WHITE PAPER

Qlik Cloud Data Integration-Data Movement

Defining Low-Latency Data Processing with Qlik Cloud Data Integration Services

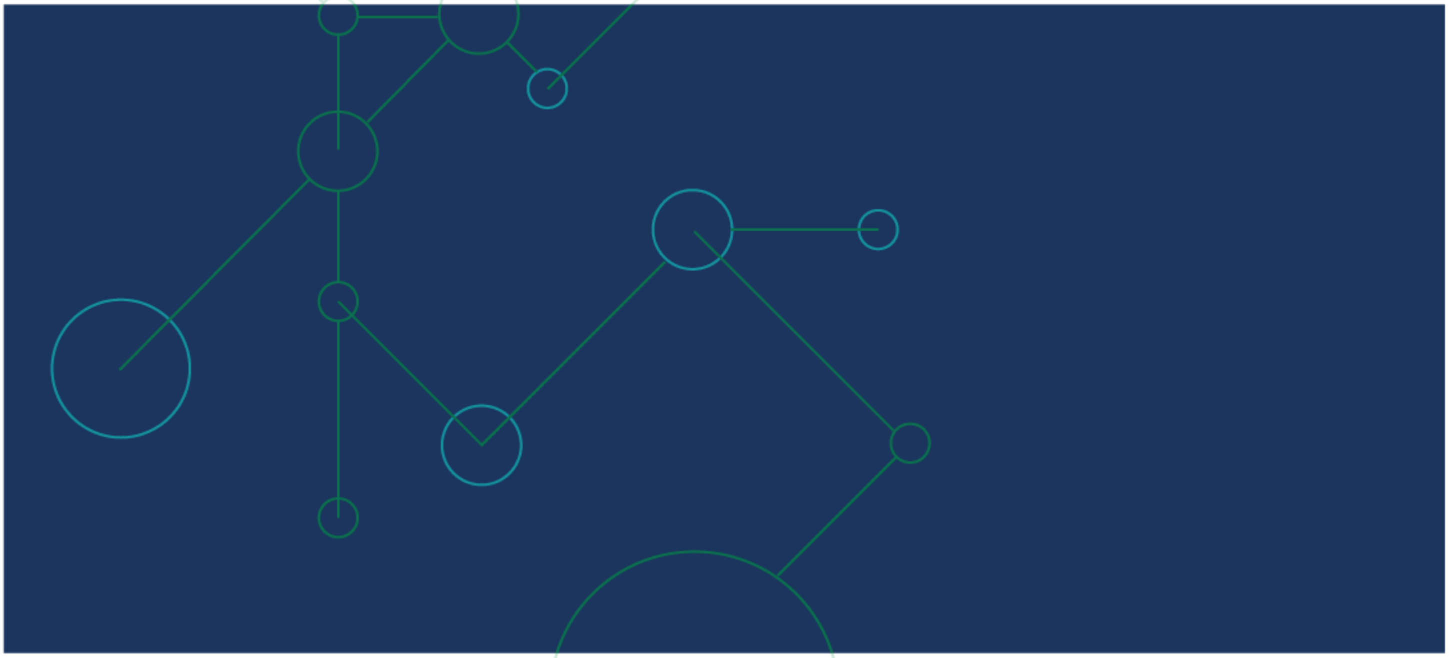


TABLE OF CONTENTS

SUMMARY

- Qlik Cloud Data Integration data movement tasks can be used to continuously land data from on-premises enterprise data sources and apply changes to keep data up to date with low latency.
- Qlik Cloud Data integration offers a SaaS solution to move data to the cloud and allow organizations to reduce their overall compute resources.

INTRODUCTION

As the need for data movement to the cloud grows, organizations want to leverage SaaS platforms to move data to cloud targets with low latency. The push to the cloud has left a lot of organizations balancing high costs from cloud platforms and the need for real-time data changes. Utilizing the Qlik Cloud Data Integration platform, organizations can manage low-latency data movement requirements efficiently and lower their overall cloud platform costs.

The Qlik Cloud Data Integration SaaS architecture can be used to migrate data from enterprise data sources to the cloud and keep them in-synch using change data capture (CDC). The solution offered by Qlik allows the organization to determine when to apply changes and that those changes are orchestrated for data transformations or direct consumption.

