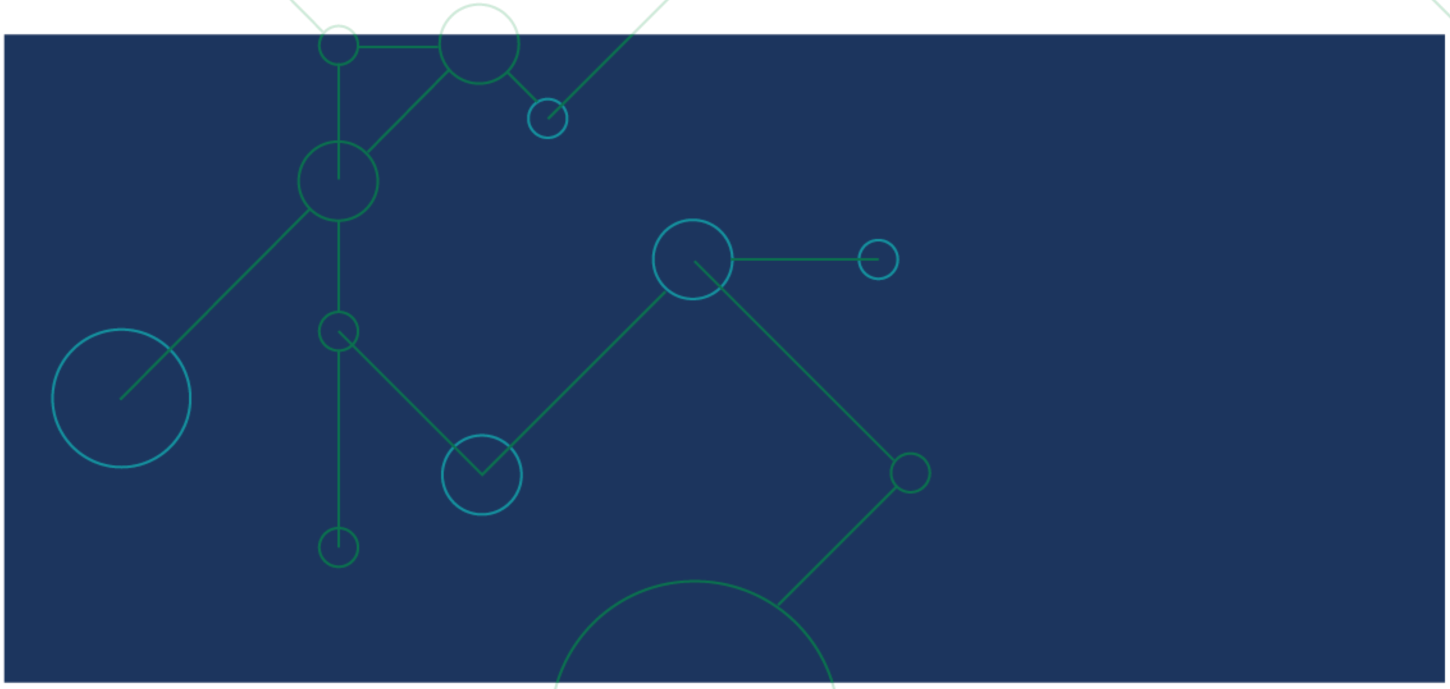
WHITE PAPER

Low-Latency Data Integration for Data Lakes

Defining Low-Latency Processing with Qlik Replicate, Qlik Compose for Data Lakes and Databricks Delta



QLIK.COM

LEAD WITH DATA™ Qlik

TABLE OF CONTENTS

SUMMARY

- Low-latency data movement to data lake storage has become a requirement for organizations.
- Qlik Replicate can capture and load incremental change data directly into Databricks Delta in near real-time.
- Qlik Compose for Data Lakes can be configured to support low-latency views utilizing a Databricks cluster.

INTRODUCTION

As data lakes mature, the need for low-latency data is growing. Organizations need analytics-ready data in near real-time for consumption. Capturing data changes in real-time can be complex with modern cloud data lake platforms. Utilizing Qlik's Data Integration platform with Databricks can help organizations achieve low-latency data requirements with their data lake.

With Qlik Replicate data can be captured from various sources and replicated directly to Databricks Delta tables. Qlik Replicate completes the full data load from the source and transitions into change data capture (CDC) mode. This allows source data transactions committed to the transaction logs to be replicated to Databricks Delta tables in near real-time.

With the soon to be launched, Qlik Compose for Data Lakes 2nd generation solution, Databricks users will be able to create a Historical Data and Operational Data Store with low-latency.

The following will describe a methodology to use Qlik Replicate and Qlik Compose for Data Lakes with Databricks Delta to provide a low-latency data solution.

Consumers of this document should have a basic understanding of Qlik Replicate and Qlik Compose for Data Lakes.

Use Case Description

Databricks Delta provides a solution for managing data using Spark within your data lake. The Qlik Data Integration Platform provides an effective low latency solution to ingest data into Databricks Delta. The integration between both platforms can provide an effective way to build and maintain a data pipeline for your modern data lake.

The whitepaper will demonstrate the use case of moving a Northwind sales dataset from a data source to a Databricks Delta target utilizing Qlik Replicate and Qlik Compose for Data Lake. The use case will show two methods to provide data in real-time to Databricks Delta. The first method utilizing only Qlik Replicate will show how a replication task will utilize the source data source logs to move change data to Databrick Delta tables in near real-time. Qlik Replicate will be used to do the initial full load of data and manage the change data for Databricks Delta tables. The second method will use the Qlik Compose for Data Lakes 2nd generation solution to create near real-time views for Historical and Operational Data Store within Databricks Delta. Qlik Compose for Data Lakes will also merge the full and change capture data into Databricks Delta tables after the execution of the Storage Task.

Both methods will show Qlik Data Integration Platform’s ability to provide a low latency alternative for Databricks Delta.

Introduction

Qlik Replicate Low Latency Data with Databricks Delta

What is Databricks Delta? Delta Lake is a storage layer that brings reliability to data lakes that utilize Spark executed on a Databricks server. Delta Lake provides ACID transactions, scalable metadata handing and unifies streaming with batch data processing.

Method #1

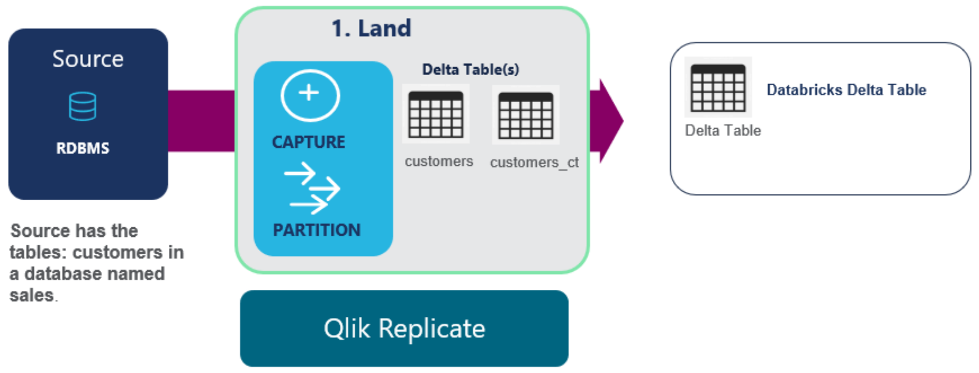
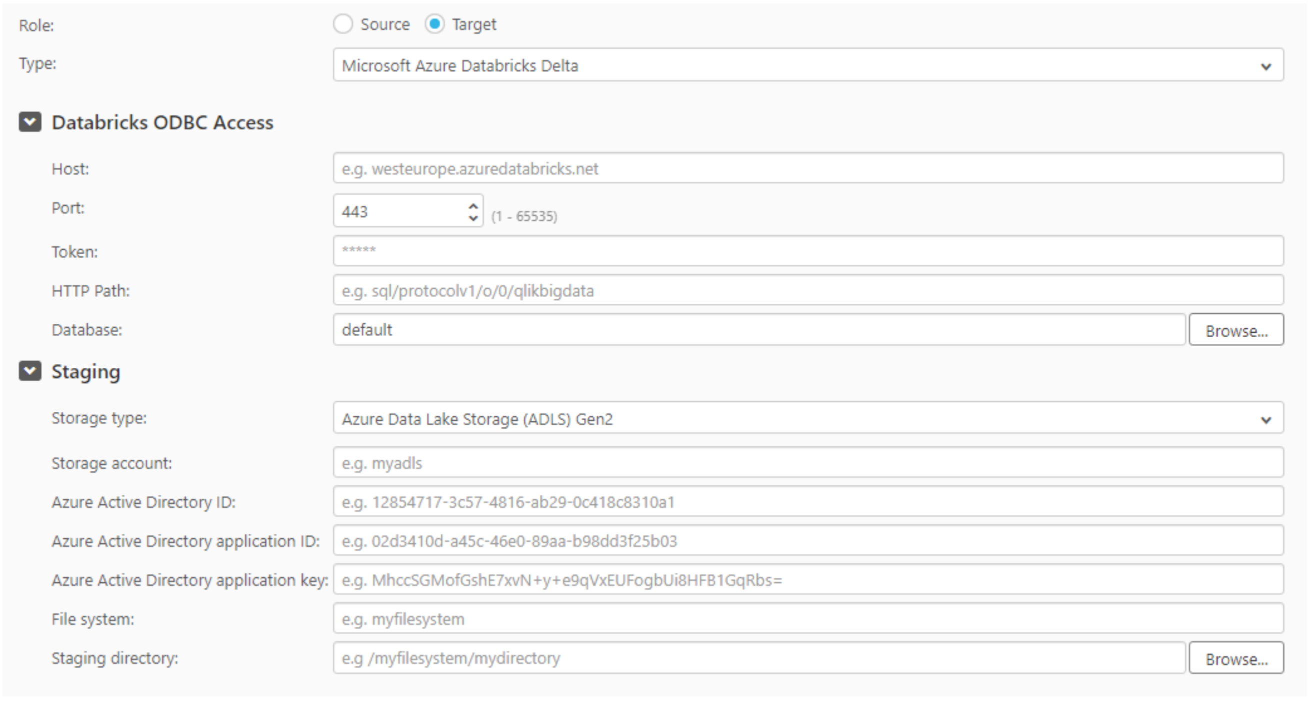


Figure 1- Qlik Replicate Ingestion to Databricks Delta Table

Utilizing Qlik Replicate, data from the source system can be replicated directly to Delta tables in Databricks. Qlik Replicate automatically creates the target change Delta table as part of the data movement automation and then performs the initial load. Once the full data has been loaded to the Delta Table, Qlik Replicate will switch to CDC mode and apply changes to the Delta table in near real-time. The changes can be stored in Databricks Delta audit tables to track all changes that are applied.

