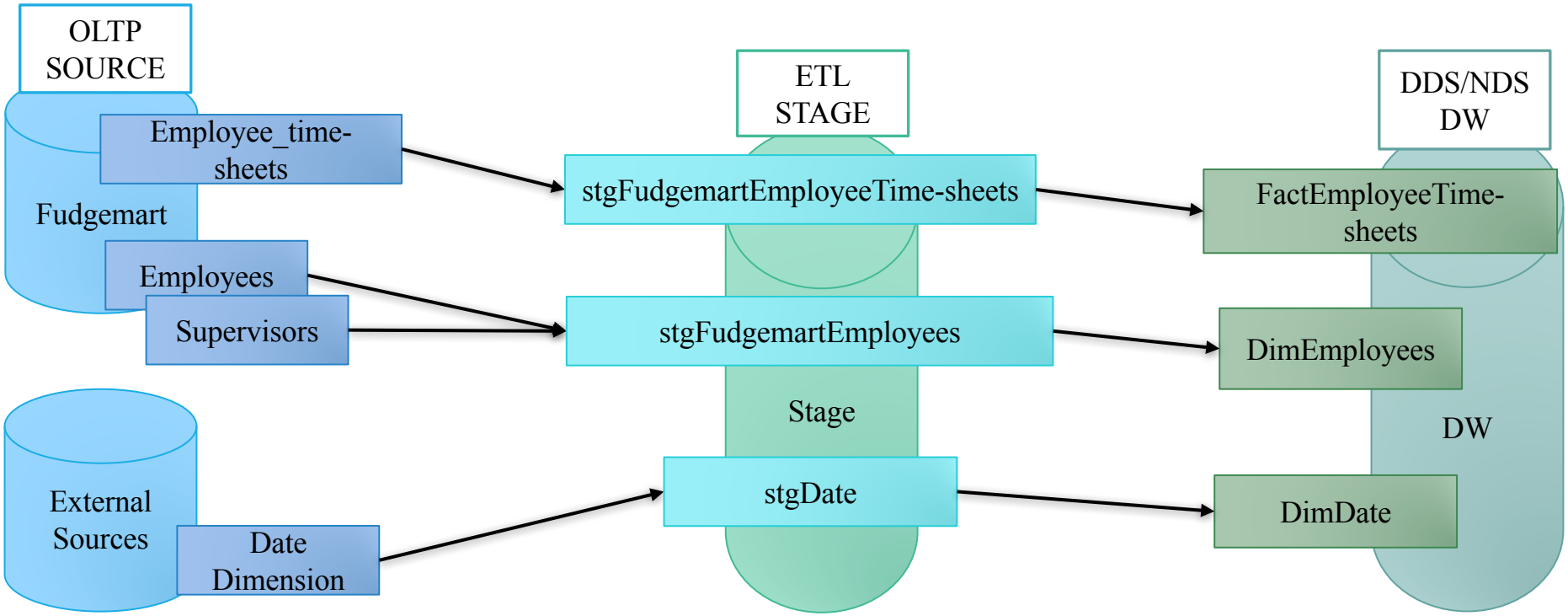
Which is SCM friendly?

```
46
47  ) AS
48  BEGIN
49      -- SET NOCOUNT ON added to prevent extra result sets from
50      -- interfering with SELECT statements.
51      SET NOCOUNT ON;
52      declare @id varchar(10)
53      declare @courseId varchar(10)
54      declare @keyid int
55      set @id = @term + '.' + @classNumber
56      set @courseId = @courseSubj + @courseNum
57      if (UPPER(@component) = 'LAB') set @courseTitle = (select 'LAB: ' + @courseTitle)
58
59      -- Courses table
60      if not exists(select * from dbo.Courses where
61            courseId=@courseId and courseTitle=@courseTitle)
62      begin
63         insert into dbo.Courses (courseId, courseTitle) values (@courseId, @courseTitle)
64      end
65
66      set @keyid = (select distinct keyId from Courses
67         where courseId=@courseId and courseTitle=@courseTitle)
68      --print 'keyid = ' + cast( @keyid as varchar(10))
69      -- Classes table
70      if exists(select * from dbo.Classes where id = @id)
71      begin
72         update dbo.Classes
73            set touched = 1
74            ,lastUpdate = GETDATE()
75         where id = @id
76      end
77      else
78      begin
79         insert into dbo.Classes ( id, scheduleSectionId, lastUpdate, touched)
80         values (@id, 0, GETDATE(), 1)
81      end
82
83      -- provClasses table
84      if exists(select * from dbo.provClasses where id = @id)
85      begin
86         update dbo.provClasses
87            set touched = 1
```