The following will describe how Qlik Cloud Data Integration migrates data from enterprise data sources to cloud-supported targets with low-latency and a lower compute imprint on the cloud target.

Consumers of this document should have a basic understanding of Software as a Service (SaaS) and Change Data Capture (CDC).

Qlik Cloud Data Integration architecture for Data Movement

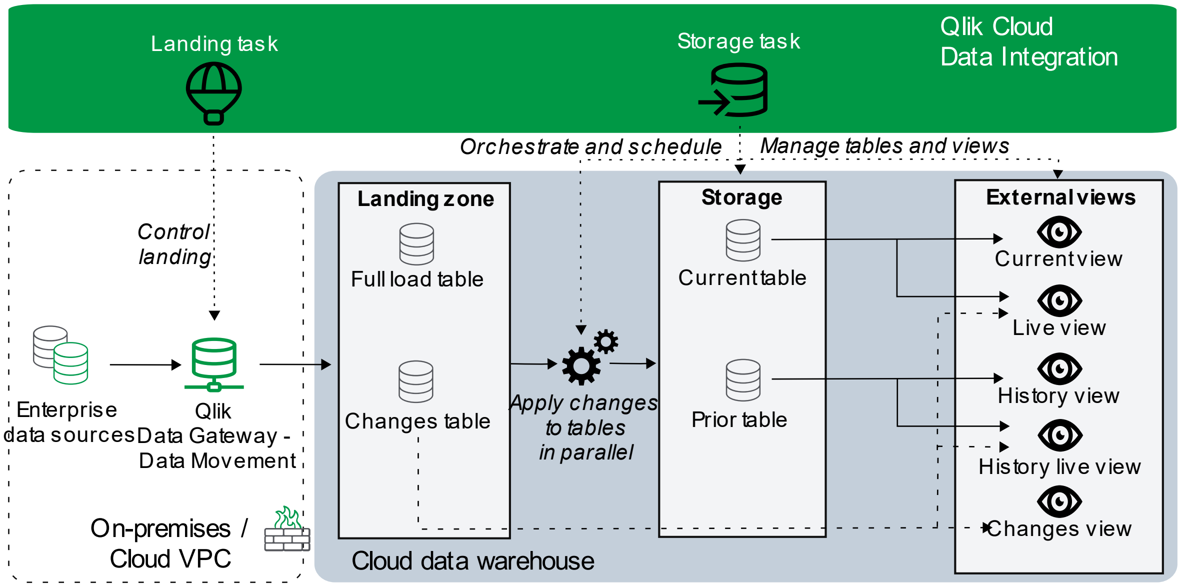


Figure 1- Qlik Cloud Data Integration data pipeline using Qlik Data Gateway Architecture

The architecture above shows the pipeline of data moving to a cloud data warehouse using the Qlik Data Gateway for Data Movement. Qlik Cloud Data Integration supports cloud native source integration in addition to moving data from supported on-premises sources through the data gateway. Onboarding data in a Qlik Cloud Data Integration project leverages two tasks for data ingestion. The landing task will deliver the full raw data and changes from the source. The storage task will apply the changes to tables and views. Live views are supported in the storage task for consuming landed data without applying the changes directly to the table.

The architecture solution of separating data movement into landing and storage zones has some high-level benefits. One significant benefit of this solution enables is that it supports a delayed merge which lowers compute cost on the initial raw data ingest. Consumers will have access to the new data changes in the low latency live views without applying a merge to the data. Consumers can than apply compute for merge on a schedule that is cost beneficial without losing consumption access to new data changes. Consumer datasets are abstracted from landing to reduce impact of data re-loads. Landed data can be separated into separate storage layers for security. Which can encapsulate raw data from applied changes to allow consumers a constructive way to govern data assets. In addition to cost benefits, the initial data loading performance will benefit since all changes will first arrive in the landing zone as an insert statement into the change table. No extra compute is needed to merge changes on the initial data load.

The architecture solution can integrate with many modern data architectures. Which allow for raw data to be captured in an initial zone before applying changes to create serving zones for consumption. Thus, the architecture is very flexible for a consumer to adopt for data movement.

