Transactional Reporting Options - from Simple to the Advanced

# Version Information

Satya Komatineni, 11/11/23

Alpha, (I expect a couple of more releases which include enterprise data, delta lakes)

Contents

[Version Information 1](#_Toc150706794)

[Goal: List Options for Transactional Reporting 2](#_Toc150706795)

[Option 1 – Use the Transactional DB as is. 3](#_Toc150706796)

[Characteristics, Features: 3](#_Toc150706797)

[Compromises 3](#_Toc150706798)

[Cost Factors 4](#_Toc150706799)

[Initial Cost 4](#_Toc150706800)

[On going cost of development 4](#_Toc150706801)

[Option 2 – Use the Transactional DB with embedded Reporting tables. 4](#_Toc150706802)

[Characteristics, Features: 4](#_Toc150706803)

[Compromises 4](#_Toc150706804)

[Cost Factors 5](#_Toc150706805)

[Initial Cost 5](#_Toc150706806)

[Ongoing Cost 5](#_Toc150706807)

[Option 3 – Native Transactional DB replication with adjoint Reporting tables. 5](#_Toc150706808)

[Characteristics, Features: 6](#_Toc150706809)

[Compromises 6](#_Toc150706810)

[Cost Factors 7](#_Toc150706811)

[Initial Cost 7](#_Toc150706812)

[Ongoing Cost 7](#_Toc150706813)

[Option 4 – Transactional DB replication with CDC and adjoint Reporting tables. 7](#_Toc150706814)

[CDC, Briefly 7](#_Toc150706815)

[How does CDC work? The 2 primary ways: 8](#_Toc150706816)

[Some CDC tools 8](#_Toc150706817)

[CDC limitations and opportunities 8](#_Toc150706818)

[The Option now 8](#_Toc150706819)

[Characteristics, Features: 9](#_Toc150706820)

[Compromises 9](#_Toc150706821)

[Cost Factors 9](#_Toc150706822)

[Initial 10](#_Toc150706823)

[Ongoing 10](#_Toc150706824)

[Option 5 – Reporting with ELK 10](#_Toc150706825)

[On ELK 10](#_Toc150706826)

[About ELK 11](#_Toc150706827)

[About PowerBI 11](#_Toc150706828)

[Contrasting ELK and PowerBI 11](#_Toc150706829)

[The option 11](#_Toc150706830)

[Characteristics, Features: 12](#_Toc150706831)

[Compromises 12](#_Toc150706832)

[Cost Factors 13](#_Toc150706833)

[Initial 13](#_Toc150706834)

[Ongoing 13](#_Toc150706835)

[Option 6 – Taking advantage of Delta Lake 13](#_Toc150706836)

[General Recommendation 13](#_Toc150706837)

[Caution about Enterprise data 13](#_Toc150706838)

[Role of virtual databases 13](#_Toc150706839)

# Goal: List Options for Transactional Reporting

A transactional application is at its simplest a CRUD application with UI, APIs, and DB.

In an IT organization very soon one is built, users want to use the data for a variety of reporting purposes.

This reporting requirement can be met in a variety of ways. Not knowing the options and their respective cost factors, business could be easily given an option that is expensive when a simpler option could have satisfied.

This document will list the latest options along with their compromises and provisions.

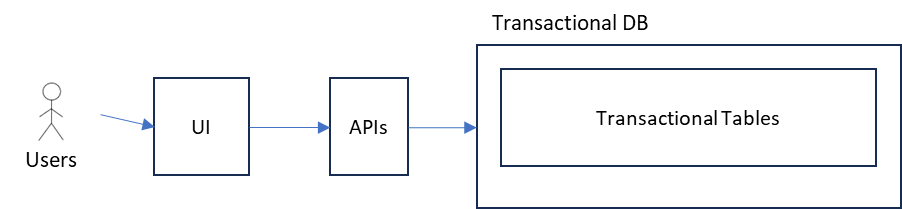
Hope this will expedite the IT decision making process for reporting. It will answer questions like the following:

1. Why is my reporting so expensive?
2. Do we have the right reporting solution that is flexible?
3. Do we have the right reporting solution for time to market?
4. If I add a few extra fields, will they be available for reporting with minimal cost and time to market?
5. What is the cost footprint of my reporting solution?

The options presented will go from the simplest to more advanced ones.

# Option 1 – Use the Transactional DB as is.

Consider a transactional database first:



This is a very simple CRUD application.

Very many times the UI itself could support reporting on the same database using the existing tables.

They are merely screens that are dedicated for administrative purposes with admin logins.