Source to Target Map

School of Information Studies
Syracuse University

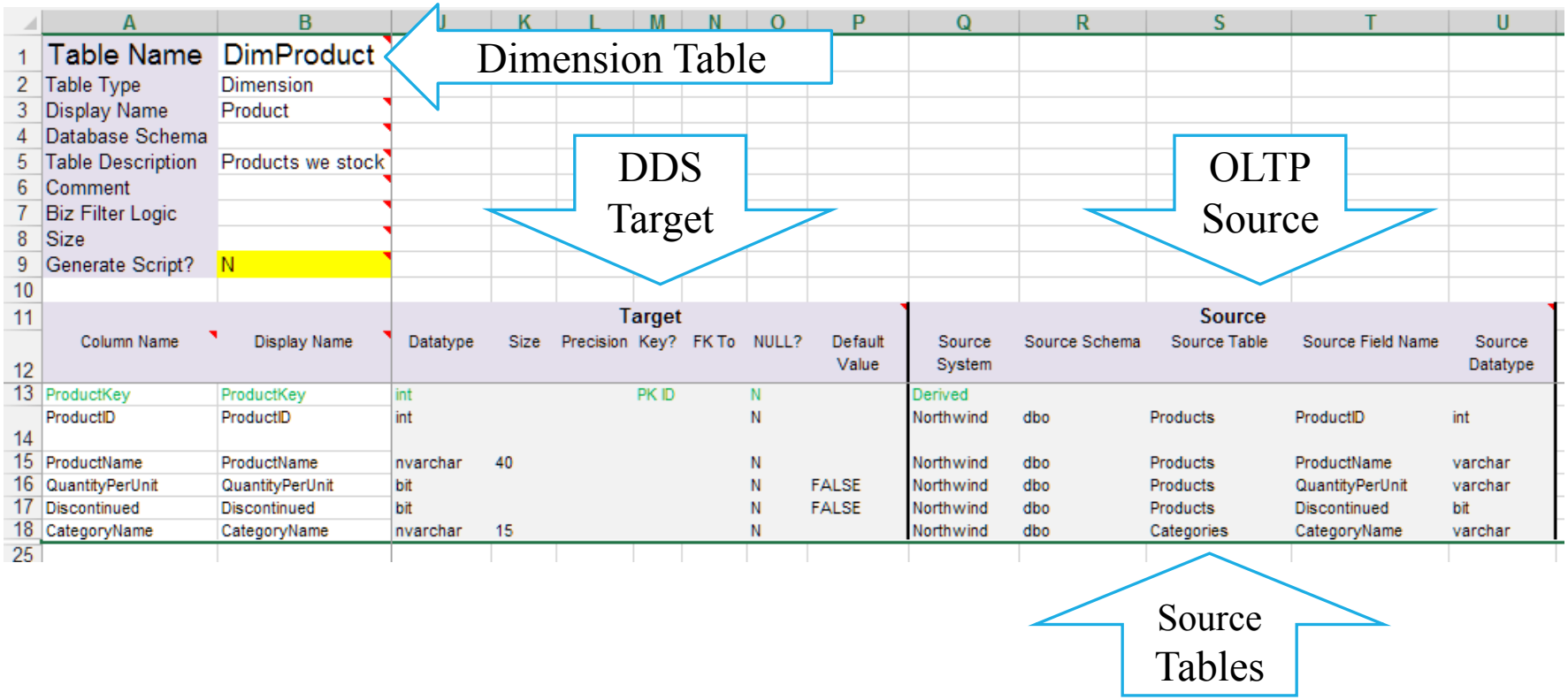
Example: Source to Target Map



ETL is complicated. Never start without a plan!

Example: Sources and Targets

Detailed modeling worksheet should list sources and targets.



