Delivering a Relational Data Warehouse

Week 4 – Loading and Maintaining a Data Warehouse

Module 12

Exploring Additional
ETL Design Concepts

Microsoft

Module Outline
12 | Exploring Additional ETL Design Concepts

Topic
► Data Warehouse Load Design
► ETL Management
► **Demo:** Developing and Managing an ETL Solution
► **Lab:** Loading and Maintaining a Data Warehouse

Module Outline
12 | Exploring Additional ETL Design Concepts

Topic
Data Warehouse Load Design
ETL Management
Demo: Developing and Managing an ETL Solution
Lab: Loading and Maintaining a Data Warehouse

Data Warehouse Load Design

- Development Methodology
- Populating Dimension Tables
- Populating Fact Tables
- Master Package
- Package Logging
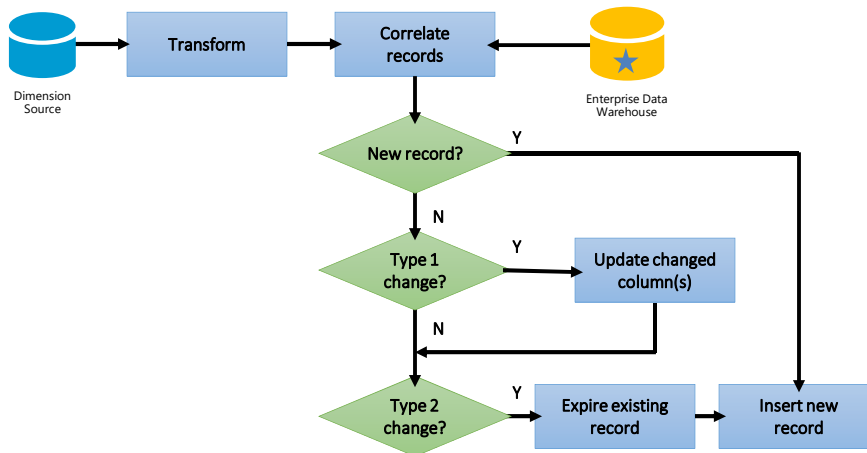- Recommended Practices

Data Warehouse Load Design
Development Methodology

1. Create an **Integration Services Project**
2. Create connection managers
3. Develop packages—typically one per data warehouse table
4. Develop a master package to orchestrate the execution of all packages
5. Optionally, trigger data model processing
6. Deploy the project
7. Configure automated processing of the master package

Data Warehouse Load Design
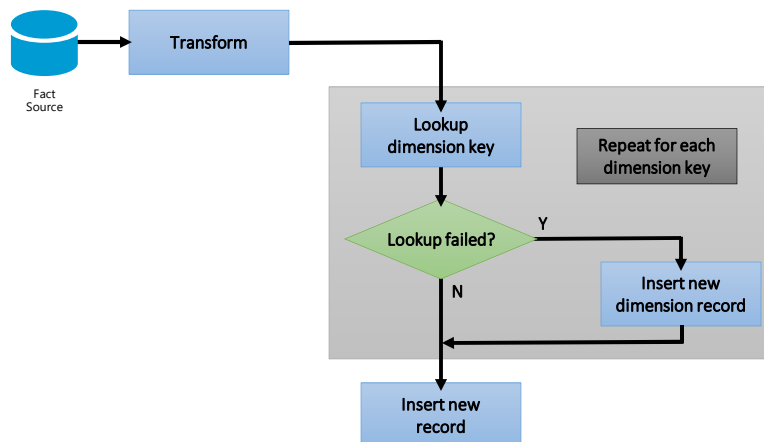Populating Dimension Tables

---

Data Warehouse Load Design
Populating Dimension Tables

- Design dimension packages to incrementally
  load new members, and appropriately change existing members
- A convenient design approach is to use the Slowly Changing
  Dimension transform, however it does not scale well
- To process larger data volumes (>10K rows),
  design a scalable solution by executing set-based
  operations on the dimension table
  – Stay tuned for the demonstration in this module

Extract, Transform and Load
Populating Fact Tables

```
[Fact Source] → [Transform]
                     │
                     ▼
        ┌─────────────────────────────────────┐
        │  [Lookup dimension key]   [Repeat for│
        │          │                 each      │
        │          ▼                dimension  │
        │     <Lookup failed?> ──Y──   key]    │
        │          │                    │      │
        │          N          [Insert new      │
        │          │          dimension record]│
        │          │◄──────────────────┘       │
        └──────────┼──────────────────────────┘
                   ▼
            [Insert new record]
```

Data Warehouse Load Design
Fact Table Loading

- Design fact packages to incrementally
  load new facts
- The packages typically rely heavily on the Lookup component to
  retrieve dimension surrogate keys

  [Lookup]

- Lookup failures can be managed in different ways:
  - Fail the fact load
  - Insert a pre-defined "unknown" member, for example key -1
  - Create a new dimension member (inferred member)

Data Warehouse Load Design
Master Package

- The master package can execute each table package by using the Execute Package task

- It should execute packages in the correct sequence
  - Dimension tables first, followed by fact tables

- Consider parameterizing the package to pass in necessary values at execution time
  - Parameter values can be passed in the Execute Package tasks also

Data Warehouse Load Design
Package Logging

- It is helpful to log details about the package execution, including:
  - Error rows (lookup failures)
  - Row counts, for example, new or changed members
  - Processing duration
  - Warning or error messages to convey to an administrator

- It is common to accumulate significant volumes of logging data, and this becomes an analytic subject area in its own right!

