powered by

traferium newwave

Migrating Oracle to Snowflake

*NewWave's Data Science Team leader, **Patty Delafuente**, maps the necessary considerations on getting your data to the cloud with Traferium.*

Why Migrate?

There are many excellent reasons to migrate a data warehouse from a traditional database management system such as Oracle to a cloud-based data warehouse solution such as Snowflake. Some of these reasons are reduced hardware maintenance, improved performance for large data processing, and the ability to scale out.

Technology Differences

Migrating from Oracle to Snowflake is not always a straight line. Modern Data Warehouses like Snowflake are engineered differently from the ground up to support business intelligence and analytic workloads. There are challenges because of this and before we discuss how to address them let us first talk about the differences between the two types of data warehouses.

The migration journey from Oracle to Snowflake is not always a straight line.

I refer to traditional data warehouses as those that are features added to relational database management systems such as Oracle, Microsoft SQL Server, and MySql. These systems are engineered ensure high data integrity, fast row-level read and write operations, low latency, transaction management and minimal data redundancy. They are tried and true for transactional needs with a mature technology stack that has stood the test of time over the past several decades.

However, the same things that make this technology wonderful for row-level operations and high data integrity, also make it less efficient for analytic and business intelligence needs. With traditional data warehouses, you typically would buy a high-end server with as much CPU and RAM as the budget allows and attach the data storage to a Storage Area Network (SAN) server. One cannot scale out compute operations across multiple servers. One scales up by adding more memory and processing power to the existing server or upgrading to a better one.

Traditional Data Warehouse Performance

Back in my database administrator days, I would spend a lot of time building and rebuilding indexes based on the most common query search arguments. By this I mean those key fields users might search on such name or city. If a different field that was not indexed is used in a query, that first query would take much longer to run, but subsequent runs are usually not as bad -- as the data is cached in memory. So data that is not cached or indexed is a costly operation.

Additionally, transactional databases are highly normalized which is essential for data integrity as it prevents data modification anomalies and reduces data redundancy. This results in a lot of tables that need to be joined together for reports. For large databases this can be expensive performance-wise and complex to maintain. The more data you have in your tables, the worse the performance for very large databases, there is a

point where even indexes and scaling up does not help.

For those very large databases, 2 methods that database administrators used to improve performance were:

- Batch jobs to join the data and copy into long, denormalized tables specific to reporting needs. Oftentimes, these scripts are complex, compute intensive and run at night to refresh and update the data routine to ensure reports had timely information. It was staff resource intensive to maintain but it worked to provide reports with decent performance.
- Online Analytic Processing (OLAP) Cubes is a feature that the relational database vendor added to their products that enabled you to create multidimensional views of data. It required that you have identified measures and dimensions to connected data sources and when you process the cube, it queries the data sources and aggregates the data by the specified measures and groups data by the dimensions. This was a step-up from the other method and provided some flexibility in that you could drill into the data by dimensions.

Modern Analytics Data Warehouse Performance

Modern Data Warehouses do quite a few things differently that enhance the performance of business intelligence and analytic operations. At its core, the data is stored differently by columns rather than indexed rows of data. This allows for fast aggregations for reports. Another key factor is that these systems distribute data across data nodes and rather than scaling up in hardware, you can just add more nodes. You no longer need the premium, expensive Storage Area Network servers combined with a high-end database management system server. You can span your data across low-cost hardware.

The compute operations are also scaled out across compute nodes. Because Snowflake is a cloud-based system, you do not have to worry about managing the hardware and you can additionally scale up and down as needed. If you have more data, you just add more storage which equates to more data nodes in the background. If you have a big, resource intensive process to run, you can add more compute resources and then remove them when your process is done. Most distributed database can provide similar performance in queries without having to preprocess into OLAP cubes.

