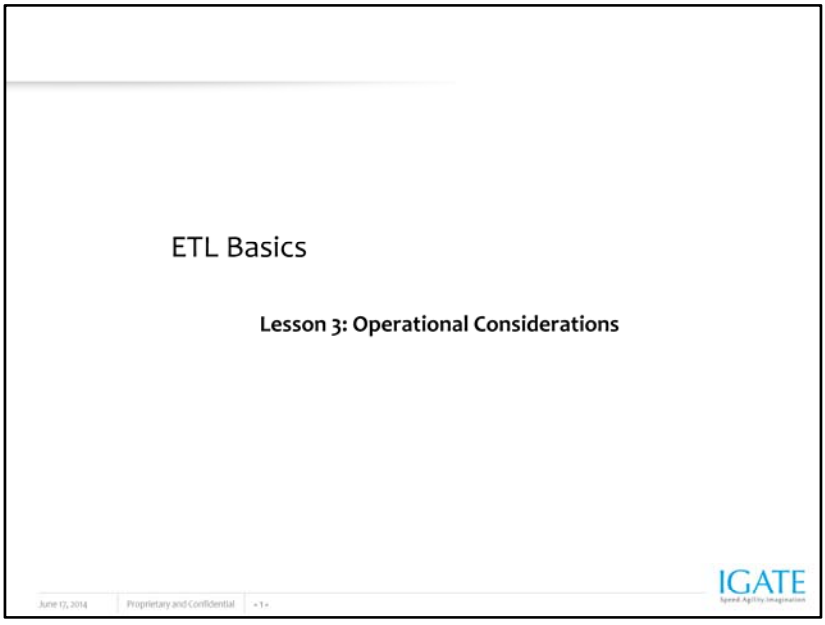
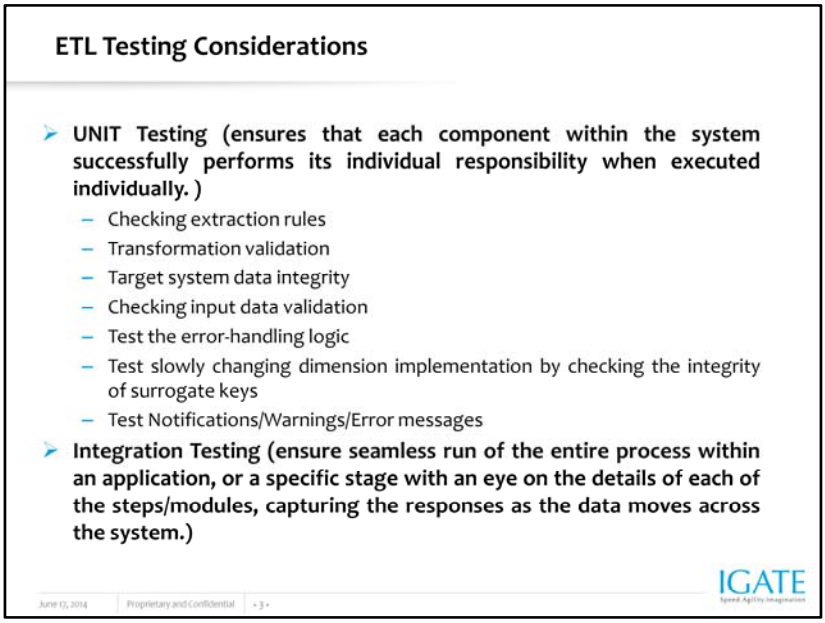