	A	B
1	Table Name	DimProduct
2	Table Type	Dimension
3	Display Name	Product
4	Database Schema	
5	Table Description	Products we stock
6	Comment	
7	Biz Filter Logic	
8	Size	
9	Generate Script?	N

Dimension Table

DDS Target

OLTP Source

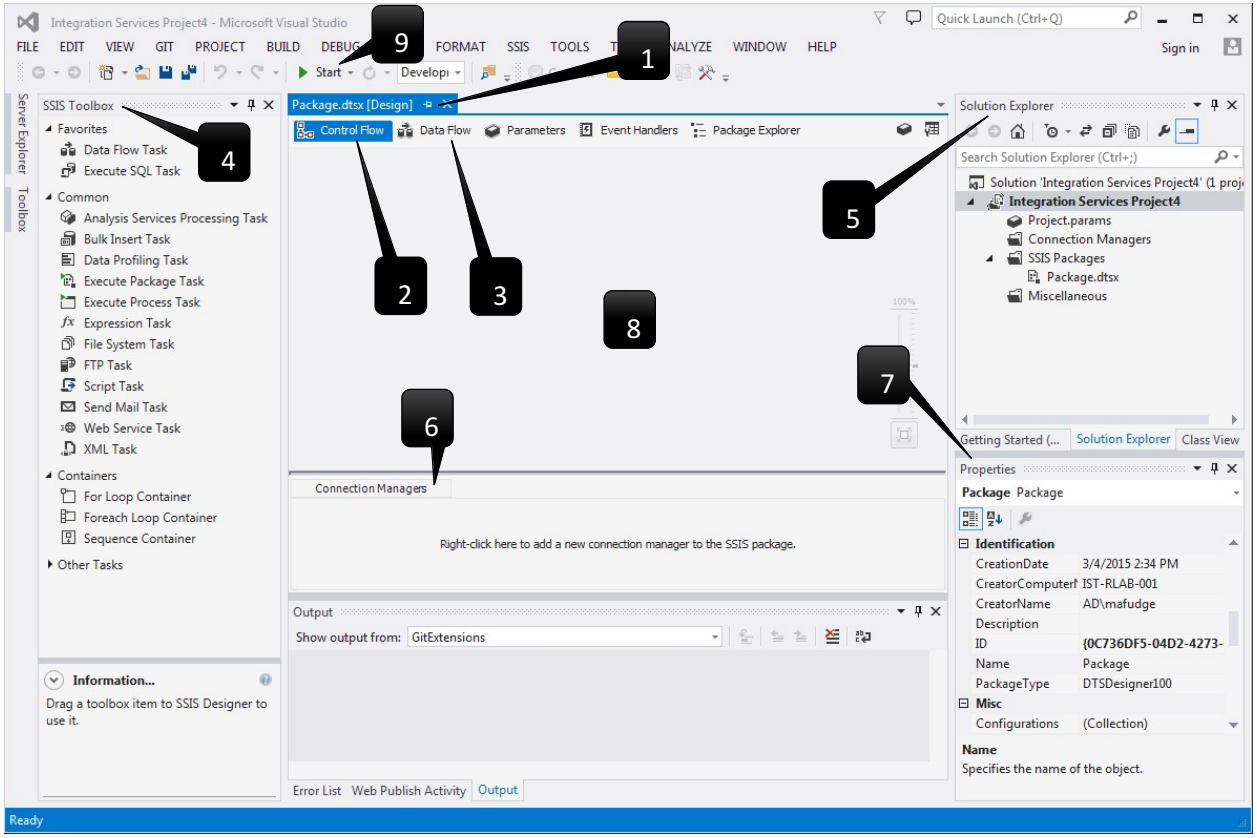
	Target								Source				
12	Column Name	Display Name	Datatype	Size	Precision	Key?	FK To	NULL?	Default Value	Source System	Source Schema	Source Table	Source Field Name	Source Datatype
13	ProductKey	ProductKey	int			PK ID		N		Derived				
14	ProductID	ProductID	int					N		Northwind	dbo	Products	ProductID	int
15	ProductName	ProductName	nvarchar	40				N		Northwind	dbo	Products	ProductName	varchar
16	QuantityPerUnit	QuantityPerUnit	bit					N	FALSE	Northwind	dbo	Products	QuantityPerUnit	varchar
17	Discontinued	Discontinued	bit					N	FALSE	Northwind	dbo	Products	Discontinued	bit
18	CategoryName	CategoryName	nvarchar	15				N		Northwind	dbo	Categories	CategoryName	varchar

