05: ETL architecture interview Q&As

Posted on November 25, 2015 by Arulkumaran Kumaraswamipillai

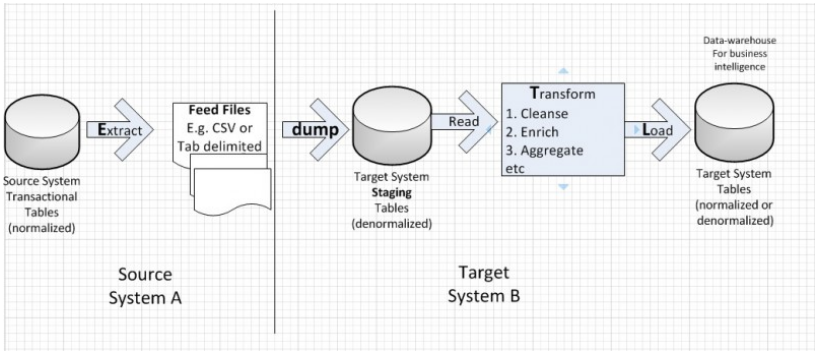
Q1. What is an ETL process?

A1. ETL is a architectural style, and it stands for **E**xtract, **T**ransform and **L**oad. Extract does the process of reading data from an input data source like file, database, etc. Transform does the converting of data into a format that could be appropriate for the output data source like file, Database, etc. Load does the process of writing the data into the output data-source.

Q2. What are the real life examples of applying the ETL architecture?

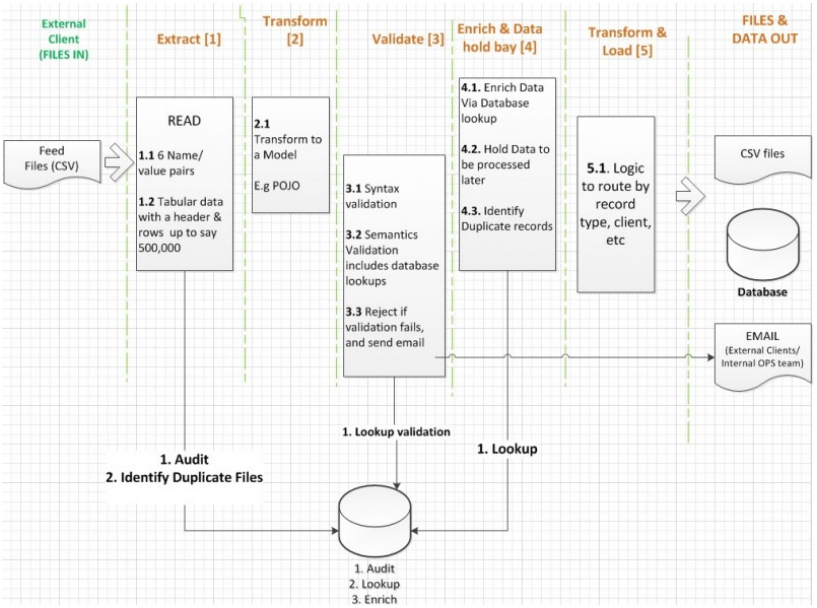
A2.

1) Extract data from the STP (Straight Through Processing), and transform by aggregating the data to be fed into target data-warehouse systems to produce multi-dimensional reporting and analysis purpose. In other words, the transactional data are aggregated and fed into “business intelligence” systems to make key strategical business decisions.



ETL for Data warehousing & Business Intelligence (BI)

2) Migrate Data from company X to Company Y, where the data are stored in different data stores and formats. The source data from :company X” often need to be validated, cleansed, enriched, and transformed before loaded into the target systems of “company Y”.



ETL flow

3 Migrate data within a company from legacy systems like "Main frame systems" to databases like Oracle feeding data to on-line web based applications. Basically, creating a copy of the data. The data in the legacy systems will still be the **source of truth**.

Q3. As a Java developer/architect, what tools and frameworks will you be using for ETL?
A3.