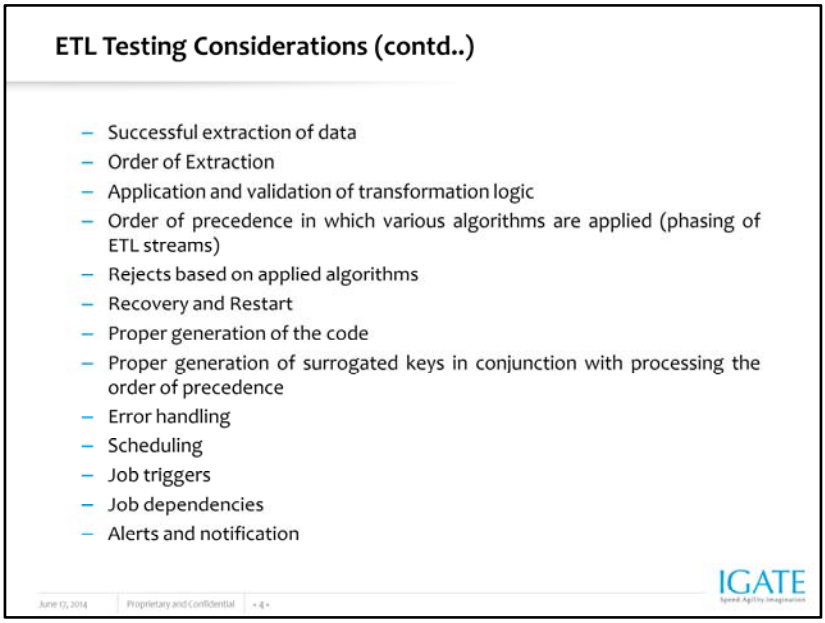
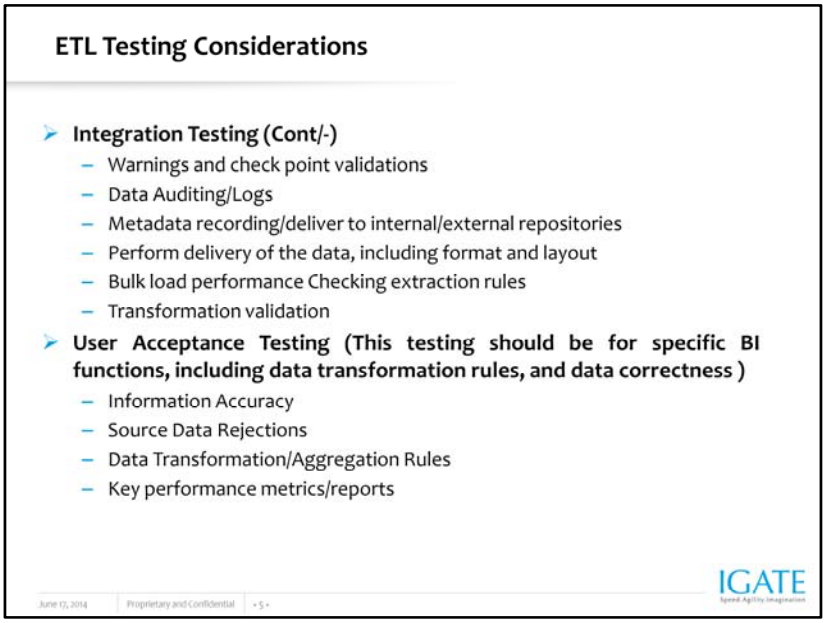