Source Tables

Quick Tour

School of Information Studies
Syracuse University

Quick Tour of SSIS



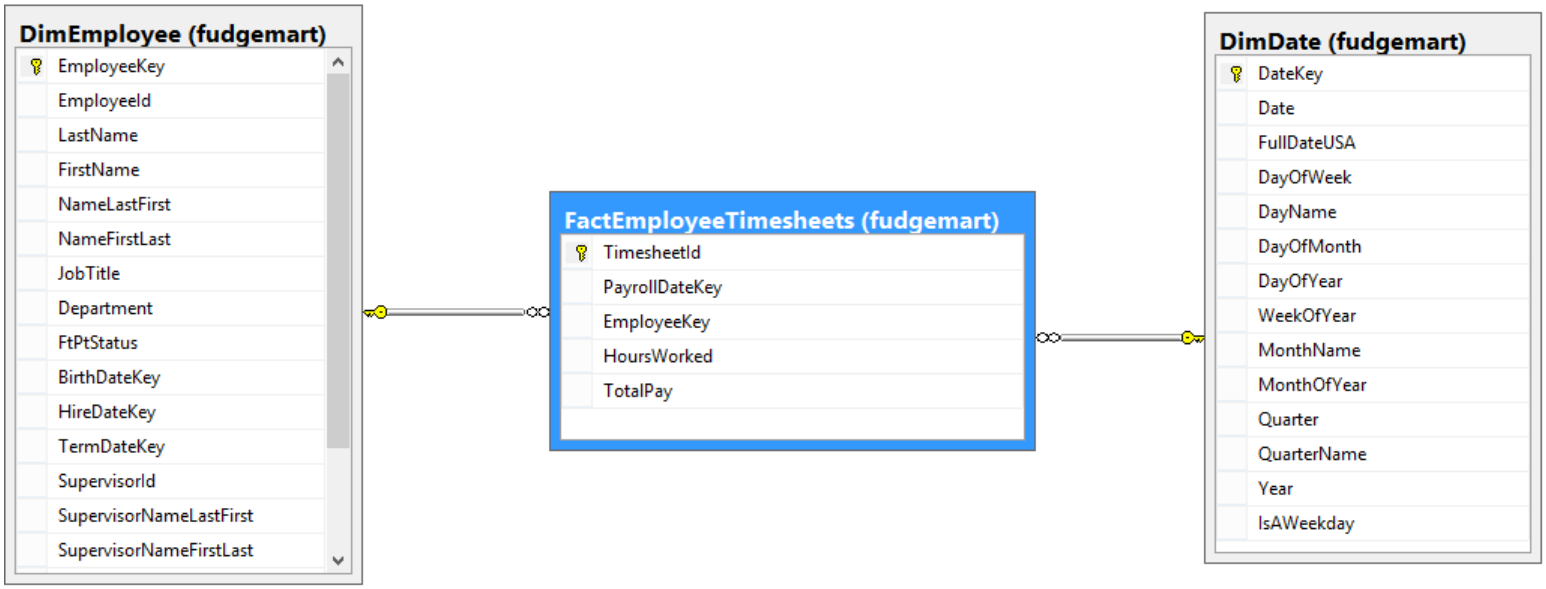
1. Packages
2. Control Flow
3. Data Flow
4. Toolbox
5. Solution Explorer
6. Connection Manager
7. Properties
8. Design Surface
9. Package Execution