To create a replication task to load source data into a Databricks Delta Table, the Databricks Delta Endpoint must be used. An ODBC connection to the cluster is used with a staging area for data.



Role:	Source [✓ Target]
Type:	Microsoft Azure Databricks Delta

Databricks ODBC Access

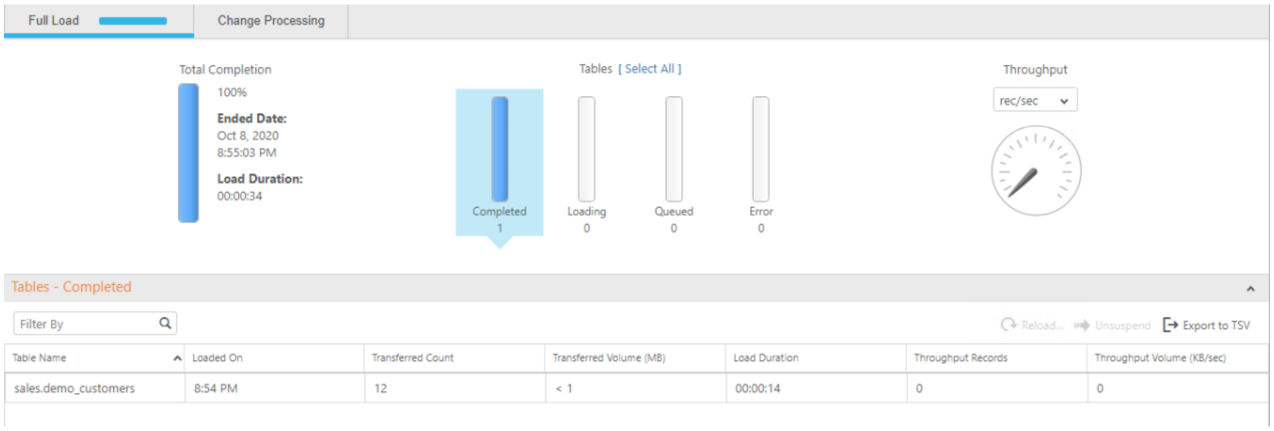
Host:	e.g. westeurope.azuredatabricks.net
Port:	443 (1 - 65535)
Token:	*****
HTTP Path:	e.g. sql/protocolv1/o/0/qlikbigdata
Database:	default Browse...

Staging

Storage type:	Azure Data Lake Storage (ADLS) Gen2
Storage account:	e.g. myadls
Azure Active Directory ID:	e.g. 12854717-3c57-4816-ab29-0c418c8310a1
Azure Active Directory application ID:	e.g. 02d3410d-a45c-46e0-89aa-b98dd3f25b03
Azure Active Directory application key:	e.g. MhccSGMofGshE7xvN+y+e9qVxEUFogbUi8HFB1GqRbs=
File system:	e.g. myfilesystem
Staging directory:	e.g /myfilesystem/mydirectory Browse...

Figure 3- Qlik Replicate Databricks Delta Example Azure Endpoint

Once the endpoint has tested successfully. A Qlik Replicate task as shown in the Full Load tab below, automatically creates the Delta tables in the database selected in the Replicate target endpoint. The data from the source database is converted into files and stored in the staging directory given in the Replicate target endpoint. The data will be accessible by querying the Delta table in the database.



Total Completion
100%
Ended Date: Oct 8, 2020 8:55:03 PM
Load Duration: 00:00:34

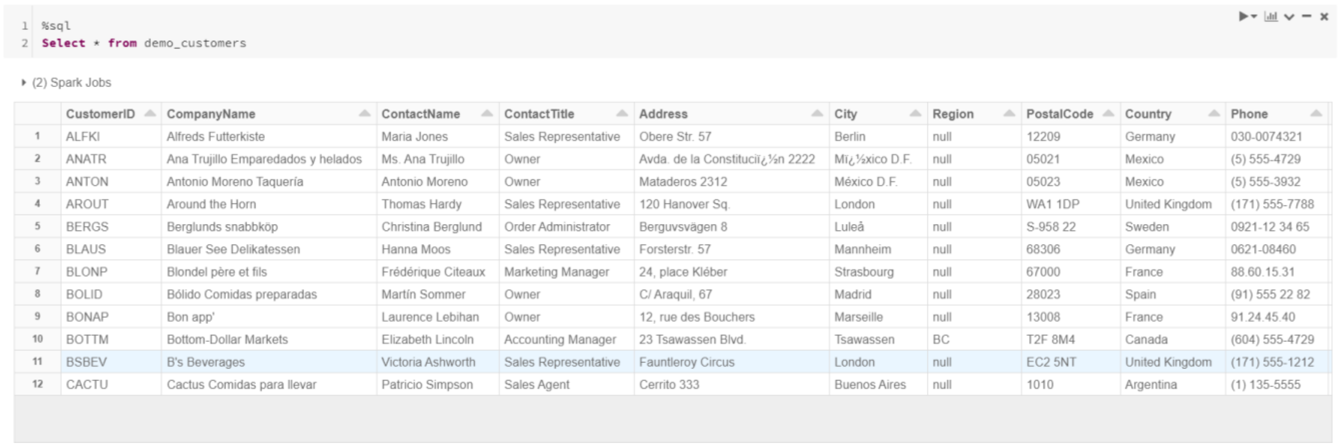
Completed	Loading	Queued	Error
1	0	0	0

Tables - Completed

Filter By   Reload... Unsuspend Export to TSV

Table Name	Loaded On	Transferred Count	Transferred Volume (MB)	Load Duration	Throughput Records	Throughput Volume (KB/sec)
sales.demo_customers	8:54 PM	12	< 1	00:00:14	0	0

Figure 4- Qlik Replicate Task Full Load Dashboard



```
%sql
Select * from demo_customers
```

(2) Spark Jobs

	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone
1	ALFKI	Alfreds Futterkiste	Maria Jones	Sales Representative	Obere Str. 57	Berlin	null	12209	Germany	030-0074321
2	ANATR	Ana Trujillo Emparedados y helados	Ms. Ana Trujillo	Owner	Avda. de la Constituciï¿½n 2222	Mï¿½xico D.F.	null	05021	Mexico	(5) 555-4729
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	null	05023	Mexico	(5) 555-3932
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	null	WA1 1DP	United Kingdom	(171) 555-7788
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	null	S-958 22	Sweden	0921-12 34 65
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	null	68306	Germany	0621-08460
7	BLONP	Blondel père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	null	67000	France	88.60.15.31
8	BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid	null	28023	Spain	(91) 555 22 82
9	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	null	13008	France	91.24.45.40
10	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen	BC	T2F 8M4	Canada	(604) 555-4729
11	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London	null	EC2 5NT	United Kingdom	(171) 555-1212
12	CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333	Buenos Aires	null	1010	Argentina	(1) 135-5555

Figure 5- Query of table from Databricks Notebook

Once the Qlik Replicate task is in change-data-capture mode, ten example insert statements executed at the source are captured and loaded into the existing table in near-real time. Also, ten update statements were executed on the source database. The updated data changes were applied to the target and optionally stored in the change table. (The DDL statements are available in the Appendix.)

Example Insert Logs

```
00002468: 2020-10-08T21:20:13 [SOURCE_CAPTURE  ]I:  > ROTATE_EVENT  (mysql_endpoint_capture.c:3157)

00004168: 2020-10-08T21:20:14 [TARGET_APPLY    ]I:  Net Changes table name for the task is
'attrep_changesEEAA8BC5F7329EB3'  (bulk_apply.c:3670)
```

Example Update Logs

```
00006436: 2020-10-08T21:54:32 [SORTER          ]I:  Task is running  (sorter.c:702)

00006436: 2020-10-08T21:59:56 [SORTER          ]I:  In the Before/After update events sequence the first event is
'BEFORE_UPDATE (16)'  (sorter_transaction.c:1288)
```

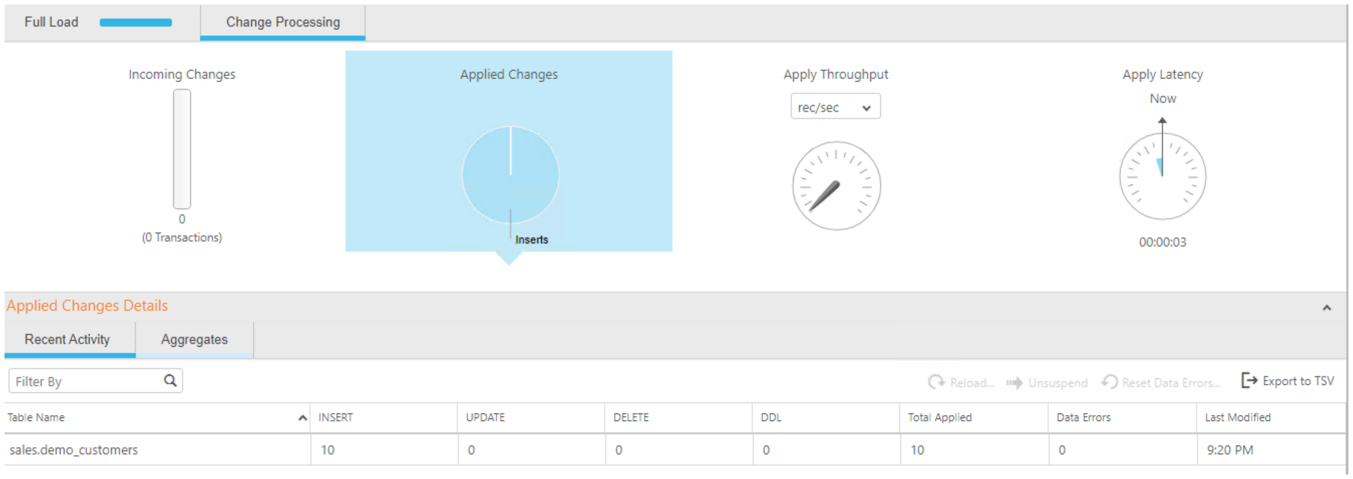
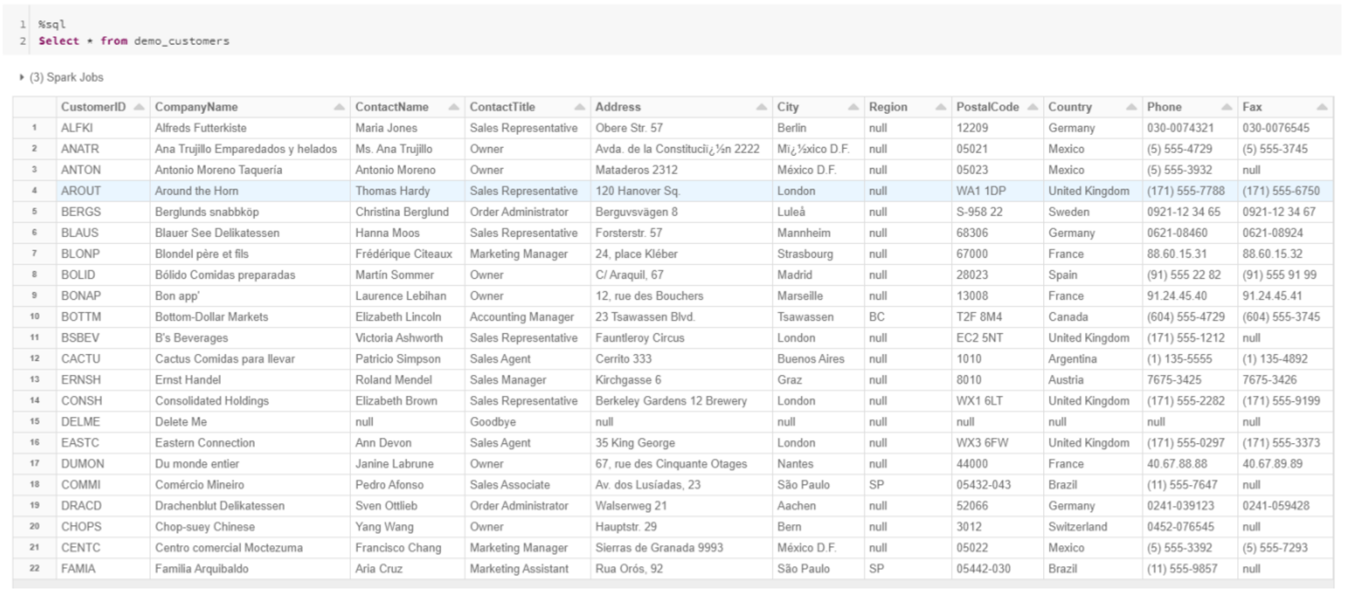