## Characteristics, Features:

1. No specialization for reporting
2. Even if there is specialization a bit, no reporting tools like PowerBi are explicitly deployed.
3. Sub Option: Potentially you can hook up a PowerBi on this database as well.

## Compromises

1. Reporting queries can slow down the primary UI.
   1. This fear is often overstated.
2. Roll up and aggregate tables may be needed requiring some ETL at which time that increased complexity may as well justify a dedicated reporting effort leading to the next option.

## Cost Factors

Cost factors have two dimensions. The initial cost and the ongoing development cost as things change, new fields added, new reports needed etc.

### Initial Cost

As the infrastructure is the same for data and developing the reports the initial cost is 0.

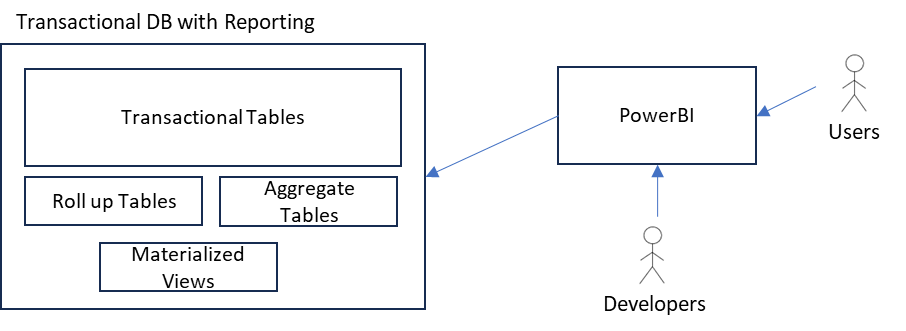
### On going cost of development

1. The UI tools may not be as optimized as a reporting tool like PowerBi
2. Still, the costs can be considered small and in line with the main development costs of a “page”

# Option 2 – Use the Transactional DB with embedded Reporting tables.

In this option, the transactional DB is supplemented by additional tables that are typically required by additional demands of reporting.

Here is what this option looks like



## Characteristics, Features:

1. Roll up tables, aggregate tables, and materialized views are all hosted by the same database.
2. Stored procs, SQL, or any ETL can be used to construct the additional reporting tables.
3. An explicit tool like PowerBI is used for reporting dashboards.
4. Developers can develop canned reports.
5. Users can further customize the reporting dashboards using PowerBi’s capabilities including ad hoc reporting.
6. Sophisticated reporting possible with PowerBi removing that drawback from the previous option

## Compromises

1. Performance impact on the transactional DB
2. ETL, simple or complex, is needed to populate reporting tables compared to option 1. All of the rest of options will carry this burden, however.

## Cost Factors

### Initial Cost

1. Work involved:
   1. New table design
   2. ETL work
   3. Setup PowerBI
2. Cost
   1. ETL can be fairly simple ETL that can be done through SQL, Stored Procs, SSIS etc.
   2. A person month is reasonable to setup the tables: 1 PM
   3. Another man month to obtain and hook up the PowerBI
   4. So, 2 PMs for that initial setup cost

### Ongoing Cost

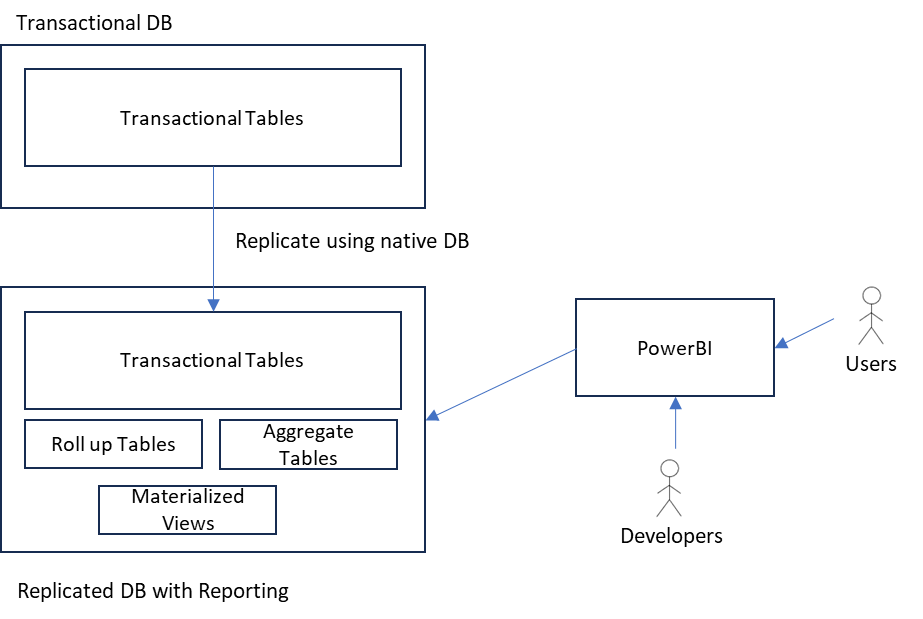
1. PowerBI is quite quick in developing reports
2. For each new report it can be as simple as a day
3. On the average though, 1 PW (Person week) per Report
4. And the licensing and purchase cost of PowerBi. Still very reasonable as this tool is often used by small and midsize companies.

