

Design Tip #99 Staging Areas and ETL Tools

By Joy Mundy

The majority of the data warehouse efforts we've recently worked with use a purchased tool to build the ETL system. A diminishing minority write custom ETL systems, which usually consist of SQL scripts, operating system scripts, and some compiled code. With the accelerated use of ETL tools, several of our clients have asked, does Kimball Group have any new recommendations with respect to data staging?

In the old days, we would extract data from the source systems to local files, transfer those files to the ETL server, and often load them into a staging relational database. That's three writes of untransformed data right there. We might stage the data several more times during the transformation process. Writes, especially logged writes into the RDBMS, are very expensive; it's a good design goal to minimize writes.

In the modern era, ETL tools enable direct connectivity from the tool to the source database. You can write a query to extract the data from the source system, operate on it in memory, and write it only once: when it's completely clean and ready to go into the target table. Although this is theoretically possible, it's not always a good idea.

- The connection between source and ETL can break mid-stream.
- The ETL process can take a long time. If we are processing in stream, we'll have a connection open to the source system. A long-running process can create problems with database locks and stress the transaction system.
- You should always make a copy of the extracted, untransformed data for auditing purposes.

How often should you stage your data between source and target? As with so many issues associated with designing a good ETL system, there is no single answer. At one end of the spectrum, imagine you are extracting directly from a transaction system during a period of activity – your business is processing transactions all the time. Perhaps you also have poor connectivity between your source system and your ETL server. In this scenario, the best approach is to push the data to a file on the source system, and then transfer that file. Any disruption in the connection is easily fixed by restarting the transfer.

At the other end of the spectrum, you may be pulling from a quiet source system or a static snapshot. You have a high bandwidth, reliable connection between snapshot and ETL system. In this case, even with large data volumes, it may be perfectly plausible to pull directly from the snapshot source and transform in stream.

Most environments are in the middle: most current ETL systems stage the data once or twice between the source and the data warehouse target. Often that staging area is in the file system, which is the most efficient place to write data. But don't discount the value of the relational engine in the ETL process, no matter which ETL tool you're using. Some problems are extremely well suited to relational logic. Although it's more expensive to write data into the RDBMS, think about (and test!) the end-to-end cost of staging data in a table and using the relational engine to solve a thorny transformation problem.