ETL Basics
Lesson 3: Operational Considerations
June 17, 2014
Proprietary and Confidential
IGATE

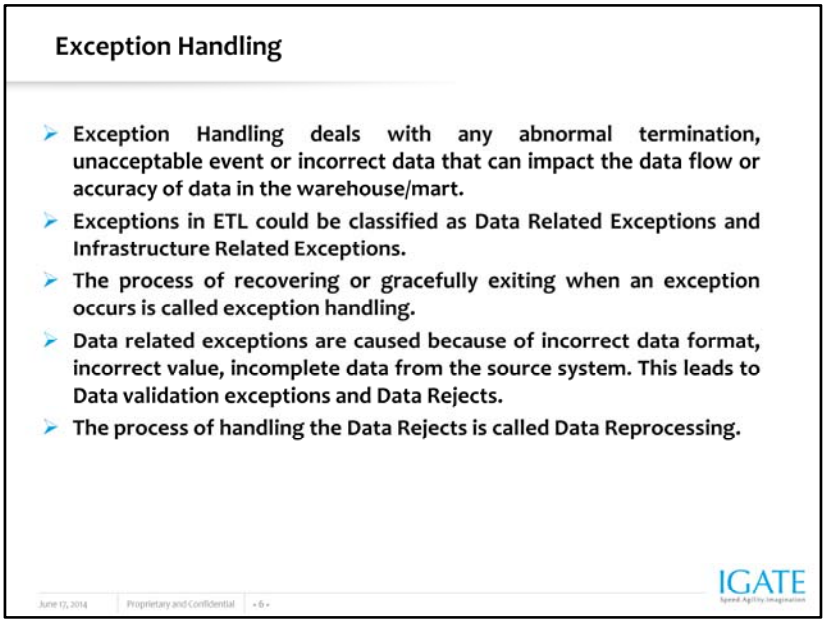
Lesson Objectives

- **On completion of this lesson on ETL basics, you will be able to understand:**
  - Handling Exceptions in ETL
  - Notifications and Alerts in ETL
  - Recovery and Restartability


ETL Testing Considerations
UNIT Testing (ensures that each component within the system successfully performs its individual responsibility when executed individually.)
Checking extraction rules
Transformation validation
Target system data integrity
Checking input data validation
Test the error-handling logic
Test slowly changing dimension implementation by checking the integrity of surrogate keys
Test Notifications/Warnings/Error messages
Integration Testing (ensure seamless run of the entire process within an application, or a specific stage with an eye on the details of each of the steps/modules, capturing the responses as the data moves across the system.)
June 17, 2014
Proprietary and Confidential
IGATE


ETL Testing Considerations (contd..)
– Successful extraction of data
– Order of Extraction
– Application and validation of transformation logic
– Order of precedence in which various algorithms are applied (phasing of ETL streams)
– Rejects based on applied algorithms
– Recovery and Restart
– Proper generation of the code
– Proper generation of surrogated keys in conjunction with processing the order of precedence
– Error handling
– Scheduling
– Job triggers
– Job dependencies
– Alerts and notification
IGATE


ETL Testing Considerations
Integration Testing (Cont/-)
Warnings and check point validations
Data Auditing/Logs
Metadata recording/deliver to internal/external repositories
Perform delivery of the data, including format and layout
Bulk load performance Checking extraction rules
Transformation validation
User Acceptance Testing (This testing should be for specific BI functions, including data transformation rules, and data correctness)
Information Accuracy
Source Data Rejections
Data Transformation/Aggregation Rules
Key performance metrics/reports
June 17, 2014
Proprietary and Confidential
IGATE