Qlik Cloud Data Integration Landing Task

Once the Qlik Cloud Data Gateway is connected, it will land data from your data sources into tables created in the cloud target landing zone. When the landing task is configured with the “Full load and Change Data Capture (CDC)” update method, it will maintain both a full load and changes table for each source object being replicated (Note: The less frequently used “Full load” update method, manages full load tables only.) With this architecture approach, changes committed to the source’s transaction logs after the full (initial) load are landed in the change tables in near real-time. During CDC the changes are delivered as inserts into the change tables (with timestamps and operation indicators). This reduces the compute required for data ingestion and the delivery interval to the cloud target since no applied operations are executed during the landing task

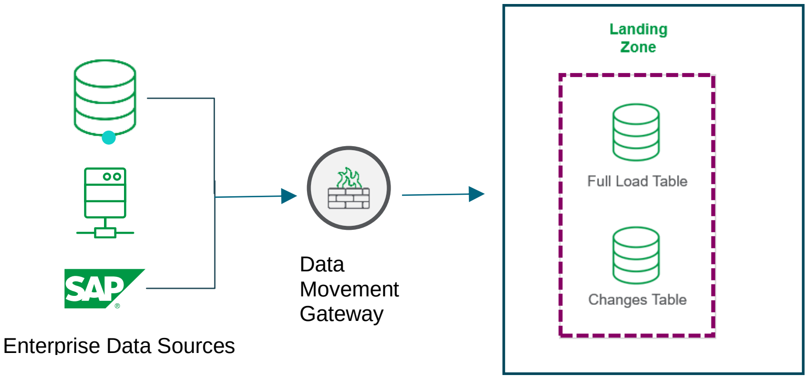
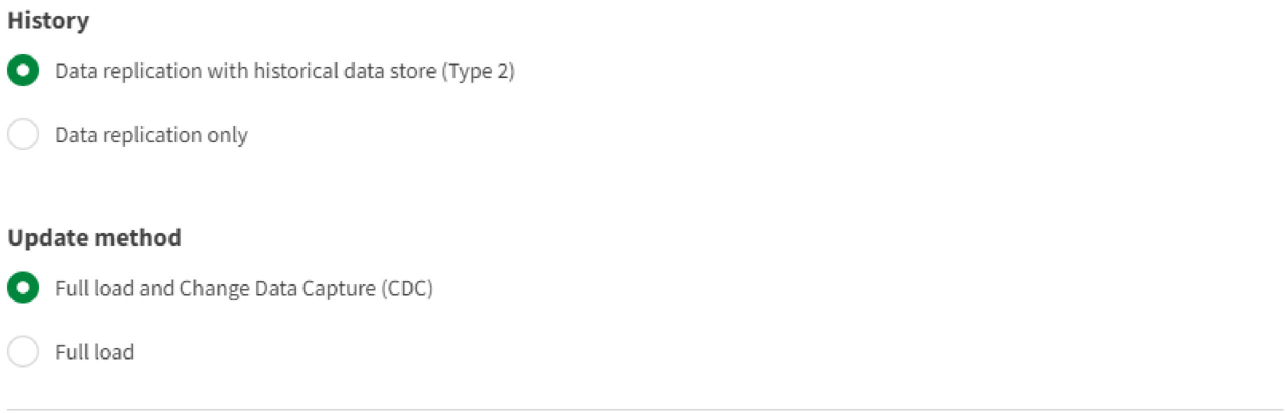


Figure 2- Qlik Cloud Data Integration data pipeline using Qlik Data Gateway to Landing Zone

Landing Task Tables



History

● Data replication with historical data store (Type 2)

○ Data replication only

Update method

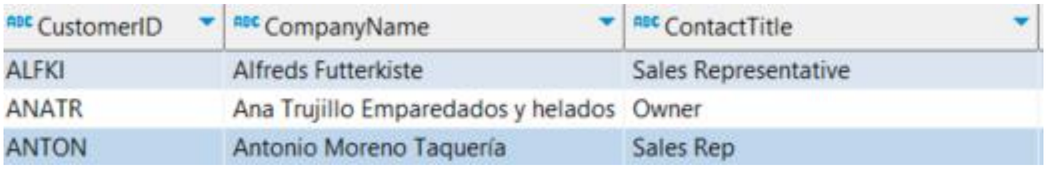
● Full load and Change Data Capture (CDC)

○ Full load

Figure 3- Qlik Cloud Data Onboard Task Configuration

Let us explore the tables created and managed by the Landing Task when the update method includes CDC. Within the landing zone, the full load table captures the initial load of data from the source system. The change tracking table (suffixed with '__ct') contains the changes captured and inserted by the data gateway.