Star Schema and Source to Target Map

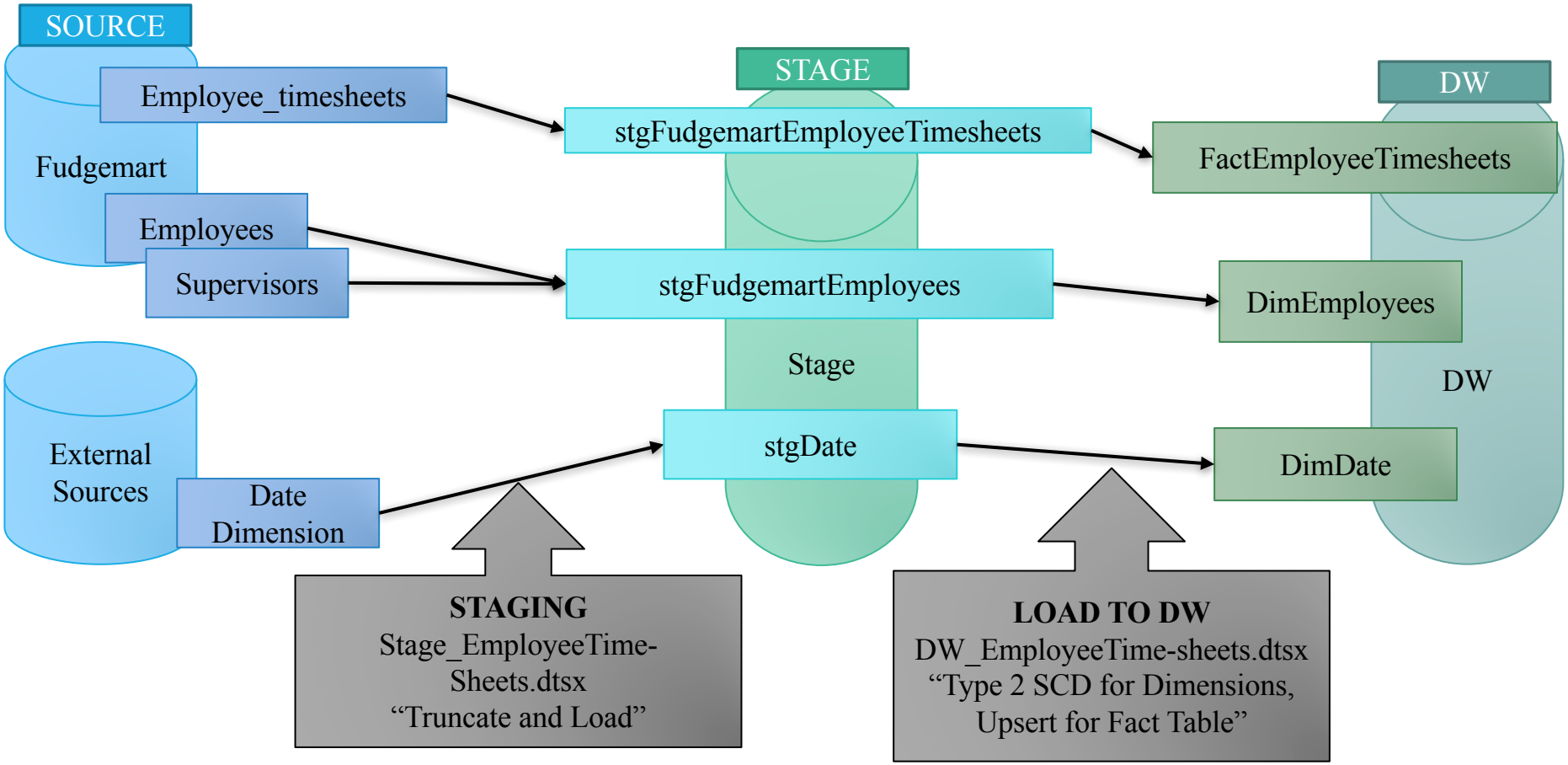
School of Information Studies
Syracuse University

Star Schema

Using the **Fudgemart employee timesheets** dimensional model from before



High-Level Source to Target Map



The ETL Packages at a Glance

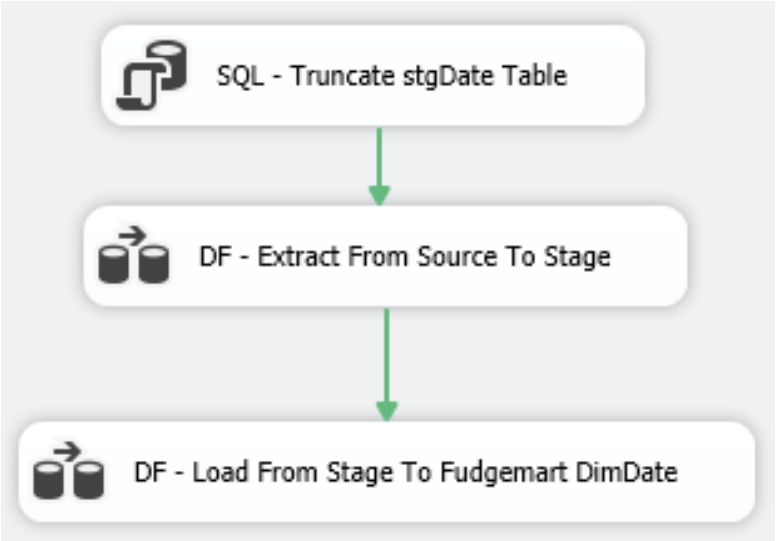
1. ***DateDimensionImport.dtsx***
   - Imports the date dimension (one-time deal).
   - One package to go from source to stage to target.
2. ***Stage_EmployeeTimesheets.dtsx***
   - Stage dimension and fact data as is using the truncate and load pattern.
3. ***DW_EmployeeTimesheets.dtsx***
   - Transform staged data into the required dimensions and facts.
   - Load with Type 2 or 1 SCD pattern, as to not reprocess the same data.
4. ***Package.dtsx***
   - Combine Steps 1 through 3 into one package.