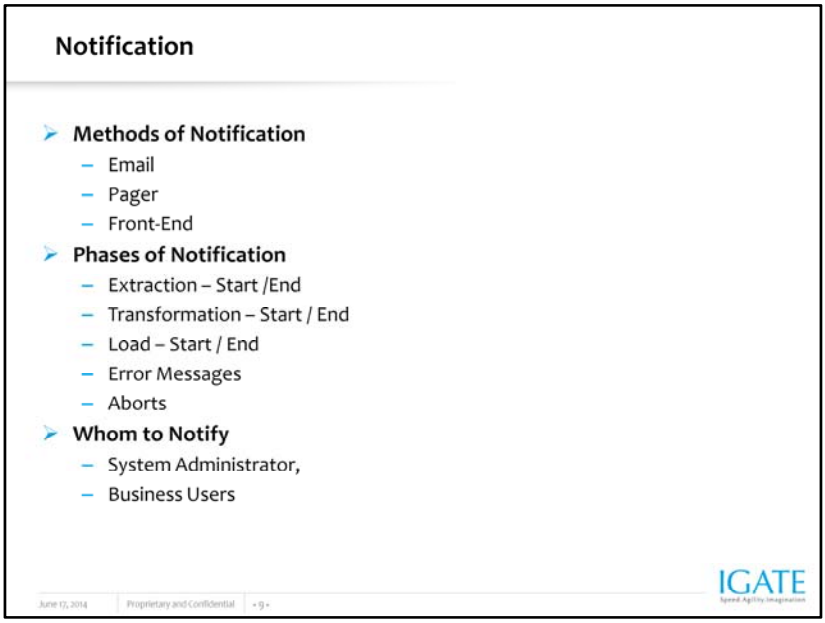

Exception Handling
Exception Handling deals with any abnormal termination, unacceptable event or incorrect data that can impact the data flow or accuracy of data in the warehouse/mart.
Exceptions in ETL could be classified as Data Related Exceptions and Infrastructure Related Exceptions.
The process of recovering or gracefully exiting when an exception occurs is called exception handling.
Data related exceptions are caused because of incorrect data format, incorrect value, incomplete data from the source system. This leads to Data validation exceptions and Data Rejects.
The process of handling the Data Rejects is called Data Reprocessing.
June 17, 2014
Proprietary and Confidential
IGATE

Exception Handling

- Infrastructure related exceptions are caused because of issues in the Network , the Database and the Operating System.
- Common Infrastructure exceptions are FTP failure, Database connectivity failure, File system full etc.
- The data related exceptions are usually documented in the requirements, if not they must be because if the data related exceptions are not handled they lead to inaccurate data in the warehouse/mart.
- We also keep a threshold of maximum number of validation or reject failures allowed per load.
- Any value above the threshold would mean the data would be too inaccurate due to too many rejections.

Exception Handling

- **There is one more exception which is the presence of inaccurate or incorrect data in the warehouse. This could happen due to**
  - Incorrect requirement or missed, leading to incorrect ETL.
  - Incorrect interpretation of requirements leading to incorrect ETL.
  - Uncaught coding defects.
  - Incorrect data from source.
- **The process of Correction of the data already loaded in the warehouse involves fixing the data already loaded and also preventing the inaccuracy to persist in the future.**


Notification
Methods of Notification
Email
Pager
Front-End
Phases of Notification
Extraction – Start /End
Transformation – Start / End
Load – Start / End
Error Messages
Aborts
Whom to Notify
System Administrator,
Business Users
June 17, 2014
Proprietary and Confidential
- 9 -
IGATE

Recovery & Restartability

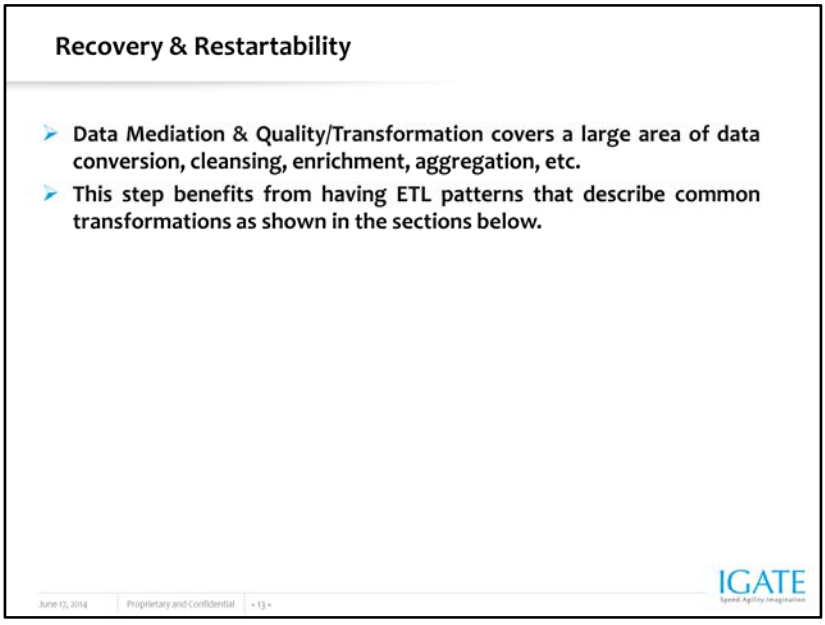
- **Design of Rollback and Recovery Procedures**
  - Rollback and Recovery Procedures define the strategy for handling load failures.
  - This includes recommendations on whether milestone points and staging are required for restarts.
  - The concept of rollback and recovery of ETL processes needs to be considered during the preliminary design stage as it affects all ETL jobs.
  - For long ETL processing blocks such as a data warehouse load the end to end process may take hours or even days to run.
  - It is important to have built in restartability to recover from fatal errors during the processing cycle.