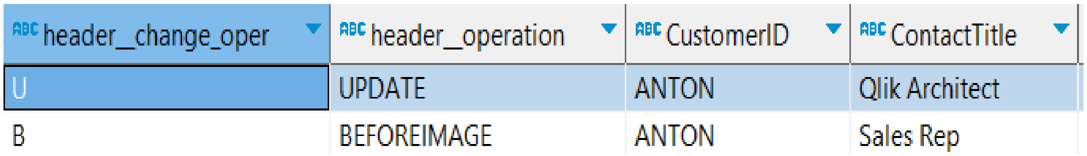
FULL LOAD for Customers Table



CustomerID	CompanyName	ContactTitle
ALFKI	Alfreds Futterkiste	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Owner
ANTON	Antonio Moreno Taquería	Sales Rep

Figure 4- Landing Task Full Load Customers Table

CHANGES for Customers Table



header__change_oper	header__operation	CustomerID	ContactTitle
U	UPDATE	ANTON	Qlik Architect
B	BEFOREIMAGE	ANTON	Sales Rep

Figure 5- Landing Task Change Customers Table

The Landing Task is responsible for landing the data in the cloud target as quickly as possible with as little compute requirement as possible.   Most consuming applications need to have the changes applied to the target datasets to provide a ‘current view’ of data or a historical type 2 view of data. The Storage task is responsible for that functionality. Separating landing and processing of the data provides the flexibility to define processing intervals with an eye on cost and compute requirements and insulates data consumers from re-load activities performed by the landing task.

Qlik Cloud Data Integration Storage Task

During the onboarding of data from source, a storage task is used to apply the changes landed in the target to the long-term persistent tables. The Storage Task is responsible for managing current (ODS) and, optionally. historical (type 2) data structures via a combination of tables and views by orchestrating the ELT process that applies changes and manages the type 2 history. Qlik Cloud Data Integration provides a ‘delayed-merge’ feature that supports low-latency SLA’s without the need to continuously run *MERGE* processes and apply the changes. This is enabled via *Live Views* (described later in this paper). The storage task processes data from landing into the long-term storage layer and applies inserts / updates /deletes captured by the data gateway to the target tables. It generates consumption views of the ‘applied’ tables as well as live views which combine applied data with un-processed changes for a low-latency view of current data.