Prebuilt Package- Date Dimension

School of Information Studies
Syracuse University

1. DateDimensionImport.dtsx

- We will walk through how it works.
- We'll skip making it for the sake of time!



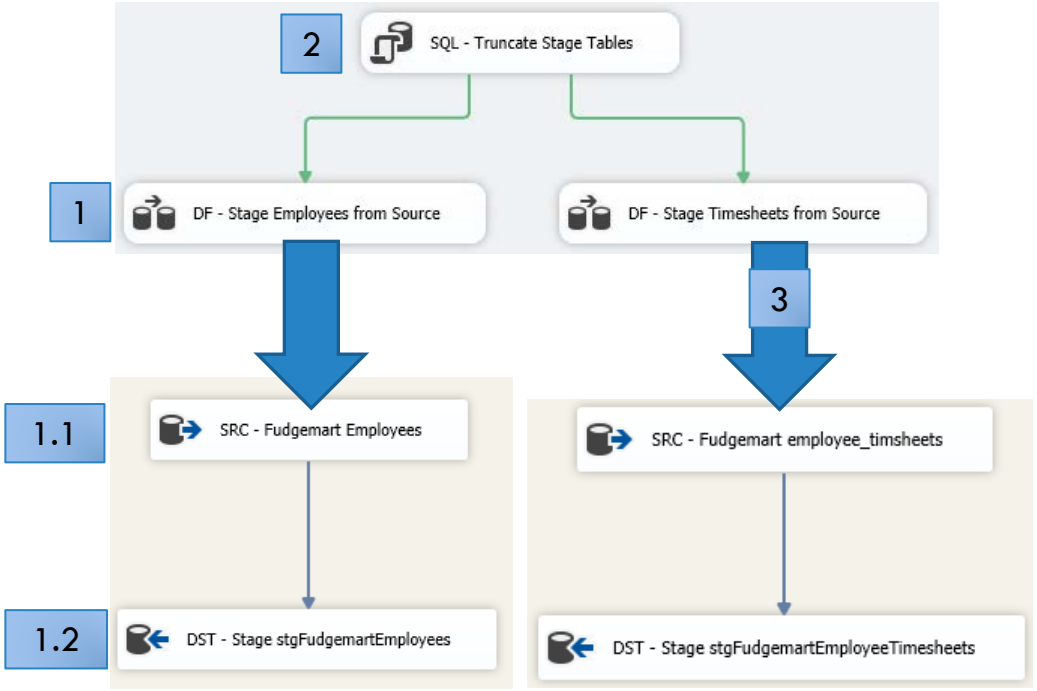
The Plan: Staging Data

School of Information Studies
Syracuse University

2. Stage_EmployeeTimesheets.dtsx

Staging process for truncate and load:

1. Data flow: from source to stage
   1. Source: Use a SQL command to match target attributes.
   2. Target: Create new staged table and import data as is.
2. Include a SQL task to truncate the table before import.
3. Repeat for each source to stage.