Table 1: *Major differences between the 2 types of data warehouse systems.*

Oracle – *Relational*

- Scale up using expensive hardware (high end servers- CPUs and large memory cache)
- Performance decreases as number of rows and columns increase
- Optimize data retrieval by adding indexes
- Optimized for row level operations- index and cache misses are costly
- Optimize reporting applications by transforming into dimensions, measures, and cubes (OLAP) or ETL scripts to flatten data

Snowflake – *Distributed*

- No need to preprocess data into OLAP Cubes or denormalize, just write your queries
- Data scales out across low-cost hardware
- Performance is maintained by adding more nodes as data increases
- Data is stored by columns which optimizes aggregation and analytic operations
- Index operations are not required
- Data retrieval and reports are optimized by distributing queries across compute/data nodes

Migration Challenges

Now that we have discussed the differences between relational Data Warehouses and the modern distributed Data Warehouses let's review how that applies to moving to Snowflake for business intelligence and analytic databases. Our work shows that there are a few pain points one may encounter during the migration.

Application data must be re-engineered:

- Database vendors such as Oracle have adapted the SQL Language to include specific key words and functions that are specific to their product. Oracle's SQL language is referred to as PL/SQL and Microsoft's is Transact-SQL. Scripts and stored procedures that make heavy use of those specific key words and functions must be rewritten
- Database vendor specific Objects might not be supported in new platform – certain features such as stored procedures and sequences are not supported and must be rewritten
- OLAP Cube Services are not supported as they are not needed. You can just write the query and add sufficient compute resources as required for performance

Other Migration Tools

Distributed databases are emerging technology so there are not a lot of migration and vendor tools out there to address the challenges.

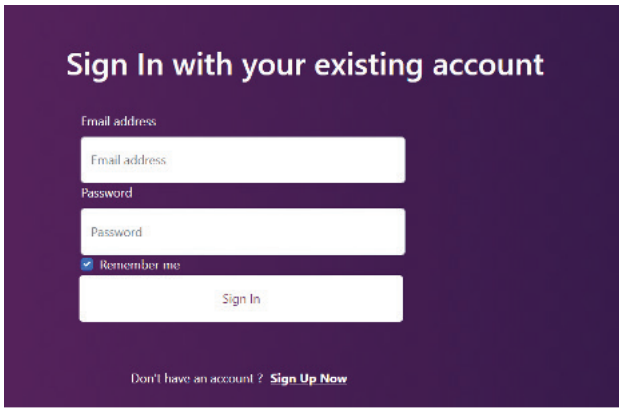
Today's market offers many data migration tools that enable migration of tables and data from one database system to another. These tools include Ab Initio, Azure Data Factory, and AWS Migration Services. However, these tools do not address the challenges that are listed above. They will migrate the tables and data but will not perform the code conversion. Depending on the complexity, some of those individual procedures or objects that need to be converted can be time consuming to rewrite manually.

Traferium

This is where Traferium steps in to address those migration pain points. Traferium is an all-in-one data migration and code conversion product. Traferium migrates your tables and data and it also does much more than that. Traferium extends beyond existing data and schema migration tools to support migration of stored procedures, functions and will convert the PL/SQL scripts to a format that can be utilized in Snowflake.

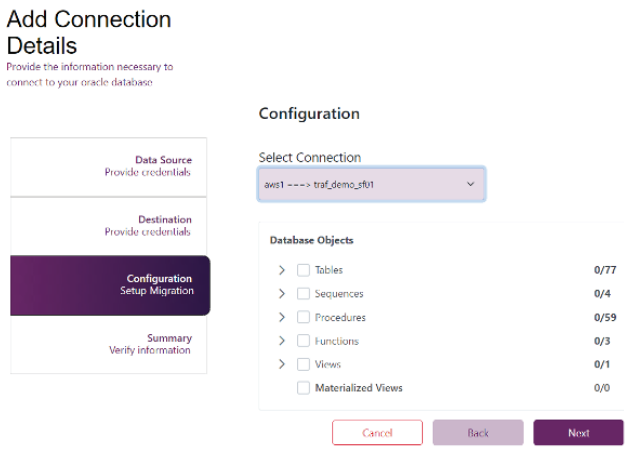
Traferium is an all-in-one data migration and code conversion product.