1) Spring batch: Spring batch tutorials

2) ETL tools like Pentaho DI (aka Kettle), IBM data stage, Informatica, etc. Pentaho ETL tool tutorials

Q4. What are the key decisions for ETL architectures?
A4.

#1. Should we use a ETL tool?

For example Spring batch Vs an ETL tool like Pentaho DI?

#1. ETL tools like Pentaho DI, etc are developed with ETL process in mind, hence can be much faster to develop with as you will just have to drag and drop steps and configure your input/output for processing. These tools are very good at solving common problems fast and provide additional customisation by allowing you to write Java based transformation code.

#2. For ETL scenarios with lots of edge cases, business logic, complex enrichment & transformation logic, etc the Spring batch will be favored for its flexibility and **code reuse**. For example, the business logic and enrichment logic can be reused by the other systems. The down side is that you need to write a lot more code.

#3. A hybrid approach of using both by playing to their strengths. This is was demonstrated via the Pentaho tutorial where rows read from a transformation step from an ETL tool

can be passed to a Java layer. Passing Pentaho DI (i.e. Kettle) rows of data to Java layer (i.e. POJOs)

This decision must also be made on the basis of available resources to build and manage the system. If you decide to use an ETL tool, don't expect a big payback in your first iteration. The advantages will become more apparent in additional iterations or projects as you leverage the development advantages of using a tool during subsequent implementations.

#2. Should we use normalized or de-normalized tables?

For example, the staging tables need to be de-normalized, and the target tables can be normalized (e.g. Transactional systems (OLTP)) or de-normalized(e.g. data-warehouse systems (OLAP)).

#3. Where are we going to stage the data?

For examples, database tables, as files, in memory, etc. The main reason for staging the data before transforming/enriching

1) A recovery/restart point is desired in the event the ETL job fails whilst processing potentially due to a break in the connection between the source and ETL environment.

2) For auditing and non-repudiation (e.g. any disputes with a client or other teams as to what was sent, etc).

3) Long running ETL processes may open a connection to the source system that create problems with database locks and that stresses the transaction system.

#4. How are we going to validate & cleanse the data?

For example,

1) Syntactic and semantic validations need to be performed.

2) Some data need to be cleansed. For example, trimming spaces, removing control characters, etc

3) If the validation fails, do we reject the whole feed file or the delta (i.e. the records that failed) or tag the records as bad and pass them through.

#5. How are we going to enrich the data?

1) Data base look-ups. E.g. static vs dynamic look-ups. Real time look-ups vs cached data look up?

2) Restful web service calls.

#6. Non functional requirements like performance, scalability?

1 What volume of data are we dealing with? Are we looking at big data for analytics? High volume ETL throughput and management require specific considerations and implementation techniques. For example, Sybase has “bulk copy (i.e. BCP)” feature. Splitting files and processing them in parallel. Spring batch allows you to set chunk sizes and multi-thread the processing. De-normalized tables without any indices for write performance.

2) Any SLAs (i.e. Service Level Agreements) like “processing 1 million records in 2 hours”, etc. Some ETL processes need to run overnight during outside business hours, whereas other ETL jobs need to run during business hours meet end-of-day processing cut-off times.

3) Transactional integrity. Loading data into target systems with transactional integrity.

4) How can feed files from/to external clients/organizations can be delivered securely? E.g, SFTP, file upload with a login, password protecting the files, etc. Also, access controls to the

folders so that “client A” would not have access to folders of “Client B”, and so on.

Popular Member Posts

♦ 11 Spring boot interview questions & answers

850 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

768 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

399 views

18 Java scenarios based interview Questions and Answers

387 views

♦ 7 Java debugging interview questions & answers

308 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

305 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

297 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

293 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

246 views

001B: ♦ Java architecture & design concepts interview questions & answers

204 views

Bio | Latest Posts

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and

often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ Passing Pentaho DI (i.e. Kettle) rows of data to Java layer (i.e. POJOs)

Converting POJOs to Pentaho meta objects and then passing to a Kettle transformation ›

Posted in Java Architecture Interview Q&A, member-paid

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category

© Disclaimer

 ↑