Source: DWS YouTube presentation: [YouTube](https://www.youtube.com/watch?v=DspXXZrSVRk&t=2611s)

I’m building up the database for the Foundr ap and I have no idea where to start. In my research and search for inspiration, I came across a data engineering post that said something about learning the Kimball methodology. I didn’t know what that was and so I decided to start reading up on it. In my search I came across a video recorded at a conference and the presentation covers the first bit of the “data warehousing toolkit” book written by Ralph Kimball and co. I will be attempting to use the universal principles for building a data warehouse to inform my database building. Such an exercise should also, hopefully, inform my understanding of the databases I encounter out in the wild.

1. Accessibility

Accessibility is about creating a meaningful connection between the data that comes in through the system of record (SOR) and the data warehouse. The system of record is primarily concerned with fast insertion. We are working with transactional databases, OLTP databases, that are primarily created for high volume data that passes through various points. Individual transactions can be accessed at this point, but the questions that can be answered are limited. The data warehouse is where we have OLAP databases that are built to answer a wider range of questions. Transformation are performed on the data to make it more meaningful and useful.

In creating a database for Foundr, there are several things that I would say constitute good practice. I recommend the following:

1. Establish a doc with a clear explanation of the naming conventions (for tables, variables, schemas)
   1. If the convention is followed, it should be easy to develop user friendly applications that sit on top of either the SOR or the data warehouse. You could have a program that links to a website explaining the different terms and it could require little manual intervention when new tables and variables are added.
2. Create a data dictionary that is kept up to date
3. Maintain a highly reliable process for logging data ingestion and changes
4. Incorporate data quality checks as far upstream as possible. Consideration required to the varying needs and data quality requirements of the business units, but the basic DQ checks can be performed from the start.
5. Ensure adherence to the naming conventions and correct use of the data warehouse
6. Apply best principles around load balancing and database locking
7. Promote a culture of continuous integration, continuous deployment
8. Develop ruthlessly fast integration points
9. Adaptability
   1. Draw out a clear data remediation process in the case of discrepancies between the various source systems, business units, servers, etc.,
   2. Set out the appropriate steps for adding or removing attributes/tables/schemas/databases
   3. Have a designated staging area that can perform things such as:
      1. Deduplication
      2. Combining of sources
      3. Cleansing (misspellings, resolving domain conflicts, missing elements, parsing issues)
      4. Assign a warehouse key (internal and intended for tracking) – surrogate key
10. Loading the presentation area
    1. Real time load of the data into the data marts if possible
    2. Establish clear communication between the various points in the process, taking careful consideration for users

Questions:

1. How do you build out the various environments that different users will need to use (dev vs uat vs prod)? Also, potential ad hoc environments.
2. What are the controls placed on sensitive data (both at rest and in transit)?
3. Where does a data lake or data lake house fit into the picture?
4. How do you decide on the various layers? Batch processing vs real-time processing? Is there a raw data vs transformed/conformed data layer?
5. How do you determine what should be stored in a SQL database vs what should be stored in a NoSQL database?
6. How do you store files more generally? Flat file vs XML vs database table.
7. How do you design dimensionally (for the data warehouse) vs relationally (for the transactions)?
8. At which points are we uncompromising about documentation?
9. Kimball or Inmon?