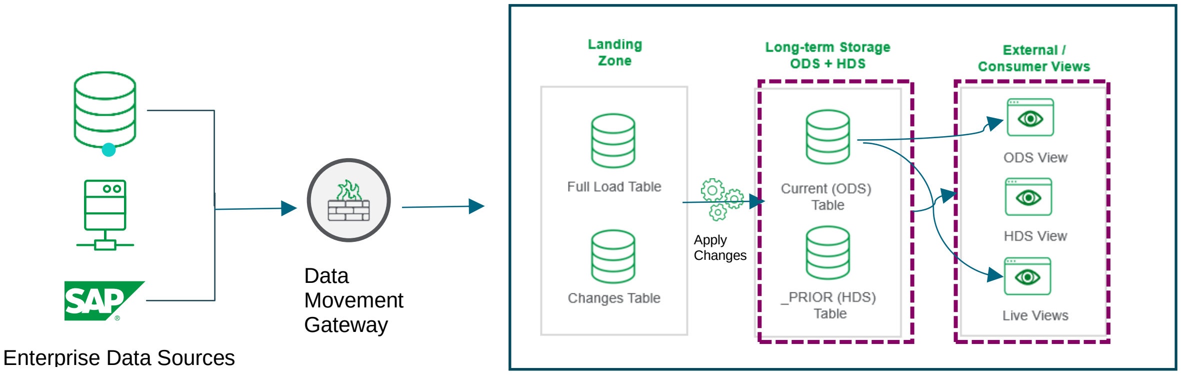


Figure 6- Qlik Cloud Data Integration Storage Task Data Flow

Storage Task

Within the storage task, current and historical data can be applied to tables and views at a specified interval within the task, in addition to the current snapshot of the merged full load and change tables inherited from the Landing Task. A snapshot of the prior data is created to store the history of the table before the incoming landing changes have been applied. The internal schema asset state table will be updated to reflect the storage task status at execution time. An Historical Data Store (HDS) view will be created and maintained to reflect the history of the data set as data changes are applied from the landing task change tables. Utilizing this approach to onboard data can reduce compute engine costs, since history can be applied at the individual table level by enabling the HDS setting for a table in the storage task or by inheriting that setting from the data asset in the Landing Task. There is no need to store all the table history from an onboarded data source. The history view will reflect the state of the data change. The historical data will have a status of prior in the view and new changes will have a status of current.

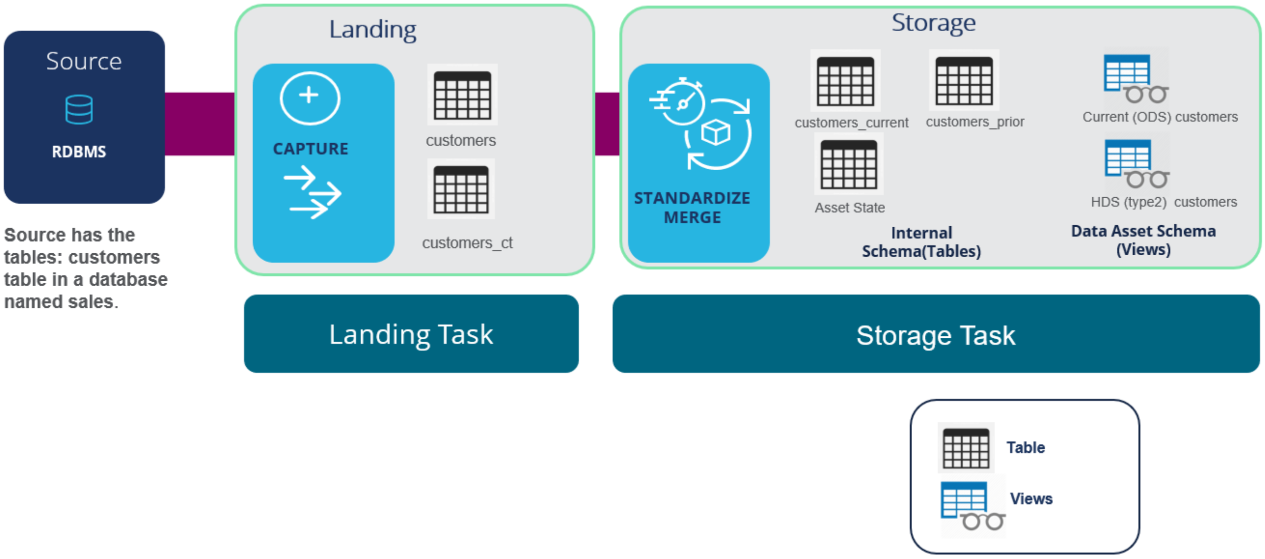
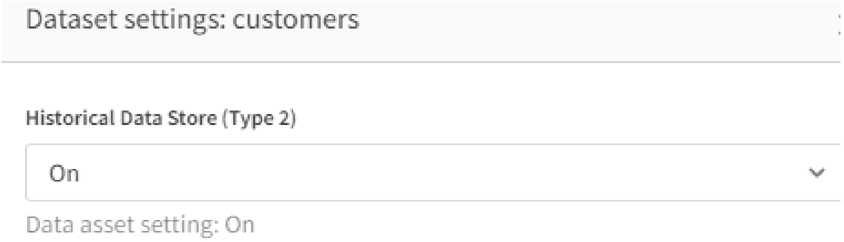


Figure 7 - Qlik Cloud Data Integration Onboarding Current and Historical Data



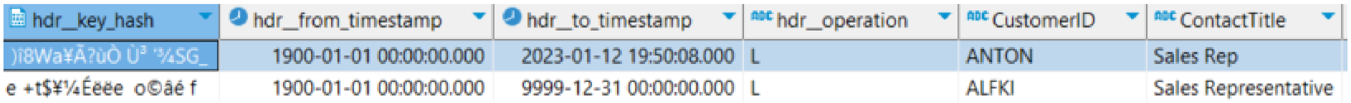
Dataset settings: customers

Historical Data Store (Type 2)