# Option 3 – Native Transactional DB replication with adjoint Reporting tables.

In this option remove the reporting load from the transactional DB by replicating the transaction tables using the native database replication mechanism.

Then add the additional 20% or so reporting tables and use any simple ETL to populate those secondary reporting tables.

This is shown in the following picture.



## Characteristics, Features:

1. Replicate the transactional DB using native database replication.
2. Attach the reporting tools to the replicated DB.
3. Add the additional (if necessary) roll up and other reporting tables to the replicated DB.
4. Performance impact on transactional DB removed.
5. **This is generally a very good “standardizable” ground! So to say a sweet spot for transactional application reporting needs.**

## Compromises

1. A reporting database may not need to have all the transactional tables or their content.
2. The ETL, simple or complex, have to track the source database in a non-trivial way to populate the secondary tables. Doing this without the help of a tool can become error prone.
3. One might not design the optimal reporting data as they rely on the existing schema as much as possible.
4. In other words, as the percentage of tables needed for reporting increases beyond 30%, may indicate a need to create an entirely dedicated schema for reporting

## Cost Factors

As this is the option that suits most use cases, you may want to pay attention to this.

### Initial Cost

1. Work involved:
   1. License the replication options
   2. Setup replication
   3. Schema for additional tables
   4. ETL for additional tables
   5. Setup PowerBI
2. Cost
   1. Setup replication 1PM
   2. ETL and schema for additional tables: 2PM
   3. Hook up PowerBI: 1PM
   4. Total: 4 PM

### Ongoing Cost

1. PowerBI is a quick tool for developing reports
2. For each new report it can be as simple as a day
3. On the average though, **1 PW (Person week) per Report**
4. And the licensing and purchase cost of PowerBi. Still very reasonable as this tool is often used by small and midsize companies.

# Option 4 – Transactional DB replication with CDC and adjoint Reporting tables.

Sometimes or often reporting databases may require a different schema than their transactional equivalents.

In a large IT organization, the data teams often assume this and design as such causing a significant cost.

However, there are times when such a new schema is better.

Using ETL to replicate a database in that manner into a new schema can be problematic if one doesn’t pay attention to the changes as opposed to full ETL every time.

It is under these circumstances that one adapts a CDC (Change data capture) architecture.

First let us understand what CDC is, how it works, what are the tools and the limitations and opportunities.

## CDC, Briefly

Capture only the changes in a source database and update the target database with those changes. There is coding and configuration required to do this.

### How does CDC work? The 2 primary ways:

#### Triggers and Change Tables

In this method each table is tracked for changes and programmed between source and targets.

The key lies in the source.

For every table a duplicate table is created which is populated on a trigger on the source table along with the record information and a time stamp.

At any given frequency the change table is then used to copy over to the target.

Very cumbersome for many tables or a full database.

#### Log Based

In this method one needs a tool that uses database logs to capture changes.

No need to track each table via creating change tables.

One writes code through a tool to pick up the changes to populate target tables that are in any format or spec. They can have the same schema or a different schema.

It depends on the tool how simple or complex the coding is depending on the target schema.

This approach also allows you to customize or get tuned to the new schema, however programming is necessary.

### Some CDC tools

1. MS SQL Server CDC
2. Oracle Golden Gate
3. Debezium (Open Source)

There may be other tools. A tool is almost always necessary to do this.

