Demystifying Data Reconciliation Misconceptions in BCBS 239

While the most discussed data topic in the entire BCBS 239 regulation is about data aggregation, with the need for transparency and integrity in data management, organizations often struggle with how to accomplish data integrity. The reason for such special attention on aggregation is because of the fact that many risk models use approximation techniques as a widely accepted statistical approach for identifying and predicting risk. The values derived through such techniques are used in elucidating risk scenarios to risk managers. The risk approximation approach assumes that data used as input are extremely accurate and precise in nature to yield optimal results.

Without data accuracy and precision, the data aggregation process is vulnerable to rolling up data that is of degraded quality resulting in misleading risk reporting metrics.

The top 5 reasons for data errors during the aggregation process are:

a. Inconsistent or non-coherent source data used in aggregation process
b. Challenges due to non-integrated data between source applications leading to aggregation errors
c. Unhandled exceptions during data transformation process
d. Constrain violations during data movement phase – leading to rejects
e. Logical issues in data structure or inherent flaws in data management applications

We can avoid such data issues during aggregation by employing data reconciliation steps before and after aggregation. Implementing a gating process within the process flow based on reconciliation will prevent aggregation data errors and avoid data related ripple effects across the data stream.

While using a data reconciliation approach in each data movement step (aka Data Extract, Transform, Load (ETL) process) is not new to the data management world, there are several misconceptions about how it’s addressed from a best practices perspective.

Let’s start by demystifying data reconciliation techniques in the context of best practices that we all would love to implement if we have time, but often are pushed to ignore since we need to deliver data on time.

Data Reconciliation is a Misunderstood Data Technique

a. Two widely misconstrued subjects are data reconciliation and data quality. Data reconciliation is a technique that is used to identify data exceptions with respect to a standard benchmark or source while data quality is the process of fixing the data quality errors. While it’s true that a data reconciliation technique is widely used during a data quality process to ensure we have right checks and balance, it’s a common misconception that the reverse is true. You should not expect a data quality program to help you in data reconciliation and bring about transparency in aggregation process.

b. The data reconciliation process scope is limited to identifying, whether there are any issues in the data or not. The process of investigating the root cause of the data exceptions to pinpoint the data issue source should be done by external process that compliments the data reconciliation process.

c. A common misconception is that data reconciliation is specific to GL reconciliation alone. This couldn’t be further from the truth as data reconciliation should be a strategic part of a data governance framework that extends the reconciliation process to help you reconcile areas such reference data, master data, and transaction data.
d. A single point data reconciliation or singular reference validation within a complex risk data process doesn’t remove aggregation errors nor bring about transparency. This is seen in circumstances where an organization assumes that just having General Ledger (GL) reconciliation between the GL system and the risk warehouse eliminates all risks – It doesn’t. You need to reconcile your data at multiple touch points, collate exceptions in reconciliation and correlate them to ensure it covers the end-to-end business process flow.

**Difference Between Data Accuracy and Data Precision in Reconciliation**

First off, the reconciliation process generates exceptions such as a list of data differences that are not in agreement with a standardized reference source used as widely accepted data benchmark. It’s important to understand the exceptions and the possible root cause of such exceptions before attempting to remediate the exception.

The following are the primary causes of exceptions in the reconciliation process.

1. **Exceptions due to permanent bias or systemic errors:** An example of this is timing issues between two systems that are creating a reconciliation difference between account balances. While you can’t fix this exception or remove it from the process, you can mitigate it by adding suspense accounts or exception plugs to circumvent this issue in order to accept such exceptions within the process as expected process deviations.

2. **Random deviations:** These are caused because of the external factors that are outside of the data creation process itself, but directly or indirectly influence the data outcomes. A good example is having a business rule that sums loan balance and interest receivable together as outstanding balance while the GL system accounts them into two different GL buckets leading to an amount mismatch. Random errors are caused due to either a faulty aggregation process or an incorrect business rule. Random exceptions can be characterized by noise in the data due to missing precision.

Let us attempt to understand the difference between accuracy and precision.

An analogy that serves well in this discussion is shooting target practice at a range. The image below provides a great way to elucidate the difference.
Now, let's elaborate on each of the above four scenarios in the context of data precision vs. accuracy.

**Precision without Accuracy:** This is analogous to calibrating our shooting angle or height to achieve accuracy which in data speak might equate to fixing a timing issue between two applications. This could be done by adding a standard exception plug to correct the issue of precision, and improve the data density (i.e. number of fill factors on a given data field with respect to the data set) to improve the accuracy.

**Accuracy without Precision:** While targets achieve accuracy they fall short of precision due to random errors that are induced. These errors are uncontrollable and might be due to wind speed, temperature, etc. For data, a good example is the randomness of data exceptions in the reconciliation process for interest rate values in a 30 year fixed mortgage portfolio which can have fluctuating numbers and might not have good precision. However, they will be accurate within a range of 0-12% (min-max interest rate controlled by state usury law).