On

Data asset setting: On

Figure 8 - Qlik Cloud Data Integration Storage Task Dataset History setting



hdr__key_hash	hdr__from_timestamp	hdr__to_timestamp	hdr__operation	CustomerID	ContactTitle
)ï8Wa¥Ã?ùÒ Ù³ '¾SG_	1900-01-01 00:00:00.000	2023-01-12 19:50:08.000	L	ANTON	Sales Rep
e +t$¥¼Éëëe o©âé f	1900-01-01 00:00:00.000	9999-12-31 00:00:00.000	L	ALFKI	Sales Representative

Figure 9 - Qlik Cloud Data Integration Storage Task Data Asset Schema Customers History view

Storage Task Live Views

Within the storage task, Live Views can be enabled for an entire data set or individual tables. Live Views provide a low latency data integration solution by providing a view of the delayed merge of change data from the landing task change tables. The target compute engine doesn't need to execute to apply changes during the storage task for the changes to be accessible in the Live Views. The Live Views combine the current and prior data, with the un-processed changes in the landing change table. (The storage task will create and maintain a Live View for each table that will capture the latest changes before and after the changes are applied. If history is enabled, a Live View will be created for the historical (HDS) table and current (ODS) table. The internal schema asset state table will be updated to reflect the storage task status at execution time. The live views architecture has been validated with cloud partners and is a viable solution to optimize compute and provide low-latency access to source data changes.

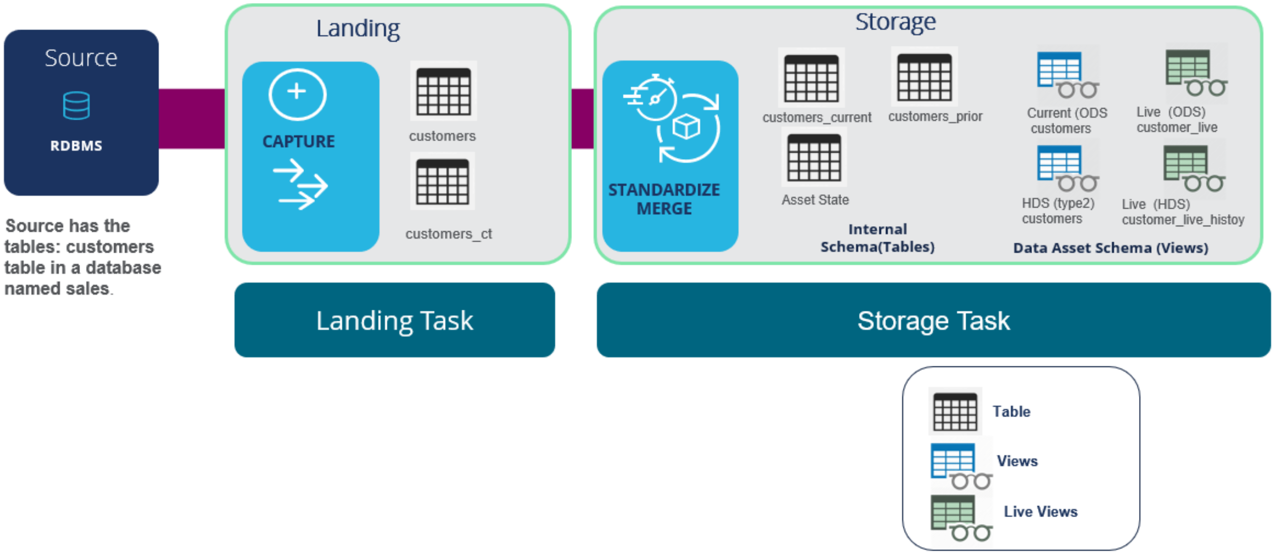
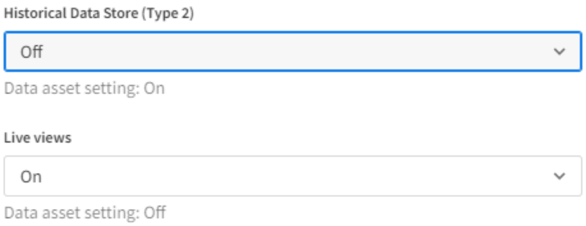