Figure 1: *Traferium log-in screen*



Traferium has a user-friendly web interface that allows you to add in multiple Oracle connections, select tables, views, stored procedures, functions, and sequences that you wish to migrate *(see figure 2)*.

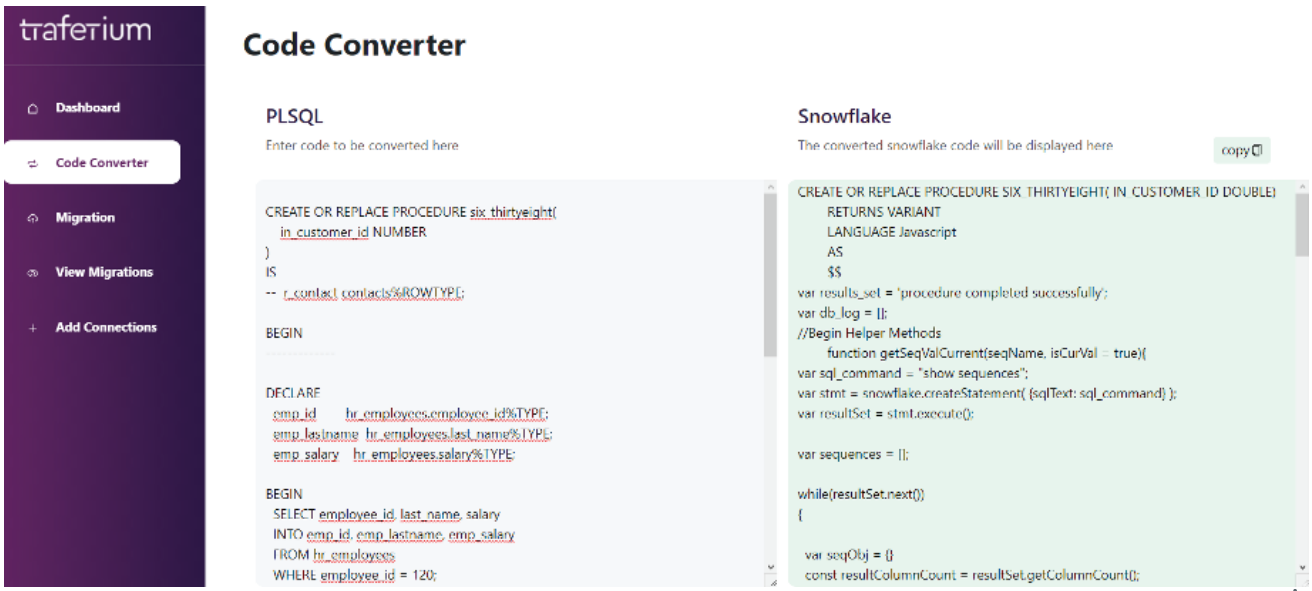
Figure 2: *Traferium connection and object-selection*



Behind the scenes, Traferium connects and retrieves all the information about the schema, tables and objects and uses a parsing engine to parse through all the metadata. It then takes this metadata and feeds it into a transpiler that then maps and converts the objects to Snowflake equivalents.

Traferium also provides an interface where you can directly copy and convert a PL/SQL script and then execute the script in Snowflake (see figure 3). Currently, Traferium only supports migrating from Oracle to Snowflake but stay tuned as we have plans to add other data sources on our roadmap.

Figure 3: Traferium code converter



This article outlined the key differences between Data Warehouse technology for the traditional, relational database management systems in comparison to modern, distributed data warehouses. Migrating from a relational system, such as Oracle, to Snowflake has lots of advantages to improving performance and maintenance of Business Intelligence and Analytic applications. There are also challenges when migrating propriety vendor scripts and objects to Snowflake which can be minimized by leveraging a code conversion tool such as Traferium.

For more information about Traferium, visit us online at **Traferium.com**

The talented Traferium Team would love the opportunity to show you our product and help ease the challenges as you migrate to Snowflake.