Table Name	INSERT	UPDATE	DELETE	DDL	Total Applied	Data Errors	Last Modified
sales.demo_customers	10	0	0	0	10	0	9:20 PM

Figure 6- Replicate Task Change Processing Dashboard



```
%sql
Select * from demo_customers
```

	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax
1	ALFKI	Alfreds Futterkiste	Maria Jones	Sales Representative	Obere Str. 57	Berlin	null	12209	Germany	030-0074321	030-0076545
2	ANATR	Ana Trujillo Emparedados y helados	Ms. Ana Trujillo	Owner	Avda. de la Constitucii¿½n 2222	Mï¿½xico D.F.	null	05021	Mexico	(5) 555-4729	(5) 555-3745
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	null	05023	Mexico	(5) 555-3932	null
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	null	WA1 1DP	United Kingdom	(171) 555-7788	(171) 555-6750
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	null	S-958 22	Sweden	0921-12 34 65	0921-12 34 67
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	null	68306	Germany	0621-08460	0621-08924
7	BLONP	Blondel père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	null	67000	France	88.60.15.31	88.60.15.32
8	BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid	null	28023	Spain	(91) 555 22 82	(91) 555 91 99
9	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	null	13008	France	91.24.45.40	91.24.45.41
10	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen	BC	T2F 8M4	Canada	(604) 555-4729	(604) 555-3745
11	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London	null	EC2 5NT	United Kingdom	(171) 555-1212	null
12	CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333	Buenos Aires	null	1010	Argentina	(1) 135-5555	(1) 135-4892
13	ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchgasse 6	Graz	null	8010	Austria	7675-3425	7675-3426
14	CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Gardens 12 Brewery	London	null	WX1 6LT	United Kingdom	(171) 555-2282	(171) 555-9199
15	DELME	Delete Me	null	Goodbye	null	null	null	null	null	null	null
16	EASTC	Eastern Connection	Ann Devon	Sales Agent	35 King George	London	null	WX3 6FW	United Kingdom	(171) 555-0297	(171) 555-3373
17	DUMON	Du monde entier	Janine Labrune	Owner	67, rue des Cinquante Otages	Nantes	null	44000	France	40.67.88.88	40.67.89.89
18	COMMI	Comércio Mineiro	Pedro Afonso	Sales Associate	Av. dos Lusíadas, 23	São Paulo	SP	05432-043	Brazil	(11) 555-7647	null
19	DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Administrator	Walserweg 21	Aachen	null	52066	Germany	0241-039123	0241-059428
20	CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29	Bern	null	3012	Switzerland	0452-076545	null
21	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.	null	05022	Mexico	(5) 555-3392	(5) 555-7293
22	FAMIA	Familia Arquibaldo	Aria Cruz	Marketing Assistant	Rua Orós, 92	São Paulo	SP	05442-030	Brazil	(11) 555-9857	null

Figure 7- Databricks query with new inserts

With Qlik Replicate, users can achieve low-latency change data capture to Databricks Delta from the data source. This will provide users with a low-latency replication solution for Databricks Delta. Do note that with Qlik Replicate, you can only generate an Operational Data Store. If the organizational need is to generate Operational and Historical Data Stores, Method 2 with Qlik Compose for Data Lake 2nd generation solution is required.

Qlik Compose for Data Lake 2nd Generation Solution with Databricks Delta

Utilizing Qlik Compose for Data Lakes with Qlik Replicate, low-latency data can be loaded from source to update **current and historical** data for Databricks Delta views and tables. Compose for Data Lakes 2nd generation solution provides a solution to support low latency. Qlik Compose for Data Lakes can benefit users that need to solve historical or operational data use cases in near-real time.

Method #2

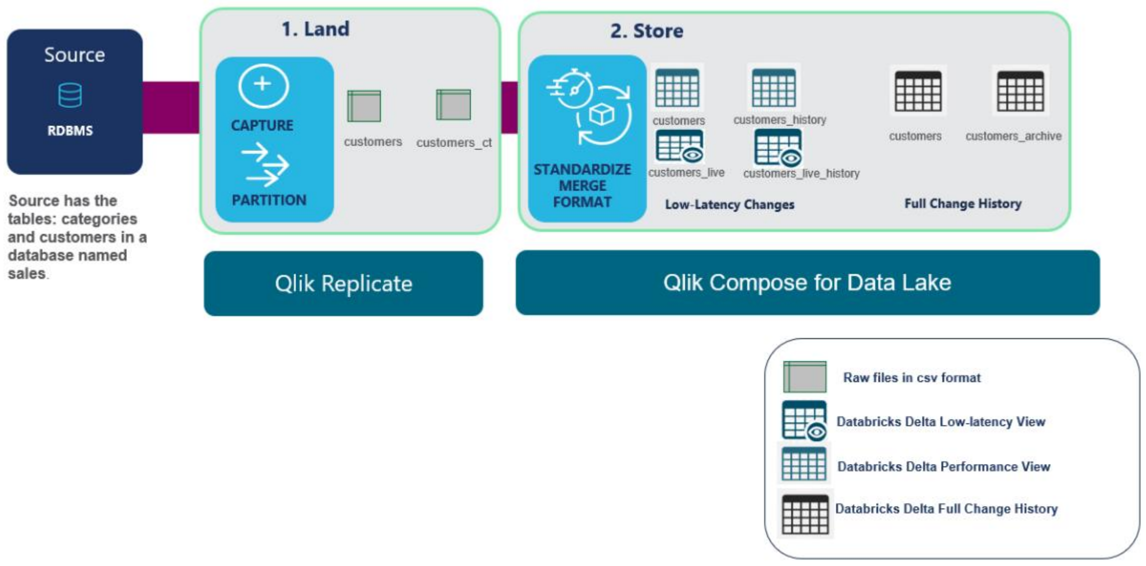


Figure 8- Qlik Compose 2nd Generation Solution

Qlik Compose for Data Lakes process data for Delta Lake consumers through scheduled micro-batch execution of the Compose workflow. The Qlik Compose for Data Lakes 1st generation solution provided processed Delta table representations of the data which would not meet the requirements for low-latency data in Databricks. The Compose for Data Lakes 2nd generation solution now generates "live views" that *merge on read* the latest unprocessed changes in the change table, including the last open partition.

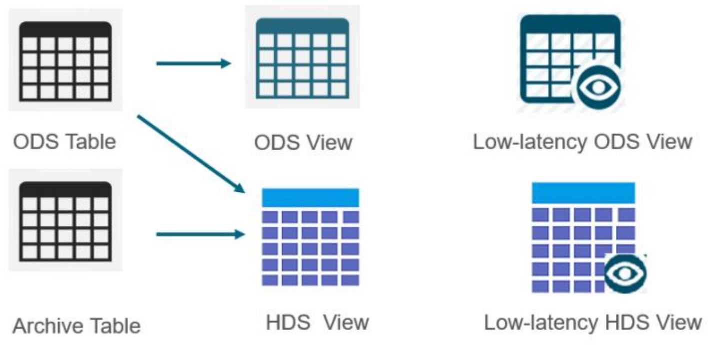


Figure 9- Qlik Compose 2nd Generation Solution

The ODS and HDS view are loaded from Compose execution of Data Storage Tasks. The low-latency ODS and HDS views read data delivered to Databricks from Replicate. Thus, the data is available with very low-latency for consumption.

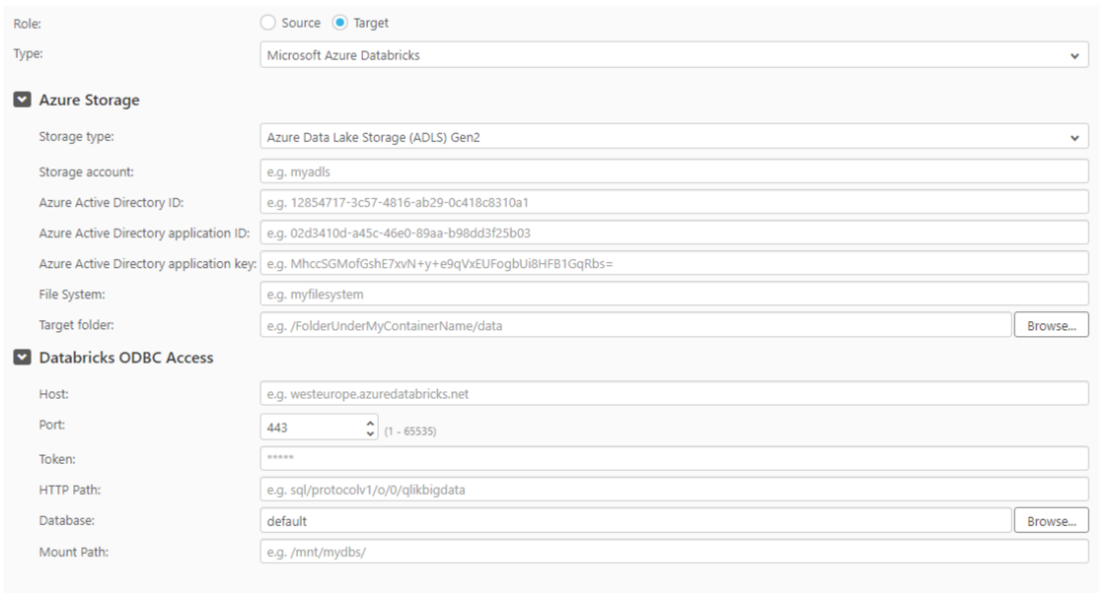
A significant change to the data ingestion architecture with gen2 is a new Replicate feature known as Speed partitions (see callout). Qlik Replicate will create the partition metadata at the start of the partition and not when the partition is closed. Also, Qlik Replicate will automatically clean up processed partitions and automatically handle reloading tables after a source table reload event.