Recovery & Restartability

- **Milestone Recovery**
  - The simplest form of recovery is to introduce vertical bands into the ETL processing cycle.
  - A vertical band is a set of jobs that run together and when complete they arrive at a milestone point.
  - If the following block of jobs fail it can be restarted from this milestone.
  - A milestone point requires some type of data staging, with the data being placed in temporary files or tables where they can be extracted by the next block.

Recovery & Restartability

- **The best practice for complex data loads is to have 3 vertical bands: Data Sourcing, Data Mediation & Quality/Transformation and Data Load.**
  - Data Sourcing involves retrieving the data from sources and delivering it to files or tables with some type of time stamping to allow for time based processing.
  - This data undergoes as little transformation as possible and once delivered it is stable.
  - This milestone point means the Data Transformation block can be started and restarted without having the source data affected by user transactions.


Recovery & Restartability
Data Mediation & Quality/Transformation covers a large area of data conversion, cleansing, enrichment, aggregation, etc.
This step benefits from having ETL patterns that describe common transformations as shown in the sections below.
June 17, 2014
Proprietary and Confidential
IGATE

Recovery & Restartability

- **Data Load involves delivering the final data to the target database.**
  - This band holds as little of the transformation logic as possible.
  - It is focused on achieving a robust database update by controlling transaction sizing and trapping database rejects.
  - Database updates are the most volatile part of the process due to the complexity of RDBMS communications and the difficulty most ETL engines have with correctly rolling back and restarting a failed update.

Recovery & Restartability

- **Individual Job Recovery**
  - In many instances it is possible to recover from a fatal error by restarting the job that failed and continuing the ETL cycle from that point.
  - Many ETL and scheduling tools provide the functionality to automate this process.
  - It may be necessary for production support to first investigate the problem and fix it before the automated recovery begins.
  - This is a short cut to restarting from previous milestone points.

Recovery & Restartability

- **Individual Job Recovery**
  - It is usually easy to do job recovery on the Sourcing and Transformation bands as these typically stage the data to temporary files or tables.
  - Restarting an individual job recreates the target output of these jobs without rollback problems.
  - In these cases the ETL scheduling tool can be used to restart the sequence from the correct point.

Recovery & Restartability

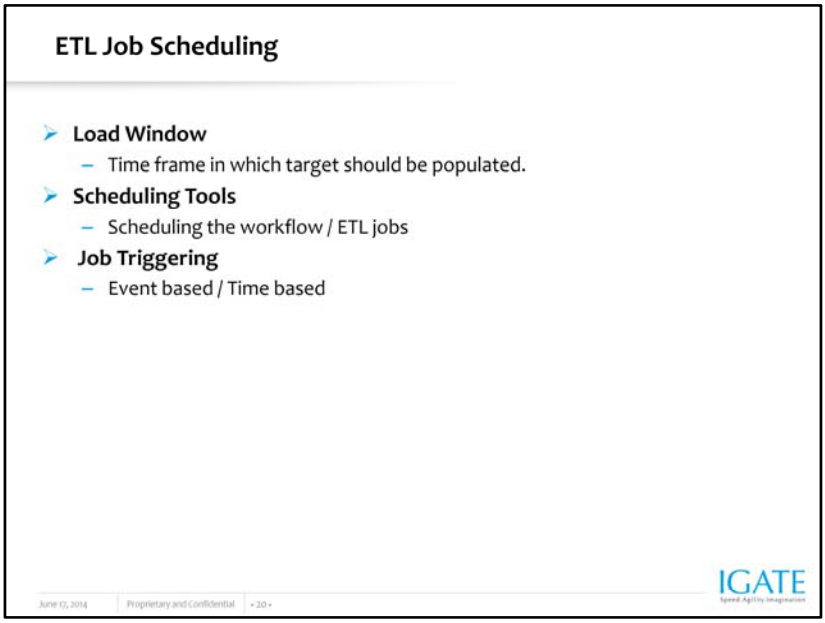
- **Individual Job Recovery**
  - The Data Load band is the most difficult for rollback and recovery as a job may fail in the process of updating a database.
  - If the update is an insert, or an update to an aggregated table then it is difficult to determine how many rows of stage data have already been processed.
  - A simple job restart may result in duplicate rows or duplicate increases to aggregate results.
  - For a Data Load job it may be possible to restart the job or it may be necessary to build full table rollback into a job restart.
  - It is worth considering enhancing the design to assist with rollback – for example, a batch number could be added to transaction tables to facilitate deletion of partial or erroneous insertions.