Data Warehouse Load Design
Package Logging (Continued)

- Logging happens automatically at execution time
  - However, it may not capture all useful information

- Custom logging can be achieved with:
  - Execute SQL task, inserting rows into custom logging tables
  - Data flow destinations loading rows into custom logging tables
  - Script task, or component

- Consider sending a notification when certain logging takes place (i.e. errors), by using the Send Mail task

Recommended Practices

- Design the ETL for scale and performance
- Exploit the capabilities of back end systems
  - Do not feel compelled to design data flow when an easier, and perhaps more optimal, design can be achieved with SQL scripts
  - If the data to be loaded exists in a staging system, it is often more efficient to perform set-based operations to transform and load the data into a data warehouse
  - The SSIS package can simply call stored procedures to achieve this

Module Outline
12 | Exploring Additional ETL Design Concepts

Topic
Data Warehouse Load Design
ETL Management
Demo: Developing and Managing an ETL Solution
Lab: Loading and Maintaining a Data Warehouse

ETL Management
Deployment and Management

- Projects can be deployed to the **SSIS Catalog**
- The SSIS Catalog is the central point for working with SSIS projects
- Features:
  - Secure package store, and permission management
  - T-SQL API to query and automate the SSIS Catalog
  - User interface with scripting options
  - Execution environments

ETL Management
Deployment and Management (Continued)

- Features:
  - Support for Windows PowerShell
  - Data taps to output a package data flow path to a CSV file
  - Built-in logging of events, configured with pre-defined logging levels
  - Built-in reports

ETL Management
Deployment and Management ► Built-In Reports

- SSIS Catalog built-in reports provide monitoring and troubleshooting capabilities
- Reports surface logged package execution data
- Reports include interactivity features:
  - Filtering
  - Sorting
  - Drill through
- Custom reports can be added

Microsoft

Module Outline
12 | Exploring Additional ETL Design Concepts
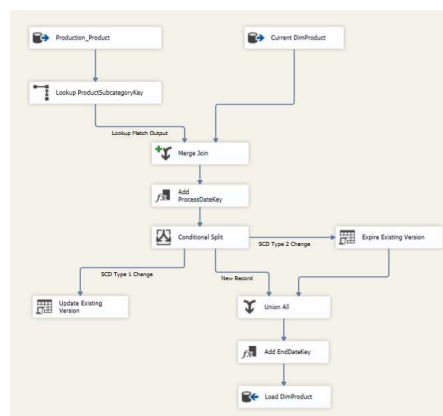
Topic
Data Warehouse Load Design
ETL Management
Demo: Developing and Managing an ETL Solution
Lab: Loading and Maintaining a Data Warehouse

Demo

Developing and Managing an ETL Solution

Demo objectives:
1. Exploring a **master** package design
2. Deploying the project to the SSIS catalog
3. Managing package execution

Module Outline
12 | Exploring Additional ETL Design Concepts

Topic
Data Warehouse Load Design
ETL Management
Demo: Developing and Managing an ETL Solution
Lab: Loading and Maintaining a Data Warehouse

Lab

03 | Loading and Maintaining a Data Warehouse

Lab exercises:

1. Exploring the **PopulateDW** SSIS Project
2. Developing the **LoadDimDate** Package
3. Completing the **LoadDimProduct** Package
4. Exploring the **LoadFactProductInventory** Package
5. Completing the **Master** Package
6. Deploying the SSIS Project

Lab

03 | Loading and Maintaining a Data Warehouse

Tips:

- Be sure to read instructions carefully, especially when executing scripts

- If you did not successfully complete last week's lab, there is a lab setup shortcut, but you will still need to provision and setup an Azure VM

- The lab setup shortcut can be used to reset and try again
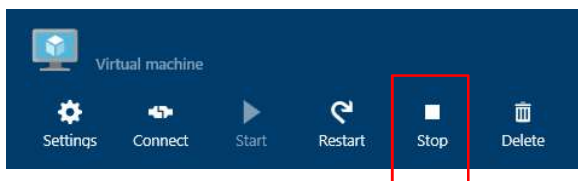
Lab

03 | Loading and Maintaining a Data Warehouse

Reminder

When you have completed, remember to stop/delete your VM

You are charged when the VM status is **Running**, but you are not charged when the VM status is **Stopped (Deallocated)**