What is a Speed Partition?

Speed partitioning is a new feature in Qlik Replicate, designed to improve the read latency of data delivered to Databricks. Replicate delivers data to Databricks using time-based partition windows. Previously Replicate created the partitions when the time-based partition window was completed. This meant that data in the change tracking layer of the landing zone could not be read until the partition was created. The new architecture creates the partition at that start of the time window instead of at the end. This allows Qlik Data Integration to manage ingestion partitions, with the added benefit of being able to read data as soon as it has been delivered instead of having to wait for the time-based partition window to close

Qlik Replication Configuration for Databricks Delta

When creating a Qlik Replicate task, users will select the Databricks endpoint. The Databricks endpoint specifies a cloud storage location for the ingested files. With the Databricks ODBC access for the Databricks landing database and a mount path for the external data files to the cloud storage location of the ingested files. (The endpoint is the same Databricks endpoint in the 1st generation solution and utilizes S3 or ADLS gen2 for cloud storage.)


Role:	○ Source ● Target
Type:	Microsoft Azure Databricks

Azure Storage

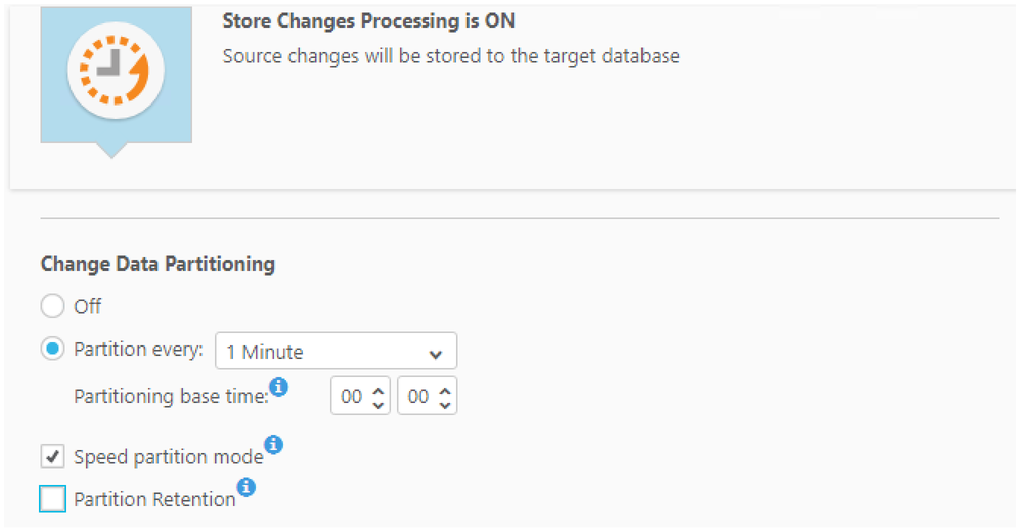
Storage type:	Azure Data Lake Storage (ADLS) Gen2
Storage account:	e.g. myadls
Azure Active Directory ID:	e.g. 12854717-3c57-4816-ab29-0c418c8310a1
Azure Active Directory application ID:	e.g. 02d3410d-a45c-46e0-89aa-b98dd3f25b03
Azure Active Directory application key:	e.g. MhccSGMofGshE7xvN+y+e9qVxEUFogbUi8HFB1GqRbs=
File System:	e.g. myfilesystem
Target folder:	e.g. /FolderUnderMyContainerName/data

Databricks ODBC Access

Host:	e.g. westeurope.azuredatabricks.net
Port:	443 (1 - 65535)
Token:	*****
HTTP Path:	e.g. sql/protocolv1/o/0/qlikbigdata
Database:	default
Mount Path:	e.g. /mnt/mydbs/

Figure 10- Qlik Replicate Databricks Example Azure Endpoint

The user will need to make a decision for the Replicate task configuration to use the change data partitioning interval (which impacts how often Compose can process) and the file change processing which dictates how often the files are delivered and thus available for the live views. When creating the Qlik Replicate task with the store changes option enabled for the same source dataset, the change data partitioning task settings must be turned on with the speed partition mode enabled. (Partitions interval should be set to a time that makes sense for you to apply changes to Databricks Delta.)


Store Changes Processing is ON

Source changes will be stored to the target database

Change Data Partitioning

○ Off

● Partition every: 1 Minute

Partitioning base time: 00 00

☑ Speed partition mode

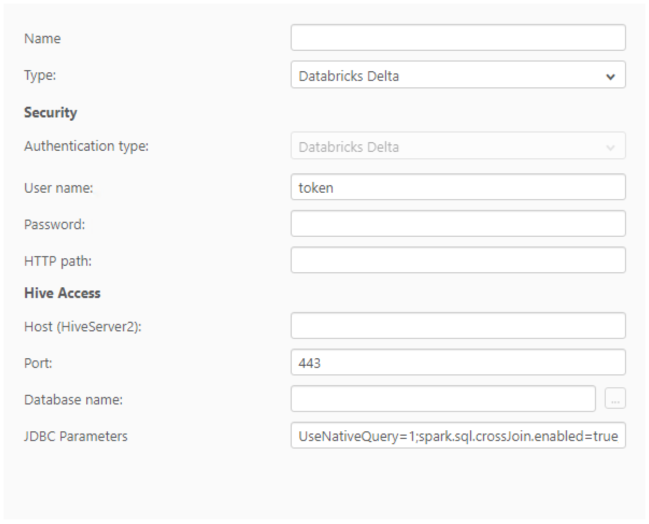
☐ Partition Retention

Figure 11 – Qlik Replicate Speed Partition mode enabled

After configuration of the Qlik Replication task, start the task for full data ingestion and the transition of the task to change data capture mode. (Make sure to mount the file system using Databricks *dbutils* commands to be able to query the data.)

Qlik Compose Configuration for Databricks Delta

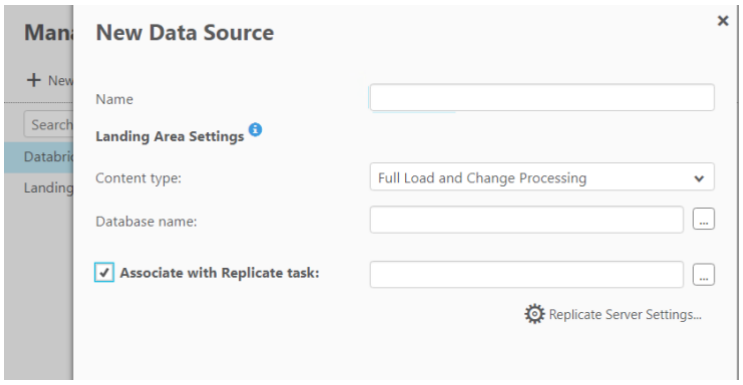
Once the Replication task is in CDC mode, configure a Databricks Delta project storage zone in Qlik Compose. Input the Databricks Cluster credentials and select the database that will be used to store the Delta tables and views.



Name	
Type:	Databricks Delta
Security	
Authentication type:	Databricks Delta
User name:	token
Password:	
HTTP path:	
Hive Access	
Host (HiveServer2):	
Port:	443
Database name:	
JDBC Parameters	UseNativeQuery=1;spark.sql.crossJoin.enabled=true

Figure 12 – Compose Configuration for Storage Zone

Configure the landing connection to use the Qlik Replicate task created to ingest data to Databricks landing zone. Once the connections are tested, discover the metadata from the landing database and make any modifications to the logical metadata.



New Data Source

Name	
Landing Area Settings	
Content type:	Full Load and Change Processing
Database name:	
✓ **Associate with Replicate task:**	

Figure 13 – Compose Landing Connection

Validate the model and select create from the storage zone drop-down. The ETL sets for the full and change data capture data mappings will be created. In addition to creating the ETL sets the Databricks Delta tables will be created for the current table data and archive table data within the database provided for the Storage Zone. The archive table will contain the historical data set. Thus, the Historical Data Store (HDS) and Operation Data Store (ODS) are stored together in one single project. The tables are loaded via the Qlik Compose for Data Lakes storage task execution.

Schema Name	Object Name	Purpose
storagezone_database	customers	Operational Data Store dataset processed by Compose
storagezone_database	customers_archive	Historical Data Store dataset processed by Compose

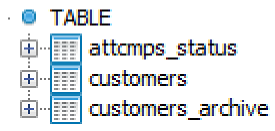
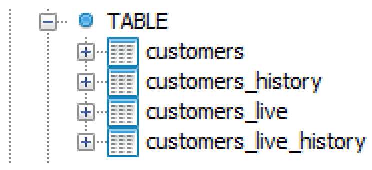


TABLE
attcmps_status
customers
customers_archive

Figure 14 – Delta tables Created for Storage Zone default schema

Two additional schemas are created for the storage zone low-latency views. The first schema created appends a **'_v'** to the Storage Zone default schema name. Example **storagezone_database_v**. In the low-latency schema-tables are created for the current and historical data for each table in the storage zone ETL set. The second schema created appends **'v_internal'** to the storage zone database name. (Example **storagezone_database_v_internal**.) The tables in the schema are for the applied archive and live changes data set. The tables are used for internal Qlik Compose for Data Lakes processing for the low-latency current and historical data view.