**Neither Precision or Accuracy:** This is a good example of troubled shooting caused by inexperience or lackadaisically shooting with poor aim. In the data exception world, such trends can indicate that there are some inconsistent data errors that are unpredictable. However, upon closer examination of such issues, one can conclude that the lack of accuracy is directly attributed to missing data elements - nulls, empty fields. Issues can also be attributed to faulty business rules or in-coherent reference data leading to aggregation processes that are not performed with the same level of data granularity.

**Precision and Accuracy:** Everyone aspires to precise and accurate shooting, but let's not forget that in the data world precision along with accuracy comes with a cost. As shown in the graph below, accuracy equates to being closer to reference points or benchmarks that you use while precision requires the bell curve to be much tighter. When it comes to data, precision and accuracy is only applied to data that is strategic in value based on ROI and cost constraints.

![Graph showing precision and accuracy](image)

In some cases, you might not want to be very precise. For example, if you have similar $200 value transactions in a day that constitutes 40-50% of the transaction volume this may indicate a red flag for fraud. It's always a delicate balance between the amount of accuracy and precision required. In some cases accuracy within a wider tolerance range is sufficient, while in other cases it's absolutely necessary to be precise within a very narrow tolerance. It's often best to layer your exceptions over a period of time and look at trending to ascertain such inference.
Data Aggregation Reconciliation Approach

Now that we have set a baseline of understanding, let’s delineate three commonly used reconciliation approaches that are widely followed in the data management space. These include:

1. **Master Data Reconciliation:** It’s one of the most widely used approaches to reconcile master data sets between source and target. Master data sets are generally unchanged and no active aggregation is done on this data set. The granularity of the master data must be the same between source and target.

   Note: Master data reconciliation should not be confused with the Master Data Management (MDM) programs. Master data in this context are the key data points that are consistent in measuring certain context in data.

   A good example is to validate a count of customers between source and target systems before and after an aggregation process. Or alternatively do a check sum validation on the important amount values between fixed ID fields along with the row count to ensure the source and target are in sync.

2. **Reference Data Reconciliation:** This is usually done between two reference data attributes that might be slowly changing in nature (or unchanged) but might be an important parameter on which actual aggregation is done. This requires a standard benchmark source as a guidance to compare and reconcile the data.

   An example would be reference data fields such as a loan type across two different loan systems. You need to aggregate the loan balances between the systems before you use it for risk calculations. If the loan type reference values are not in sync between the systems you might end up aggregating the loan balances between unlike asset types that can lead to data inconsistency in risk reporting.

3. **Transactional Data Reconciliation:** Sales quantity, revenue, principal balance amounts, etc. are classic examples of transactional data sets. Transactional data are basic measures used in any risk calculations and reporting. Any mismatch in such data can directly impact risk calculations and reporting results. Hence it’s important to reconcile the transactional data across multiple points in an aggregation process within the risk data flow.

   Transactional data reconciliation is ordinarily done in terms of total sum to prevent any mismatch and exceptions otherwise caused due to varying granularity of the qualifying dimension. Transactional recon can be done on full data or on incremental data; hence you would need a time reference to perform the transactional data reconciliation process.

   An example of such rollup parameters would be reconciliation of asset values with the General Ledger (GL) data by amount field. In this example, aggregation happens by asset type on the amount field while the GL system stores the aggregated amount value as benchmark to compare the final results and generate an exception report as necessary.
Implementing Data Reconciliation

**Manual ETL based Reconciliation:** This is a widely used reconciliation approach which is a combination of manual and ETL based reconciliation that is custom coded to ensure data integrity and bring about transparency.

It's done using combination of ETL tools capabilities and scripting techniques to get counts and balances at the source level when the application is providing the data extract. ETL tools are used to read the feed information and track them by updating and loading metadata tables into your data warehouse. At the end of the process, custom coded reconciliation reports are generated to ensure transparency around the data aggregation process in addition to transformations that occur during the data movement.

**Tool based Reconciliation as Control Process:** This is a fairly automated, tool based environment that is independent and reusable in nature. Has quick time to deploy with self-documenting and reporting capabilities to promote end user trust. Tool based reconciliation controls are best suited for large data warehouses (DW) or complex data aggregation processes which need repeatable reconciliation process and require validation across multiple reconciliation touch points.
The figure below provides a representative example of a process flow of how such reconciliation controls would work in a loan data processing environment for a bank.

Tool based reconciliation can stretch your reconciliation capabilities beyond single data sets (i.e. daily data extract) and can span across applications (i.e. can be deployed at source applications, data movement tools, target DW and reports) creating end-to-end reconciliation capabilities.

Using tool based reconciliation can help reduce errors that are due to manual data process in order to monitor reconciliation exception trends to identify repeating reconciliation issues and remediate them. A manual ETL based reconciliation approach, though is simple to implement, doesn’t scale effectively to large scale processes and is expensive to implement due to an array of data movement processes that require reconciliation. This approach can quickly become inefficient when expanded to meet large data warehouse reconciliation requirements due to the complex volumetric involved in such reconciliation.

Infogix Enterprise Data Analysis Platform can save you time and money. Visit www.infogix.com or call 1.630.649.6800 (US, Canada and International), +44 1242 674 137 (UK and Europe).