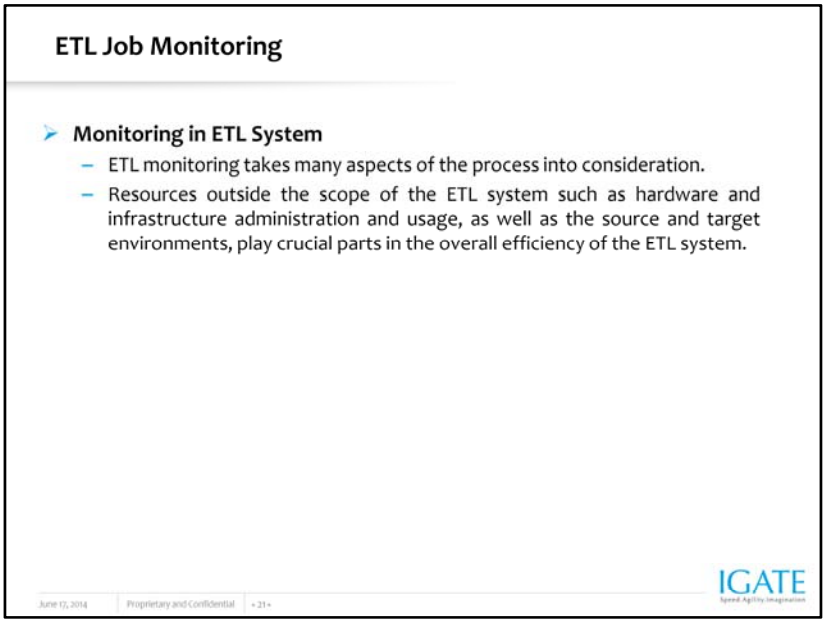
Restartability Matrix

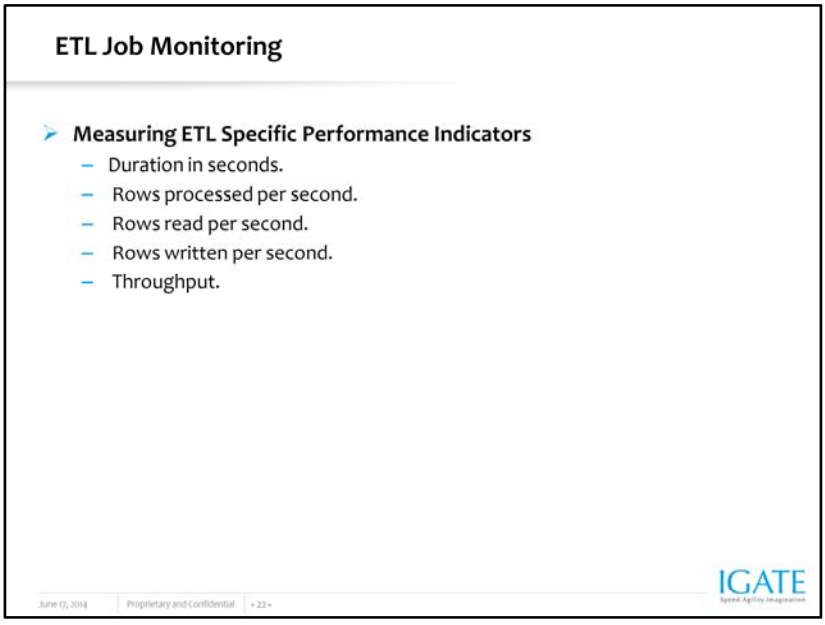
Issue	Steps to Mitigate Impact on Restartability	Party Responsible for Ensuring Steps are Completed
Data in source table changes frequently	**Append source data with a datestamp, and store a snapshot of** source data in a backup schema until the session has completed successfully	Database Administrator (creates backup schema in repository) Data Integration Developer (ensures that session calls backup schema when session recovery is performed)
Mappings in certain sessions are dependent on data produced by mappings in other sessions	**Arrange sessions in a sequential batch; configure sessions to run only if previous sessions are completed successfully**	Data Integration Developer
Session uses the Bulk Loading parameter	If sessions fail frequently due to external problems (e.g., network downtime), reconfigure the session to normal load. Bulk loading bypasses the database log, making session unrecoverable	Data Integration Developer
Only the Informatica Administrator can recover or restart sessions	Configure the session to send an email to the Informatica administrator when a session fails	Data Integration Developer
Multiple sessions within a concurrent batch fail	Work with database administrator to determine when failed sessions should be recovered, and when targets should be truncated and entire session run again.	Data Integration Developer Database Administrator