Schema Name	Object Name	Purpose
storagezone_database_v	customers	Shows loaded data and changes combined data set from landing zone tables.
storagezone_database_v	customers_history	Static Type 2 history that has been processed by Compose
storagezone_database_v	customers_live	Type 1 view that shows r eal-time ODS dataset
storagezone_database_v	customers_live_history	Type 2 history view that shows real-time history
storagezone_database_v_internal	customers_live_changes	Shows changes applied to ODS table in the storagezone_database schema
storagezone_database_v_internal	customers_applied_archive	Shows changes applied to archive table in the storagezone_database schema



```
TABLE
  customers
  customers_history
  customers_live
  customers_live_history
```

Figure 15- Delta tables created for Storage Zone Low-Latency Views

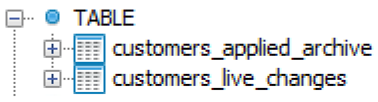
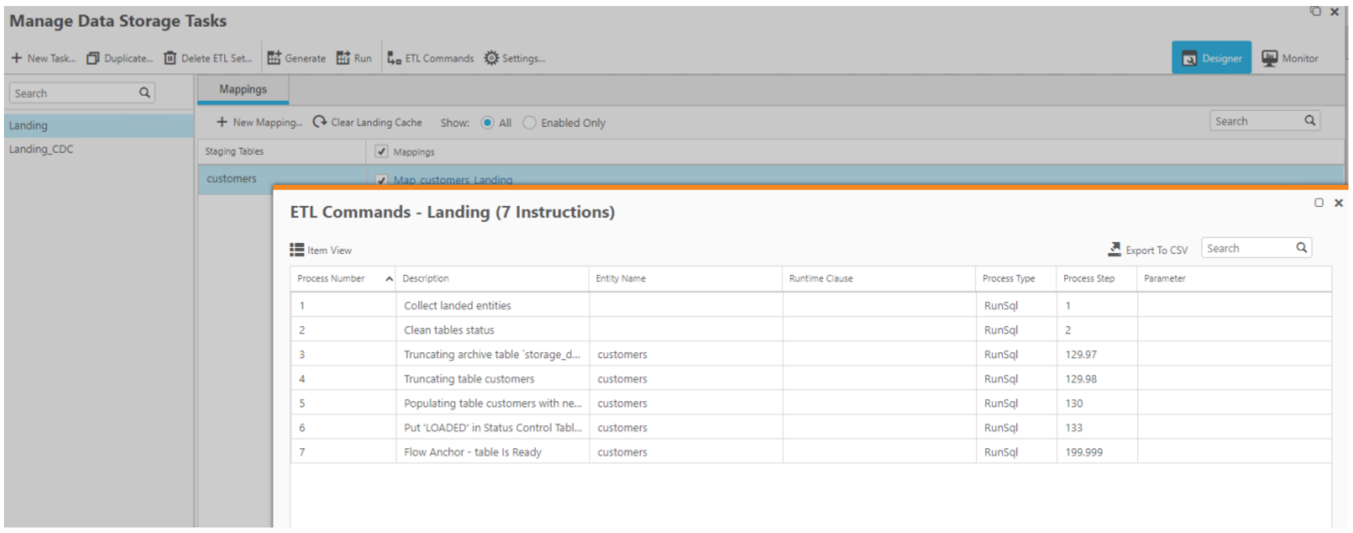


TABLE
customers_applied_archive
customers_live_changes

Figure 16- Delta tables created for Storage Zone Internal Latency Views

After Qlik Compose for Data Lakes creates the Databricks Delta schema and tables, the full load and change data capture ETL sets in the storage zone will contain mappings for the table. Within the mapping users can apply lookups and expressions for staging column attributes in the mapping. After adding any required lookups or expressions to the mappings, the ETL commands can be generated from clicking on the Generate icon in the manage data storage tasks GUI.


Manage Data Storage Tasks
New Task... Duplicate... Delete ETL Set... Generate Run ETL Commands Settings...
Designer Monitor
Search
Landing
Landing_CDC
Mappings
New Mapping... Clear Landing Cache Show: All Enabled Only
Search
Staging Tables
Mappings
customers
Map customers Landing
Item View
Export To CSV
Search

ETL Commands - Landing (7 Instructions)

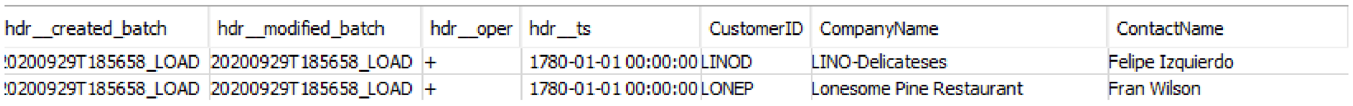
Process Number	Description	Entity Name	Runtime Clause	Process Type	Process Step	Parameter
1	Collect landed entities			RunSql	1	
2	Clean tables status			RunSql	2	
3	Truncating archive table `storage_d...	customers		RunSql	129.97	
4	Truncating table customers	customers		RunSql	129.98	
5	Populating table customers with ne...	customers		RunSql	130	
6	Put 'LOADED' in Status Control Tabl...	customers		RunSql	133	
7	Flow Anchor - table Is Ready	customers		RunSql	199.999	

Figure 17- ETL Commands Generated in Storage Task for Full Data Load

Once the ETL commands are generated they can be executed immediately. The commands will be executed on the Databricks cluster. After the initial execution of the Full ETL set, the customers table is loaded with the full data set from the landing customer table. The customer, customer_history, customer_live and customer_history_live table in the low-latency view schema are loaded with the landing customer table full data set.

Explanation of Data Sets after Initial Full Data Load

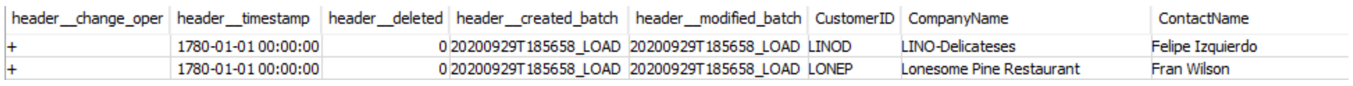
The customers table in the default storage zone schema was loaded with the initial full load of data from the customers table in the initial landing schema.



hdr__created_batch	hdr__modified_batch	hdr__oper	hdr__ts	CustomerID	CompanyName	ContactName
0200929T185658_LOAD	20200929T185658_LOAD	+	1780-01-01 00:00:00	LINOD	LINO-Delicateses	Felipe Izquierdo
0200929T185658_LOAD	20200929T185658_LOAD	+	1780-01-01 00:00:00	LONEP	Lonesome Pine Restaurant	Fran Wilson

Figure 18- Query results of customers table after Full Data Load

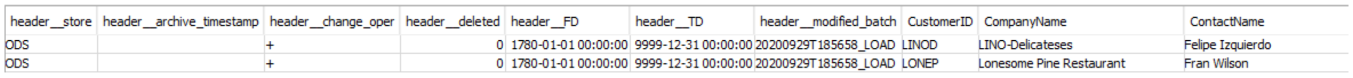
The customers table in the low-latency schema is loaded with the initial full load of data from the landing schema customer table.



header__change_oper	header__timestamp	header__deleted	header__created_batch	header__modified_batch	CustomerID	CompanyName	ContactName
+	1780-01-01 00:00:00	0	20200929T185658_LOAD	20200929T185658_LOAD	LINOD	LINO-Delicateses	Felipe Izquierdo
+	1780-01-01 00:00:00	0	20200929T185658_LOAD	20200929T185658_LOAD	LONEP	Lonesome Pine Restaurant	Fran Wilson

Figure 19- Query results of customers table in low-latency schema after Full Data Load

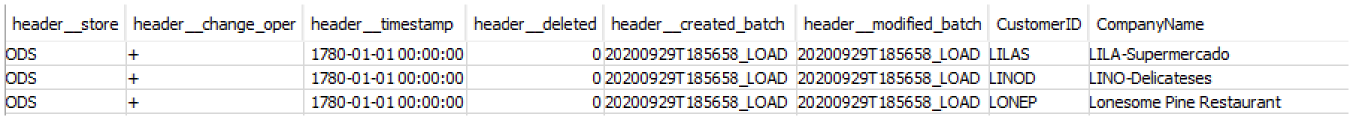
The customers_history table in the low-latency schema is loaded with the initial full load of data from the landing schema customer table.



header__store	header__archive_timestamp	header__change_oper	header__deleted	header__FD	header__TD	header__modified_batch	CustomerID	CompanyName	ContactName
ODS		+	0	1780-01-01 00:00:00	9999-12-31 00:00:00	20200929T185658_LOAD	LINOD	LINO-Delicateses	Felipe Izquierdo
ODS		+	0	1780-01-01 00:00:00	9999-12-31 00:00:00	20200929T185658_LOAD	LONEP	Lonesome Pine Restaurant	Fran Wilson

Figure 20- Query results of customers_history table in low-latency schema after Full Data Load

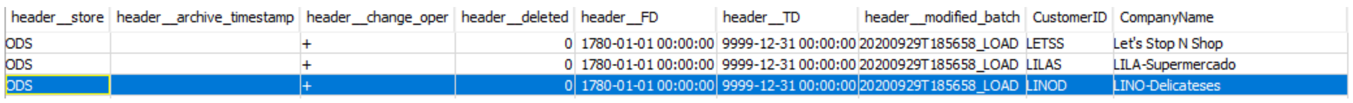
The customers_live table in the low-latency schema is loaded with the initial full load of data from the landing schema customer table.



header__store	header__change_oper	header__timestamp	header__deleted	header__created_batch	header__modified_batch	CustomerID	CompanyName
ODS	+	1780-01-01 00:00:00	0	20200929T185658_LOAD	20200929T185658_LOAD	LILAS	LILA-Supermercado
ODS	+	1780-01-01 00:00:00	0	20200929T185658_LOAD	20200929T185658_LOAD	LINOD	LINO-Delicateses
ODS	+	1780-01-01 00:00:00	0	20200929T185658_LOAD	20200929T185658_LOAD	LONEP	Lonesome Pine Restaurant

Figure 21- Query results of customers_live table in low-latency schema after Full Data Load

The customers_live_history table in the low-latency schema is loaded with the initial full load of data from the landing schema customer table.



