Things need to be considered before making a parity plan:

* What tables to check
* How many days/weeks/years of data need to be checked
* What is the source of truth?
* What is the expectation? (1%, 3% or 5%)
* How much time you have confirm parity? This needs to be factored into release plan.
* Any expected discrepancy due to different methodology?
  + Additivity
  + Data curing

When making a parity plan:

1. Who does what when
2. Scope of data backfill
3. Reparity after backfill
4. Reparity after process changes
5. Reparity after data migration from db to db?
6. Automation of parity check

What we’ve done:

* For single day/week…, every row is compared in a semi-manual way in Excel if possible
* Then pull a few days/weeks…, check metrics at all levels of aggregation on each desired dimension. Addictiveness of the data is ignored here. Sum is taken for all grouped values.
* We use a python script to compare multiple metrics across multiple dimensions.
* If transformation is involved, we check the final result first and only look at raw/intermediate steps if final result doesn’t match up.

Process we’ve used when something doesn’t match:

* Check whether we need to repull one or both data sets due to business definition change or data curing issue
* Does the multitude of discrepancy tell some information?
  + If it is high level of difference, forget a filter?
  + If low level of difference, could it be due to low base number or calculation rounding error?
* Is the discrepancy systematic (overall higher or lower) or specific?
* If it is systematic, does the trend reveal a transformation mistake?
* If it is specific, try to isolate the discrepancy to certain dimensions or certain values
* If we can isolate the discrepancy, check whether we make any obvious mistake specific to that segment
* If no obvious mistakes can be identified, the last resort is to use a few reports as example and pull raw data manually using both method to see whether it’s in the raw data or in the intermediate process (although intermediate transformation is less likely if problem is isolated to specific segments).

What we’ve found:

* There is a high level of consistency/predictability on data pulled automatically (through API call and aggregated in database), unless the business requirement was wrong/changed.
* The source of truth is often an existing dashboard or process that is assembled manually, most likely with assistance of Excel. Unfortunately, it is frequently found with unpredictable human/system error that needs to be fixed along the way. The good thing is that we know that we are doing the right thing to replace an error-prone system with an error-prof one, the difficult thing is that we are matching with a moving target.