Figure 10 - Qlik Cloud Data Integration Storage Task Data Asset Schema Live Views



Historical Data Store (Type 2)

Off

Data asset setting: On

Live views

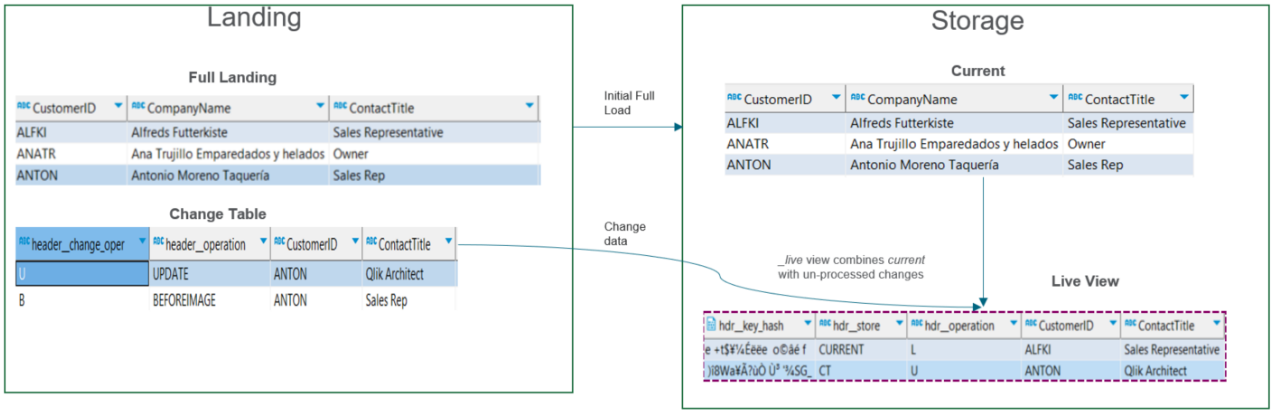
On

Data asset setting: Off

Figure 11 - Qlik Cloud Data Integration Storage Task Dataset Live View setting

Please note that for simplicity in this example we have turned off the Historical Data Store (HDS) and will be only focusing on the Operational Data Store (ODS), but the same basic paradigm holds for the historical data use case.

After Initial Full Load and changes- (Live View shows CT for new change)



Full Landing

CustomerID	CompanyName	ContactTitle
ALFKI	Alfreds Futterkiste	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Owner
ANTON	Antonio Moreno Taquería	Sales Rep

Change Table

header_change_oper	header_operation	CustomerID	ContactTitle
U	UPDATE	ANTON	Qlik Architect
B	BEFOREIMAGE	ANTON	Sales Rep

Current

CustomerID	CompanyName	ContactTitle
ALFKI	Alfreds Futterkiste	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Owner
ANTON	Antonio Moreno Taquería	Sales Rep

Live View

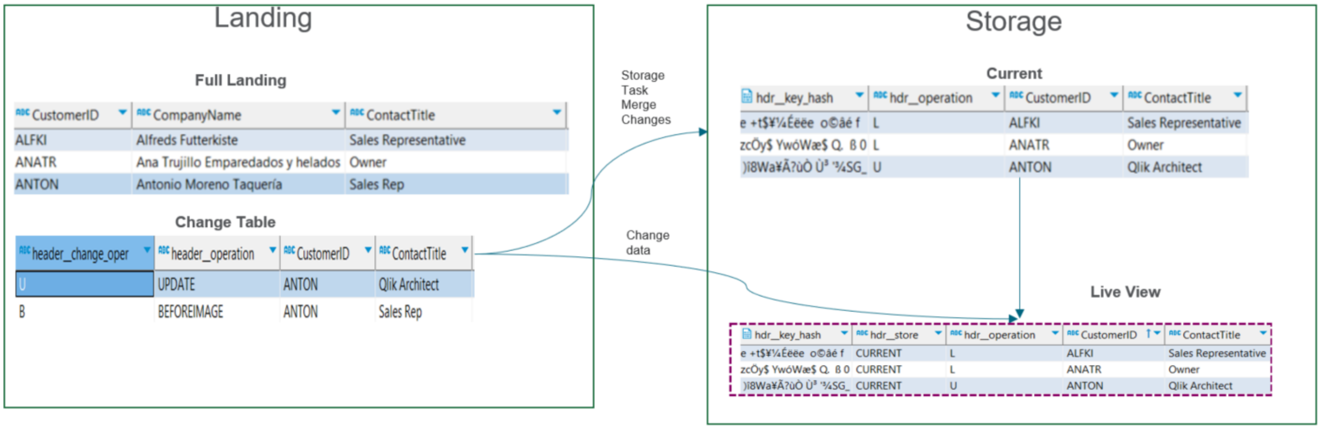
hdr_key_hash	hdr_store	hdr_operation	CustomerID	ContactTitle
e +t$¥¼Éëëe o©âé f	CURRENT	L	ALFKI	Sales Representative
)i8Wa¥Ã?ùÒ Ù³ '¾SG_	CT	U	ANTON	Qlik Architect