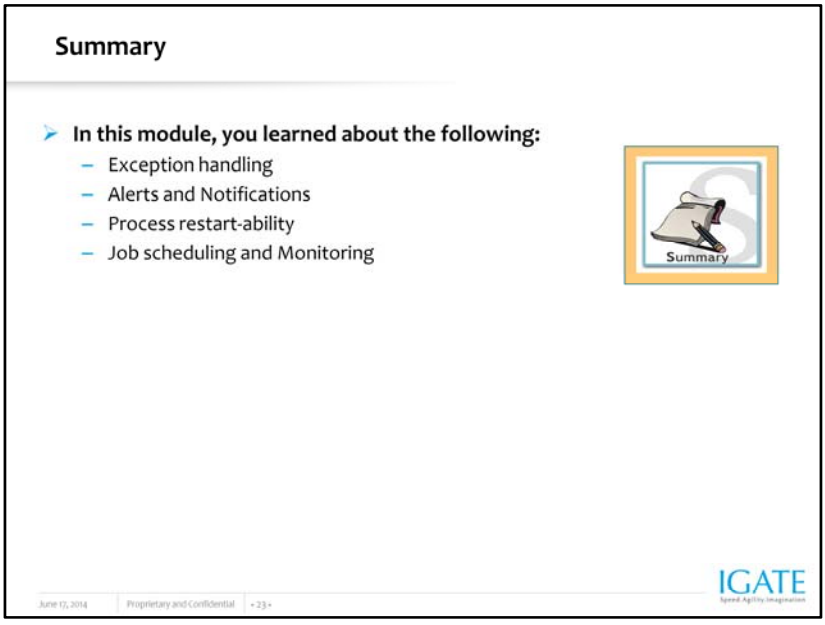
ETL Job Scheduling

- **ETL Job Scheduling is an operational process which is required to determine the sequence and time of execution of the various data flows (Jobs/Mappings).**
- **ETL schedule is dependent on the following**
  - Load Order
    - Order in which target data will be populated.
  - External Dependencies like
    - Timeframe of the source data availability.
    - Warehouse/Mart database Maintenance, like database backup time.
    - Operating System Maintenance, like file system backup time.
  - ETL's inter process flow dependencies like conformed dimensions ETL before the subject area specific dimension ETL, Dimension table ETL before the Fact Table ETL etc.


ETL Job Scheduling
Load Window
Time frame in which target should be populated.
Scheduling Tools
Scheduling the workflow / ETL jobs
Job Triggering
Event based / Time based
June 17, 2014
Proprietary and Confidential
IGATE


ETL Job Monitoring
Monitoring in ETL System
ETL monitoring takes many aspects of the process into consideration.
Resources outside the scope of the ETL system such as hardware and infrastructure administration and usage, as well as the source and target environments, play crucial parts in the overall efficiency of the ETL system.
June 17, 2014
Proprietary and Confidential
IGATE


ETL Job Monitoring
Measuring ETL Specific Performance Indicators
Duration in seconds.
Rows processed per second.
Rows read per second.
Rows written per second.
Throughput.
June 17, 2014
Proprietary and Confidential
+ 22 +
IGATE



Add the notes here.