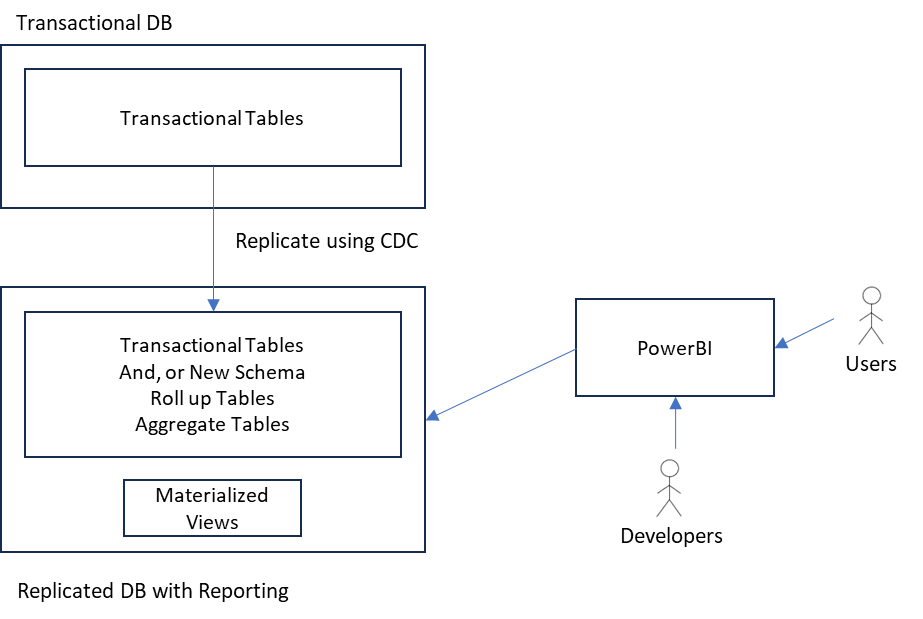
### CDC limitations and opportunities

Depending on the target schema, even the log based CDC replications requires coding and planning.

However, it has the benefit of being better than managed ETL coding.

## The Option now

The picture looks very similar to the replicated database picture of the previous option, it is just that how the data is replicated.



## Characteristics, Features:

1. Replicate the transactional DB using CDC tools, preferably using log based approach
2. Create a new schema that contains a combination of transactional, rolled up, aggregated and other reporting specific tables.
3. Use CDC inputs to write ETL to populate the target schemas.
4. Some of the ETL could be provided by the CDC tool itself.
5. Performance impact on transactional DB removed.
6. Accommodates flexible desired schema for reporting.
7. Suitable when reporting tables significantly diverge from transactional tables.
8. Sub Options
   1. Reporting schema closely tracks the original (say 20% variation)
   2. Reporting schema dramatically different (say 50% variation)

## Compromises

1. Expensive. Because a) new schema b) CDC tools c) CDC ETL
2. Any changes to the transactional database if needed in the reporting database will incur development costs mainly because the target schema is by necessity different from the source.
3. In this option the expenses CAN and likely go up with the schema size of the database

## Cost Factors

This is the Gold option. Does not have too many limitations. However it could cost for small companies without bureaucracy as much as $250,000 to a large company $1M to $2M.

### Initial

1. Work
   1. Assumes log based.
   2. Design the target schema.
   3. Setup the CDC tool pointing to log files and giving access to the logs
   4. Write the necessary code to populate the target schema based on CDC logs
2. Cost
   1. Design the target schema – 3 PM
   2. Setup CDC – 2 PM
   3. Write CDC ETL into the target – 6 PM
   4. Setup PowerBI and hook it up to the target – 1PM
   5. For an average database with 50 tables total: 12 PM
   6. Further it can increase with the number of tables that needs to be in the target database

### Ongoing

1. Work
   1. Adjustment cost
      1. Adjust the target schema if needed.
      2. Write the CDC ETL
      3. Likely 1 PW to a 1 PM for each adjustment
   2. Ongoing reporting cost
      1. Write the reports in PowerBI – 1 PW per Report

# Option 5 – Reporting with ELK

Let’s understand the features and suitability of ELK first

## On ELK

Elastic search is a data storage and search mechanism for content, especially text. Furthermore, it indexes all fields, and very efficient at locating a set of rows (or documents) given even partial fields.

So, it is really suited for google search like open queries on large sets of data.

ELK stands for Elasticsearch, Logstash, and Kibana.

Logstash is how you move data into Elastic search.

Kibana is the web-based dashboard to author and see reports.

ELK is often used for monitoring reports of real time events and time slicing.

It is also useful for unstructured data where the documents you are searching, form a varied collection.

### About ELK

ELK is commonly used for log and event data analysis, monitoring, and searching. It excels in handling large volumes of unstructured or semi-structured data.

ELK is flexible and scalable, making it suitable for scenarios where data may be rapidly changing, and where real-time search and analysis are critical.

### About PowerBI

Power BI is commonly used for business intelligence, reporting, and analytics. It's well-suited for scenarios where data is structured, and users need interactive and visually appealing reports.

Power BI is highly flexible and integrates well with various data sources. It's often used for traditional business reporting and analysis, especially when working with structured data.