Figure 12 - Qlik Cloud Data Integration Storage Task Live View before Applied Change

The snapshot in Figure 12 shows a delayed merge and how the live view combines data from the current view in the storage zone with unprocessed data from the change table in the landing zone. Please note that while the full landing and change tables shown in the landing zone are physical tables, all queries of the storage zone shown in this example use views. The physical table that the current view mirrors is maintained in the internal schema and omitted from this example.

In Figure 12, the update of ContactTitle to ‘Qlik Architect’ from ‘Sales Rep’ for Customer ID ‘ANTON’ has been landed, but not yet processed into storage. The live view allows changes to be seen in the storage zone before spending compute to process them physically, which provides the means to find an optimal balance between reduced latency and compute processing costs.

After Changes Applied in Storage Task- (Live View shows Current)



Full Landing

CustomerID	CompanyName	ContactTitle
ALFKI	Alfreds Futterkiste	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Owner
ANTON	Antonio Moreno Taquería	Sales Rep

Change Table

header_change_oper	header_operation	CustomerID	ContactTitle
U	UPDATE	ANTON	Qlik Architect
B	BEFOREIMAGE	ANTON	Sales Rep

Current

hdr__key_hash	hdr__operation	CustomerID	ContactTitle
e +t$¥¼Éëëe o©âé f	L	ALFKI	Sales Representative
zcÖy$ YwóWæ$ Q. ß 0	L	ANATR	Owner
)î8Wa¥Ã?ùÒ Ù³ '¾SG_	U	ANTON	Qlik Architect

Live View

hdr__key_hash	hdr__store	hdr__operation	CustomerID	ContactTitle
e +t$¥¼Éëëe o©âé f	CURRENT	L	ALFKI	Sales Representative
zcÖy$ YwóWæ$ Q. ß 0	CURRENT	L	ANATR	Owner
)î8Wa¥Ã?ùÒ Ù³ '¾SG_	CURRENT	U	ANTON	Qlik Architect

Figure 13 – Qlik Cloud Data Integration Storage Task Live View after Applied Changes

The snapshot in Figure 13 shows the current view in the storage zone after the physical merge is completed. Note the change of ContactTitle to ‘Qlik Architect’ for the CustomerID ‘ANTON’. As the changes in the landing zone change table continue to grow, performance of Live Views will eventually slow if the merge interval is too long, which is why periodic merging changes into the current table is important to keeping Live Views highly performant.

Conclusion

Qlik Cloud Data Integration with the data movement gateway provides a solution to onboard enterprise on-premises data to a cloud platform by using SaaS capabilities to build a robust data pipeline. By capturing the source changes in a write-optimized format and using a store task to merge applied changes in a read-optimized format. Data can be captured in near real-time and applied with lower cloud compute costs.

LEAD WITH DATA™

About Qlik

Qlik's vision is a data-literate world, one where everyone can use data to improve decision-making and solve their most challenging problems. Only Qlik offers end-to-end, real-time data integration and analytics solutions that help organizations access and transform all their data into value. Qlik helps companies lead with data to see more deeply into customer behavior, reinvent business processes, discover new revenue streams, and balance risk and reward. Qlik does business in more than 100 countries and serves over 50,000 customers around the world.

qlik.com