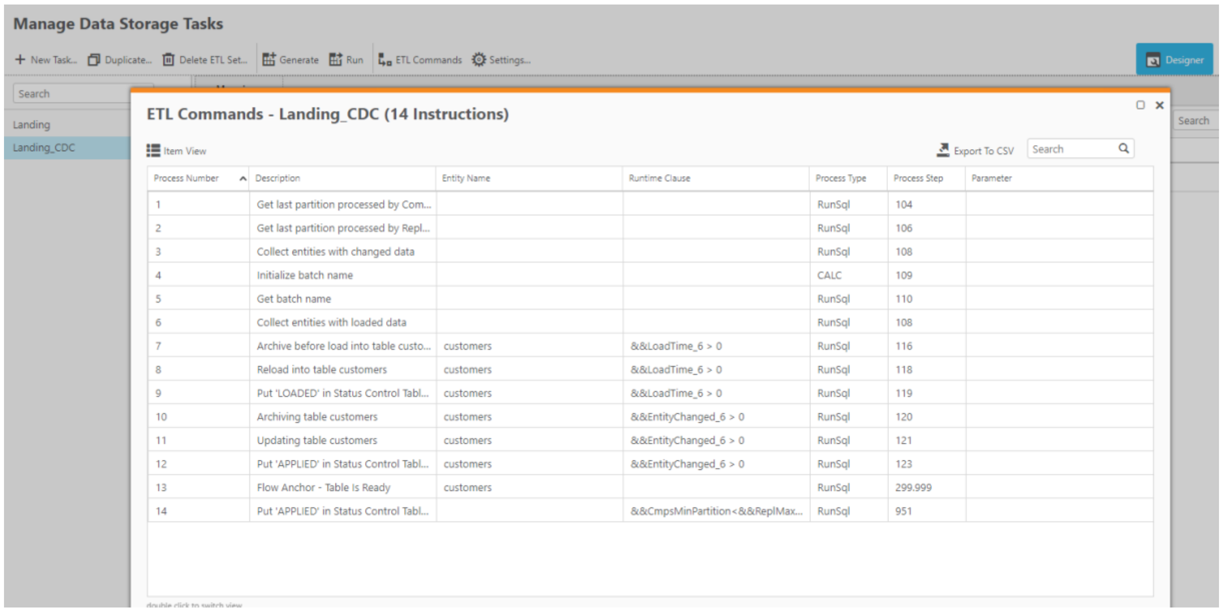
header__store	header__archive_timestamp	header__change_oper	header__deleted	header__FD	header__TD	header__modified_batch	CustomerID	CompanyName
ODS		+	0	1780-01-01 00:00:00	9999-12-31 00:00:00	20200929T185658_LOAD	LETSS	Let's Stop N Shop
ODS		+	0	1780-01-01 00:00:00	9999-12-31 00:00:00	20200929T185658_LOAD	LILAS	LILA-Supermercado
ODS		+	0	1780-01-01 00:00:00	9999-12-31 00:00:00	20200929T185658_LOAD	LINOD	LINO-Delicateses

Figure 22- Query results of customers_live_history table in low-latency schema after Full Data Load

After the initial full data load execution with Qlik Compose for Data Lakes, there is no data loaded into the internal low-latency schema tables: customers_applied_archive and customers_live_changes.

Qlik Compose Change Data Capture Storage Task

In the data storage task management GUI, you can generate the ETL Commands for the CDC ETL set. Before executing the CDC ETL set, which will load the data into the Qlik Compose for Data Lakes storage zone, execute changes on the source database to capture the data changes. (See Appendix for statements.)



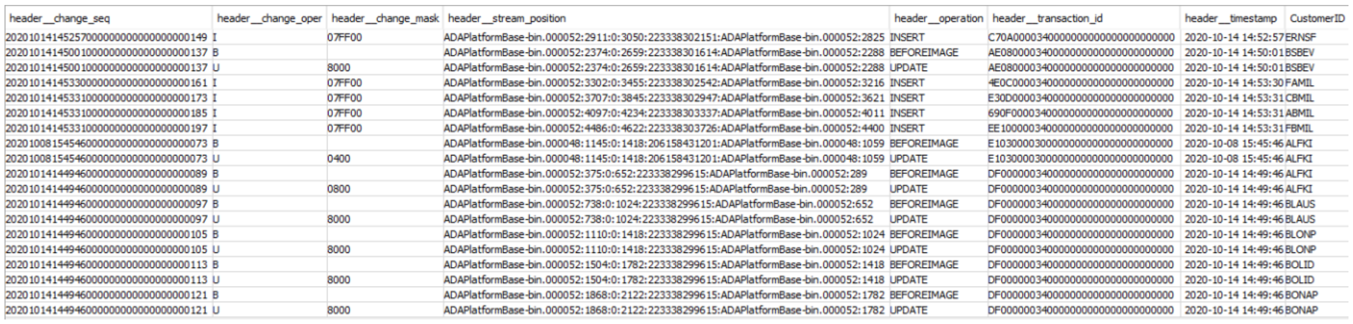
ETL Commands - Landing_CDC (14 Instructions)

Process Number	Description	Entity Name	Runtime Clause	Process Type	Process Step	Parameter
1	Get last partition processed by Com...			RunSql	104	
2	Get last partition processed by Repl...			RunSql	106	
3	Collect entities with changed data			RunSql	108	
4	Initialize batch name			CALC	109	
5	Get batch name			RunSql	110	
6	Collect entities with loaded data			RunSql	108	
7	Archive before load into table custo...	customers	&&LoadTime_6 > 0	RunSql	116	
8	Reload into table customers	customers	&&LoadTime_6 > 0	RunSql	118	
9	Put 'LOADED' in Status Control Tabl...	customers	&&LoadTime_6 > 0	RunSql	119	
10	Archiving table customers	customers	&&EntityChanged_6 > 0	RunSql	120	
11	Updating table customers	customers	&&EntityChanged_6 > 0	RunSql	121	
12	Put 'APPLIED' in Status Control Tabl...	customers	&&EntityChanged_6 > 0	RunSql	123	
13	Flow Anchor - Table Is Ready	customers		RunSql	299.999	
14	Put 'APPLIED' in Status Control Tabl...		&&CmpsMinPartition<&&ReplMax...	RunSql	951	

Figure 23- Qlik Compose ETL commands for CDC data load

Explanation of Data Sets before Compose Storage Task Change Data Capture execution

After execution of the source data changes, The data changes will be staged in the landing customer_ct table.

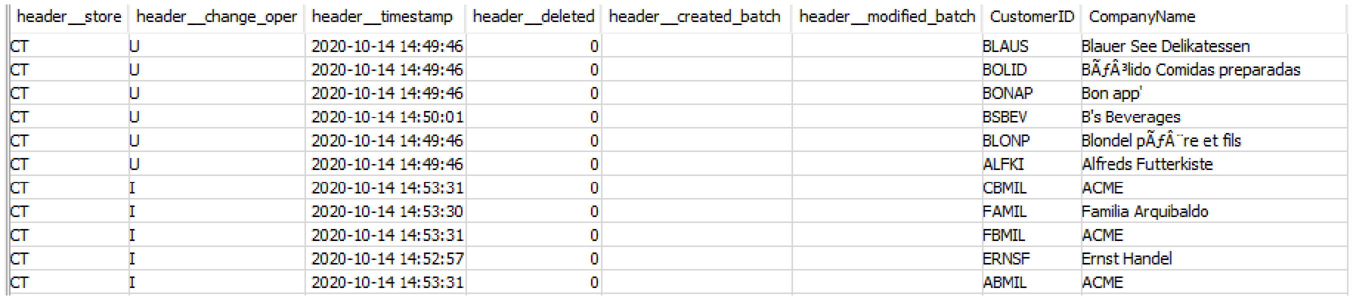


header__change_seq	header__change_oper	header__change_mask	header__stream_position	header__operation	header__transaction_id	header__timestamp	CustomerID
20201014145257000000000000000000149	I	07FF00	ADAPlatformBase-bin.000052:2911:0:3050:223338302151:ADAPlatformBase-bin.000052:2825	INSERT	C70A0000340000000000000000000000	2020-10-14 14:52:57	ERNSF
20201014145001000000000000000000137	B		ADAPlatformBase-bin.000052:2374:0:2659:223338301614:ADAPlatformBase-bin.000052:2288	BEFOREIMAGE	AE080000340000000000000000000000	2020-10-14 14:50:01	BSBEV
20201014145001000000000000000000137	U	8000	ADAPlatformBase-bin.000052:2374:0:2659:223338301614:ADAPlatformBase-bin.000052:2288	UPDATE	AE080000340000000000000000000000	2020-10-14 14:50:01	BSBEV
20201014145330000000000000000000161	I	07FF00	ADAPlatformBase-bin.000052:3302:0:3455:223338302542:ADAPlatformBase-bin.000052:3216	INSERT	4E0C0000340000000000000000000000	2020-10-14 14:53:30	FAMIL
20201014145331000000000000000000173	I	07FF00	ADAPlatformBase-bin.000052:3707:0:3845:223338302947:ADAPlatformBase-bin.000052:3621	INSERT	E30D0000340000000000000000000000	2020-10-14 14:53:31	CBMIL
20201014145331000000000000000000185	I	07FF00	ADAPlatformBase-bin.000052:4097:0:4234:223338303337:ADAPlatformBase-bin.000052:4011	INSERT	690F0000340000000000000000000000	2020-10-14 14:53:31	ABMIL
20201014145331000000000000000000197	I	07FF00	ADAPlatformBase-bin.000052:4486:0:4622:223338303726:ADAPlatformBase-bin.000052:4400	INSERT	EE100000340000000000000000000000	2020-10-14 14:53:31	FBMIL
20201008154546000000000000000000073	B		ADAPlatformBase-bin.000048:1145:0:1418:206158431201:ADAPlatformBase-bin.000048:1059	BEFOREIMAGE	E1030000300000000000000000000000	2020-10-08 15:45:46	ALFKI
20201008154546000000000000000000073	U	0400	ADAPlatformBase-bin.000048:1145:0:1418:206158431201:ADAPlatformBase-bin.000048:1059	UPDATE	E1030000300000000000000000000000	2020-10-08 15:45:46	ALFKI
20201014144946000000000000000000089	B		ADAPlatformBase-bin.000052:375:0:652:223338299615:ADAPlatformBase-bin.000052:289	BEFOREIMAGE	DF000003400000000000000000000000	2020-10-14 14:49:46	ALFKI
20201014144946000000000000000000089	U	0800	ADAPlatformBase-bin.000052:375:0:652:223338299615:ADAPlatformBase-bin.000052:289	UPDATE	DF000003400000000000000000000000	2020-10-14 14:49:46	ALFKI
20201014144946000000000000000000097	B		ADAPlatformBase-bin.000052:738:0:1024:223338299615:ADAPlatformBase-bin.000052:652	BEFOREIMAGE	DF000003400000000000000000000000	2020-10-14 14:49:46	BLAUS
20201014144946000000000000000000097	U	8000	ADAPlatformBase-bin.000052:738:0:1024:223338299615:ADAPlatformBase-bin.000052:652	UPDATE	DF000003400000000000000000000000	2020-10-14 14:49:46	BLAUS
20201014144946000000000000000000105	B		ADAPlatformBase-bin.000052:1110:0:1418:223338299615:ADAPlatformBase-bin.000052:1024	BEFOREIMAGE	DF000003400000000000000000000000	2020-10-14 14:49:46	BLONP
20201014144946000000000000000000105	U	8000	ADAPlatformBase-bin.000052:1110:0:1418:223338299615:ADAPlatformBase-bin.000052:1024	UPDATE	DF000003400000000000000000000000	2020-10-14 14:49:46	BLONP
20201014144946000000000000000000113	B		ADAPlatformBase-bin.000052:1504:0:1782:223338299615:ADAPlatformBase-bin.000052:1418	BEFOREIMAGE	DF000003400000000000000000000000	2020-10-14 14:49:46	BOLID
20201014144946000000000000000000113	U	8000	ADAPlatformBase-bin.000052:1504:0:1782:223338299615:ADAPlatformBase-bin.000052:1418	UPDATE	DF000003400000000000000000000000	2020-10-14 14:49:46	BOLID
20201014144946000000000000000000121	B		ADAPlatformBase-bin.000052:1868:0:2122:223338299615:ADAPlatformBase-bin.000052:1782	BEFOREIMAGE	DF000003400000000000000000000000	2020-10-14 14:49:46	BONAP
20201014144946000000000000000000121	U	8000	ADAPlatformBase-bin.000052:1868:0:2122:223338299615:ADAPlatformBase-bin.000052:1782	UPDATE	DF000003400000000000000000000000	2020-10-14 14:49:46	BONAP