### Contrasting ELK and PowerBI

ELK is well-suited for unstructured or semi-structured data, such as logs and events. Power BI is more aligned with structured data commonly found in relational databases.

ELK is often chosen for IT operations, security monitoring, and log analysis. Power BI is commonly employed for business reporting, dashboards, and traditional BI use cases.

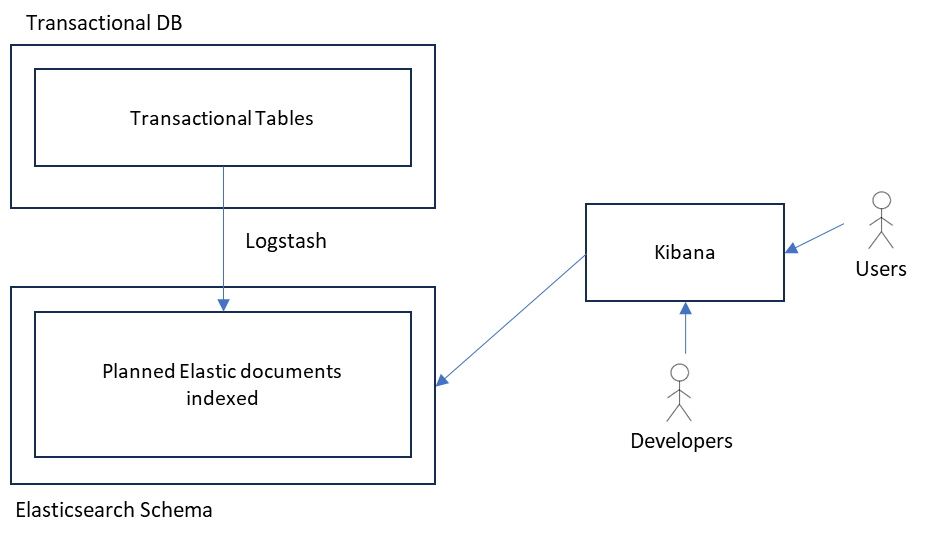
ELK might require more configuration and setup, especially when dealing with complex data processing pipelines. Power BI tends to be more straightforward in terms of implementation.

## The option

In this option the reporting schema is reimagined as a set of Elasticseaerch indexes. Each index represents a set of documents or searchable entities (or tables) that you can slice and dice.

So it is a new schema different from the original.

Here is what the architecture looks like



## Characteristics, Features:

1. Reimagine the schema for reporting purposes.
2. Procure and configure the distributed ELK infrastructure.
3. Configure the elastic instance with the necessary indexes and documents (like tables)
4. Use Logstash and other ETL tools to capture changing data and move that to Elastic.
5. Use Kibana to author reports.
6. Searches are fast.
7. Slicing and dicing is easy.
8. Data can be large.
9. Open ended text search in locating reports.
10. **Sub Option –** Sometimes the ELK infrastructure might already be in place for text search reasons. In those cases the ELK reporting can be paired with any of the previous options for accommodating the secondary reporting tables in ELK.

## Compromises

1. Expensive. Because a) new infrastructure b) distributed complex infrastructure c) new schema b) a non-relational schema
2. IT will have to develop the reports and Business will have a less understanding of the underlying data structures.
3. Ongoing changes take time to implement.
4. Likely more expensive than the CDC replicated PowerBI option.
5. Resources and skill sets are harder compared to PowerBI.

## Cost Factors

It is likely that the initial cost will be 50 to 100% more compared to PowerBI if the requirements can be met by both. Resulting in from $500K to $2M for a full implementation.

In some cases, the requirements of events and logging will force you to ELK and you won’t have an option.

### Initial

1. ELK Infrastructure – 12 PM
2. New schema provisioning in ELK – 3 PM
3. Total: 15 PM

### Ongoing

1. Adjust new schemas – 1PM
2. Develop reports. 1PW for each report

# Option 6 – Taking advantage of Delta Lake

With the rise of delta lake in the cloud, it presents another compelling option for Reporting.

This is like the data replication directly or via CDC into a delta lake.

The delta lake then can be used both for reporting and analytics and ETLs etc. A multipurpose mechanism.

# General Recommendation

For most scenarios Option 3 (Natively Replicated DB with adjoint reporting tables) is in the goldilocks zone.

## Caution about Enterprise data

Note though, this recommendation at this point has not considered the need to aggregate data from other data sources in the enterprise. Such a solution can be imagined based on the current recommendation with a virtual view on top.

In the next release of this document will explore that option in more detail.

# Role of virtual databases

With most of these options you can still use technologies like MS Polybase to map tables from various data sources to finally drive the reporting.