Exhibit 9
Kimball's 38 Processes for ETL

1. **Extract system.** Source data adapters, push/pull/dribble job schedulers, filtering and sorting at the source, proprietary data format conversions, and data staging after transfer to ETL environment.
2. **Change data capture system.** Source log file readers, source date and sequence number filters, and CRC-based record comparison in ETL system.
3. **Data profiling system.** Column property analysis including discovery of inferred domains, and structure analysis including candidate foreign key — primary relationships, data rule analysis, and value rule analysis.
4. **Data cleansing system.** Typically a dictionary driven system for complete parsing of names and addresses of individuals and organizations, possibly also products or locations. "De-duplication" including identification and removal usually of individuals and organizations, possibly products or locations. Often uses fuzzy logic. "Surviving" using specialized data merge logic that preserves specified fields from certain sources to be the final saved versions. Maintains back references (such as natural keys) to all participating original sources.
5. **Data conformer.** Identification and enforcement of special conformed dimension attributes and conformed fact table measures as the basis for data integration across multiple data sources.
6. **Audit dimension assembler.** Assembly of metadata context surrounding each fact table load in such a way that the metadata context can be attached to the fact table as a normal dimension.
7. **Quality screen handler.** In line ETL tests applied systematically to all data flows checking for data quality issues. One of the feeds to the error event handler (see subsystem 8).
8. **Error event handler.** Comprehensive system for reporting and responding to all ETL error events. Includes branching logic to handle various classes of errors, and includes real-time monitoring of ETL data quality
9. **Surrogate key creation system.** Robust mechanism for producing stream of surrogate keys, independently for every dimension. Independent of database instance, able to serve distributed clients.
10. **Slowly Changing Dimension (SCD) processor.** Transformation logic for handling three types of time variance possible for a dimension attribute: Type 1 (overwrite), Type 2 (create new record), and Type 3 (create new field).
11. **Late arriving dimension handler.** Insertion and update logic for dimension changes that have been delayed in arriving at the data warehouse.
12. **Fixed hierarchy dimension builder.** Data validity checking and maintenance system for all forms of many-to-one hierarchies in a dimension.
13. **Variable hierarchy dimension builder.** Data validity checking and maintenance system for all forms of ragged hierarchies of indeterminate depth, such as organization charts, and parts explosions.
14. **Multivalued dimension bridge table builder.** Creation and maintenance of associative (bridge) table used to describe a many-to-many relationship between dimensions. May include weighting factors used for allocations and situational role descriptions.
15. **Junk dimension builder.** Creation and maintenance of dimensions consisting of miscellaneous low cardinality flags and indicators found in most production data sources.

16. **Transaction grain fact table loader.** System for updating transaction grain fact tables including manipulation of indexes and partitions. Normally append mode for most recent data. Uses surrogate key pipeline (see subsystem 19).
17. **Periodic snapshot grain fact table loader.** System for updating periodic snapshot grain fact tables including manipulation of indexes and partitions. Includes frequent overwrite strategy for incremental update of current period facts. Uses surrogate key pipeline (see subsystem 19).
18. **Accumulating snapshot grain fact table loader.** System for updating accumulating snapshot grain fact tables including manipulation of indexes and partitions, and updates to both dimension foreign keys and accumulating measures. Uses surrogate key pipeline (see subsystem 19).
19. **Surrogate key pipeline.** Pipelined, multithreaded process for replacing natural keys of incoming data with data warehouse surrogate keys.
20. **Late arriving fact handler.** Insertion and update logic for fact records that have been delayed in arriving at the data warehouse.
21. **Aggregate builder.** Creation and maintenance of physical database structures, known as aggregates, that are used in conjunction with a query-rewrite facility, to improve query performance. Includes stand-alone aggregate tables and materialized views.
22. **Multidimensional cube builder.** Creation and maintenance of star schema foundation for loading multidimensional (OLAP) cubes, including special preparation of dimension hierarchies as dictated by the specific cube technology.
23. **Real-time partition builder.** Special logic for each of the three fact table types (see subsystems 16, 17, and 18) that maintains a “hot partition” in memory containing only the data that has arrived since the last update of the static data warehouse tables.
24. **Dimension manager system.** Administration system for the “dimension manager” who replicates conformed dimensions from a centralized location to fact table providers. Paired with subsystem 25.
25. **Fact table provider system.** Administration system for the “fact table provider” who receives conformed dimensions sent by the dimension manager. Includes local key substitution, dimension version checking, and aggregate table change management.
26. **Job scheduler.** System for scheduling and launching all ETL jobs. Able to wait for a wide variety of system conditions including dependencies of prior jobs completing successfully. Able to post alerts.
27. **Workflow monitor.** Dashboard and reporting system for all job runs initiated by the Job Scheduler. Includes number of records processed, summaries of errors, and actions taken.
28. **Recovery and restart system.** Common system for resuming a job that has halted, or for backing out a whole job and restarting. Significant dependency on backup system (see subsystem 36).
29. **Parallelizing/pipelining system.** Common system for taking advantage of multiple processors, or grid computing resources, and common system for implementing streaming data flows. Highly desirable (eventually necessary) that parallelizing and pipelining be invoked automatically for any ETL process that meets certain conditions, such as not writing to the disk or waiting on a condition in the middle of the process.
30. **Problem escalation system.** Automatic plus manual system for raising an error condition to the appropriate level for resolution and tracking. Includes simple error log entries, operator notification, supervisor notification, and system developer notification.

31. **Version control system.** Consistent "snapshotting" capability for archiving and recovering all the metadata in the ETL pipeline. Check-out and check-in of all ETL modules and jobs. Source comparison capability to reveal differences between different versions.
32. **Version migration system.** development to test to production. Move a complete ETL pipeline implementation out of development, into test, and then into production. Interface to version control system to back out a migration. Single interface for setting connection information for entire version. Independence from database location for surrogate key generation.
33. **Lineage and dependency analyzer.** Display the ultimate physical sources and all subsequent transformations of any selected data element, chosen either from the middle of the ETL pipeline, or chosen on a final delivered report (lineage). Display all affected downstream data elements and final report fields affected by a potential change in any selected data element, chosen either in the middle of the ETL pipeline, or in an original source (dependency).
34. **Compliance reporter.** Comply with regulatory statutes to prove the lineage of key reported operating results. Prove that the data and the transformations haven't been changed. Show who has accessed or changed any such data.
35. **Security system.** Administer role-based security on all data and metadata in the ETL pipeline. Prove that a version of a module hasn't been changed. Show who has made changes.
36. **Backup system.** Backup data and metadata for recovery, restart, security, and compliance requirements.
37. **Metadata repository manager.** Comprehensive system for capturing and maintaining all ETL metadata, including all transformation logic. Includes process metadata, technical metadata, and business metadata.
38. **Project management system.** Comprehensive system for keeping track of all ETL development.

Sources:

- Kimball, Ralph, Margy Ross, Warren Thornthwaite, Joy Mundy, and Bob Becker, *The Data Warehouse Life Cycle Toolkit, Second Edition*, Wiley, 2008, ISBN 978-0-470-14977-5
- http://www.kimballgroup.com/2004/12/the-38-subsystems-of-etl/