Figure 24- Qlik Replicate change table with CDC changes

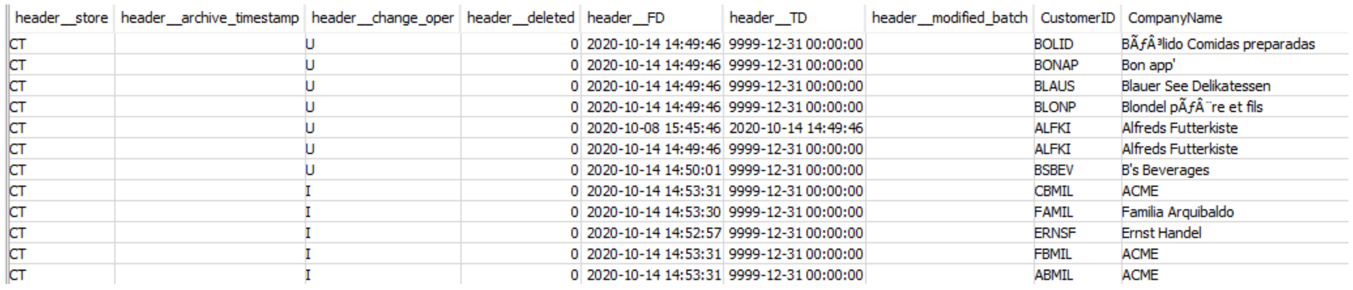
Since the low-latency tables are merge on read, the latest changes, including the last open partition from Qlik Replicate customer_ct table, will be available for consumption before the Qlik Compose CDC storage task is executed.

In the customers_live and customers_live_history table within the low-latency data storage schema the new changes are available for consumption.



header__store	header__change_oper	header__timestamp	header__deleted	header__created_batch	header__modified_batch	CustomerID	CompanyName
CT	U	2020-10-14 14:49:46	0			BLAUS	Blauer See Delikatessen
CT	U	2020-10-14 14:49:46	0			BOLID	BÃƒÂ³lido Comidas preparadas
CT	U	2020-10-14 14:49:46	0			BONAP	Bon app'
CT	U	2020-10-14 14:50:01	0			BSBEV	B's Beverages
CT	U	2020-10-14 14:49:46	0			BLONP	Blondel pÃƒÂ¨re et fils
CT	U	2020-10-14 14:49:46	0			ALFKI	Alfreds Futterkiste
CT	I	2020-10-14 14:53:31	0			CBMIL	ACME
CT	I	2020-10-14 14:53:30	0			FAMIL	Familia Arquibaldo
CT	I	2020-10-14 14:53:31	0			FBMIL	ACME
CT	I	2020-10-14 14:52:57	0			ERNSF	Ernst Handel
CT	I	2020-10-14 14:53:31	0			ABMIL	ACME

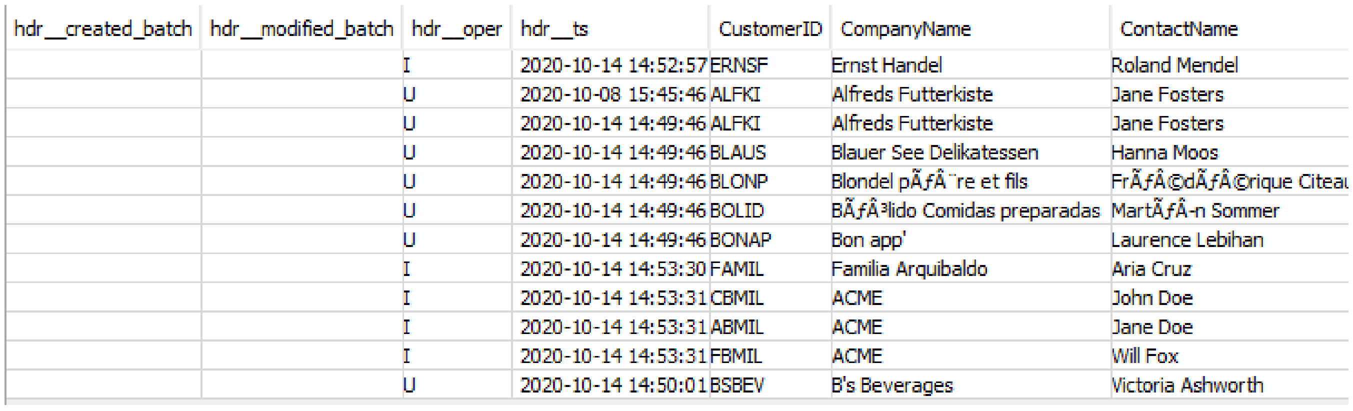
Figure 25- Changes available in the customers_live low-latency table



header__store	header__archive_timestamp	header__change_oper	header__deleted	header__FD	header__TD	header__modified_batch	CustomerID	CompanyName
CT		U	0	2020-10-14 14:49:46	9999-12-31 00:00:00		BOLID	BÃƒÂ³lido Comidas preparadas
CT		U	0	2020-10-14 14:49:46	9999-12-31 00:00:00		BONAP	Bon app'
CT		U	0	2020-10-14 14:49:46	9999-12-31 00:00:00		BLAUS	Blauer See Delikatessen
CT		U	0	2020-10-14 14:49:46	9999-12-31 00:00:00		BLONP	Blondel pÃƒÂ¨re et fils
CT		U	0	2020-10-08 15:45:46	2020-10-14 14:49:46		ALFKI	Alfreds Futterkiste
CT		U	0	2020-10-14 14:49:46	9999-12-31 00:00:00		ALFKI	Alfreds Futterkiste
CT		U	0	2020-10-14 14:50:01	9999-12-31 00:00:00		BSBEV	B's Beverages
CT		I	0	2020-10-14 14:53:31	9999-12-31 00:00:00		CBMIL	ACME
CT		I	0	2020-10-14 14:53:30	9999-12-31 00:00:00		FAMIL	Familia Arquibaldo
CT		I	0	2020-10-14 14:52:57	9999-12-31 00:00:00		ERNSF	Ernst Handel
CT		I	0	2020-10-14 14:53:31	9999-12-31 00:00:00		FBMIL	ACME
CT		I	0	2020-10-14 14:53:31	9999-12-31 00:00:00		ABMIL	ACME

Figure 26- Changes available in the customers_live_history low-latency table

The changes will also be available in the internal low-latency schema customers_live_changes table.

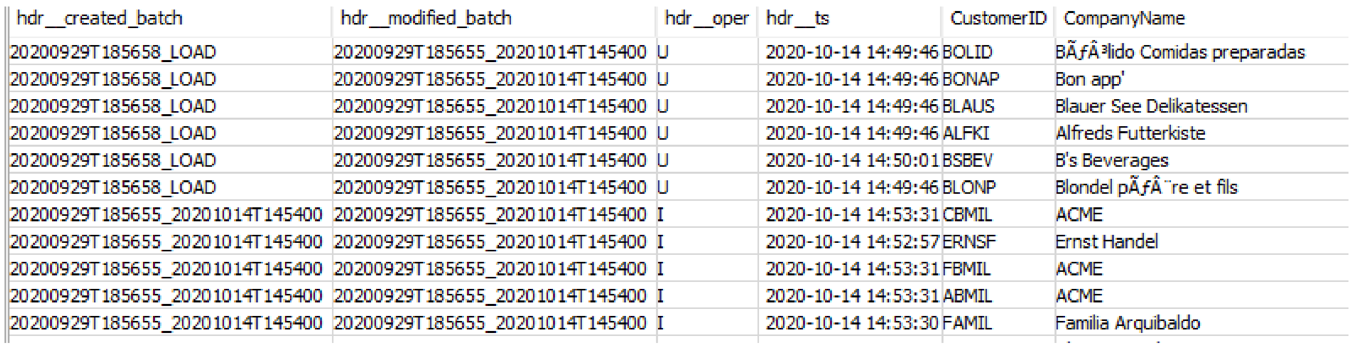


hdr__created_batch	hdr__modified_batch	hdr__oper	hdr__ts	CustomerID	CompanyName	ContactName
		I	2020-10-14 14:52:57	ERNSF	Ernst Handel	Roland Mendel
		U	2020-10-08 15:45:46	ALFKI	Alfreds Futterkiste	Jane Fosters
		U	2020-10-14 14:49:46	ALFKI	Alfreds Futterkiste	Jane Fosters
		U	2020-10-14 14:49:46	BLAUS	Blauer See Delikatessen	Hanna Moos
		U	2020-10-14 14:49:46	BLONP	Blondel pÃƒÂ¨re et fils	FrÃƒÂ©dÃƒÂ©rique Citeau
		U	2020-10-14 14:49:46	BOLID	BÃƒÂ³lido Comidas preparadas	MartÃƒÂ-n Sommer
		U	2020-10-14 14:49:46	BONAP	Bon app'	Laurence Lebihan
		I	2020-10-14 14:53:30	FAMIL	Familia Arquibaldo	Aria Cruz
		I	2020-10-14 14:53:31	CBMIL	ACME	John Doe
		I	2020-10-14 14:53:31	ABMIL	ACME	Jane Doe
		I	2020-10-14 14:53:31	FBMIL	ACME	Will Fox
		U	2020-10-14 14:50:01	BSBEV	B's Beverages	Victoria Ashworth

Figure 27- Changes available in the customers_live_changes internal low-latency table

Explanation of Data Sets after Compose Storage Task Change Data Capture execution