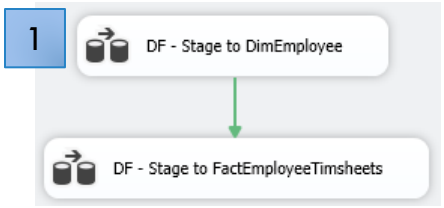
SSIS: Stage Fudgemart Timesheet Data

1. Stage employees.
2. SQL task to truncate to complete truncate and load.
3. Repeat for employee timesheets.

The Plan: Loading Dimensions

School of Information Studies
Syracuse University

3. DW_EmployeeTimesheets.dtsx

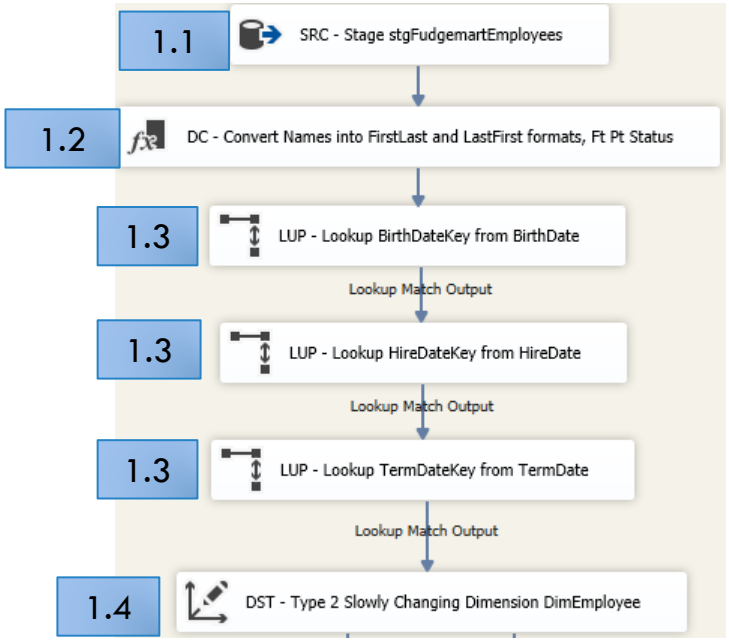


Type 2 SCD processing of **DimEmployee**

1. Data flow from stage to dimension
   1. Load data source from stage.
   2. Transform data from source to match target.
   3. Look up surrogate key pipeline.
   4. Process changes using SCD Type 2.

Repeat these steps for each dimension

Steps 1.2 and 1.3 will vary based on the data source and need of the dimension.



SSIS Demo: DW Fudgemart Timesheet Employee Dimension Processing

1. Create dataflow.
2. Add data conversions.
3. Tangent: the data viewer.
4. Look up transformations.
5. SCD wizard.

The Plan: Loading the Fact Table

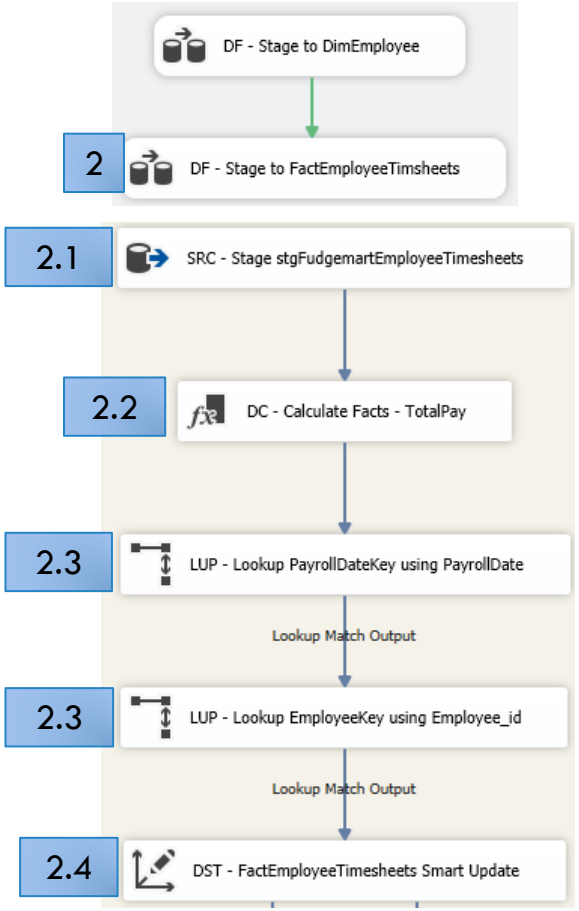
School of Information Studies
Syracuse University

3. DW_EmployeeTimesheets.dtsx

Upsert processing of **FactEmployeeTimesheets**.

2. Data flow from stage to fact
    1. Load data source from stage.
    2. Transform data from source to match target—calculate facts.
    3. Look up surrogate key pipeline.
    4. Process changes using SCD Type 1 (Upsert).

All steps are required for most fact tables.



SSIS Demo: DW Fudgemart Timesheet Fact Table Processing

1. Create data flow.
2. Fact calculations.
3. Surrogate key pipeline.
4. Loading via Upsert.

The Plan: Combining Packages

School of Information Studies
Syracuse University

4. Package.dtsx

One package to execute the others.

This package would get scheduled to execute on a routine basis.

Production changes:

- No date dimension.
- Do not stage all data, but stage based on last processed.