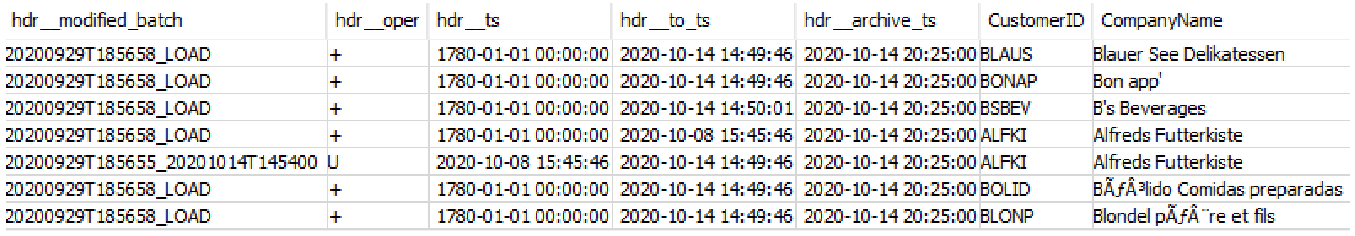
On execution of the CDC storage task, Qlik Compose for Data Lakes will process all the data from the landing customers_ct table that is within closed partitions. After execution of the CDC storage task, the data will be available in the customers and customers_archive table.



hdr__created_batch	hdr__modified_batch	hdr__oper	hdr__ts	CustomerID	CompanyName
20200929T185658_LOAD	20200929T185655_20201014T145400	U	2020-10-14 14:49:46	BOLID	BÃƒÂ³lido Comidas preparadas
20200929T185658_LOAD	20200929T185655_20201014T145400	U	2020-10-14 14:49:46	BONAP	Bon app'
20200929T185658_LOAD	20200929T185655_20201014T145400	U	2020-10-14 14:49:46	BLAUS	Blauer See Delikatessen
20200929T185658_LOAD	20200929T185655_20201014T145400	U	2020-10-14 14:49:46	ALFKI	Alfreds Futterkiste
20200929T185658_LOAD	20200929T185655_20201014T145400	U	2020-10-14 14:50:01	BSBEV	B's Beverages
20200929T185658_LOAD	20200929T185655_20201014T145400	U	2020-10-14 14:49:46	BLONP	Blondel pÃƒÂ¨re et fils
20200929T185655_20201014T145400	20200929T185655_20201014T145400	I	2020-10-14 14:53:31	CBMIL	ACME
20200929T185655_20201014T145400	20200929T185655_20201014T145400	I	2020-10-14 14:52:57	ERNSF	Ernst Handel
20200929T185655_20201014T145400	20200929T185655_20201014T145400	I	2020-10-14 14:53:31	FBMIL	ACME
20200929T185655_20201014T145400	20200929T185655_20201014T145400	I	2020-10-14 14:53:31	ABMIL	ACME
20200929T185655_20201014T145400	20200929T185655_20201014T145400	I	2020-10-14 14:53:30	FAMIL	Familia Arquibaldo

Figure 28- Changes available in the Delta table within the customers Storage schema

The customers_archive table contains the insert and updated records before the source data changes.



hdr__modified_batch	hdr__oper	hdr__ts	hdr__to_ts	hdr__archive_ts	CustomerID	CompanyName
20200929T185658_LOAD	+	1780-01-01 00:00:00	2020-10-14 14:49:46	2020-10-14 20:25:00	BLAUS	Blauer See Delikatessen
20200929T185658_LOAD	+	1780-01-01 00:00:00	2020-10-14 14:49:46	2020-10-14 20:25:00	BONAP	Bon app'
20200929T185658_LOAD	+	1780-01-01 00:00:00	2020-10-14 14:50:01	2020-10-14 20:25:00	BSBEV	B's Beverages
20200929T185658_LOAD	+	1780-01-01 00:00:00	2020-10-08 15:45:46	2020-10-14 20:25:00	ALFKI	Alfreds Futterkiste
20200929T185655_20201014T145400	U	2020-10-08 15:45:46	2020-10-14 14:49:46	2020-10-14 20:25:00	ALFKI	Alfreds Futterkiste
20200929T185658_LOAD	+	1780-01-01 00:00:00	2020-10-14 14:49:46	2020-10-14 20:25:00	BOLID	BÃƒÂ³lido Comidas preparadas
20200929T185658_LOAD	+	1780-01-01 00:00:00	2020-10-14 14:49:46	2020-10-14 20:25:00	BLONP	Blondel pÃƒÂ¨re et fils

Figure 29- Changes available in the table customers_archive within the Storage schema

With the 2nd generation Qlik Compose for Data Lakes solution, data is available in low-latency tables as soon as it is replicated to the Databricks target. Once the Qlik Compose for Data Lakes CDC storage task is executed, the data is available in the storage zone Databricks Delta schema tables. Thus, Qlik Compose for Data Lakes provides a highly scalable and flexible alternative solution for low-latency integration with Databricks Delta.

Determine which QDI method to integrate data with Databricks Delta

The Qlik Data Integration Platform provides two different methodologies for low-latency data with Databricks Delta. The user can use the method that is best for their use case.

Qlik Replicate is suited for use cases that require very light transformation of the data from source. Although changes are applied to the Databricks Delta target. Qlik Replicate does not provide the ability to create a historical data store for the data.

Qlik Compose for Data Lake 2nd generation solution is suited for use cases in which transformations and lookup values can be applied to data before loading in Databricks Delta tables. The solution will make change data capture data available in low-latency tables without execution of the Compose CDC data storage task. The execution of the CDC data storage task will merge the update transactions with the existing table. The delete transactions will be handled as logical deletes within the existing table.

The 2nd generation solution also provides the ability to provide a historical data store for the data lake. Having a full history of the data is very useful for many types of analytics and data science use cases that will be sourced from the data lake. Unless history has been captured and maintained, by a process like the Qlik Compose for Data Lakes 2nd generation solution, going back and getting history can be a very expensive proposition. The historical data store differs from Databricks Time Travel feature, which versions all operations written into a Databricks Delta table or directory. Databricks Time Travel seeks to resolve the challenges: audit data changes, reproduce reports for model training and rollback bad data for downstream consumers, not support trend analysis or strategic planning activities that historical data store supports. By default Delta tables only commit history for 30 days but if you execute the VACUUM feature on the Delta Table, you lose the ability to go back to a version older than the default 7 day retention period. With Qlik Compose for Data Lakes 2nd generation solution you can reproduce the history of your data from the persistent and low-latency views.

The Qlik Compose for Data Lakes 2nd generation solution also provides a cost-effective option to achieve low latency for high-scale operational data store use cases, particularly those where the tables to be managed in the data lake number in the hundreds or even thousands.

Conclusion

Qlik Data Integration Platform provides two ways to enable low-latency data integration with Databricks Delta. The first is a low-latency solution that utilizes Qlik Replicate to apply directly to the Databricks Delta tables. The second is a low-latency solution that uses Qlik Compose for Data lakes to produce current and historical data sets in the Databricks Delta Lake.

Appendix: Script for Source Tables

SQL Statements used to create source tables in your environment to test the solution. These statements may need to modify for your source RDBMS (these are written for MySQL). The source tables will be added to your Qlik Replicate Task.

1.Create Source Tables for Replication Methodology

```
);

CREATE TABLE salesdemo_.customers(
		CustomerID varchar(5) NOT NULL,
		CompanyName varchar(50) NULL,
		ContactName varchar(30) NULL,
		ContactTitle varchar(30) NULL,
		Address varchar(60) NULL,
		City varchar(15) NULL,
		Region varchar(15) NULL,
		PostalCode varchar(10) NULL,
		Country varchar(15) NULL,
		Phone varchar(24) NULL,
		Fax varchar(24) NULL,
PRIMARY KEY
(
		CustomerID
)
) ;
```

2. Insert an Initial sample set of records and then apply new inserts and update statements for CDC. (Labeled CDC in SQL comment.)

```
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'ALFKI', N'Alfreds Futterkiste', N'Maria Jones', N'Sales Representative', N'Obere Str. 57', N'Berlin', NULL, N'12209',
N'Germany', N'030-0074321', N'030-0076545' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'ANATR', N'Ana Trujillo Emparedados y helados', N'Ms. Ana Trujillo', N'Owner', N'Avda. de la Constitucii¿½n 2222',
N'Mï¿½xico D.F.', NULL, N'05021', N'Mexico', N'(5) 555-4729', N'(5) 555-3745' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'ANTON', N'Antonio Moreno Taquería', N'Antonio Moreno', N'Owner', N'Mataderos  2312', N'México D.F.', NULL,
N'05023', N'Mexico', N'(5) 555-3932', NULL );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'AROUT', N'Around the Horn', N'Thomas Hardy', N'Sales Representative', N'120 Hanover Sq.', N'London', NULL,
N'WA1 1DP', N'United Kingdom', N'(171) 555-7788', N'(171) 555-6750' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'BERGS', N'Berglunds snabbköp', N'Christina Berglund', N'Order Administrator', N'Berguvsvägen  8', N'Luleå', NULL,
N'S-958 22', N'Sweden', N'0921-12 34 65', N'0921-12 34 67' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'BLAUS', N'Blauer See Delikatessen', N'Hanna Moos', N'Sales Representative', N'Forsterstr. 57', N'Mannheim',
NULL, N'68306', N'Germany', N'0621-08460', N'0621-08924' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'BLONP', N'Blondel père et fils', N'Frédérique Citeaux', N'Marketing Manager', N'24, place Kléber', N'Strasbourg',
NULL, N'67000', N'France', N'88.60.15.31', N'88.60.15.32' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'BOLID', N'Bólido Comidas preparadas', N'Martín Sommer', N'Owner', N'C/ Araquil, 67', N'Madrid', NULL, N'28023',
N'Spain', N'(91) 555 22 82', N'(91) 555 91 99' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'BONAP', N'Bon app''', N'Laurence Lebihan', N'Owner', N'12, rue des Bouchers', N'Marseille', NULL, N'13008',
N'France', N'91.24.45.40', N'91.24.45.41' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'BOTTM', N'Bottom-Dollar Markets', N'Elizabeth Lincoln', N'Accounting Manager', N'23 Tsawassen Blvd.',
N'Tsawassen', N'BC', N'T2F 8M4', N'Canada', N'(604) 555-4729', N'(604) 555-3745' );
```

```
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'BSBEV', N'B''s Beverages', N'Victoria Ashworth', N'Sales Representative', N'Fauntleroy Circus', N'London', NULL,
N'EC2 5NT', N'United Kingdom', N'(171) 555-1212', NULL );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'CACTU', N'Cactus Comidas para llevar', N'Patricio Simpson', N'Sales Agent', N'Cerrito 333', N'Buenos Aires', NULL,
N'1010', N'Argentina', N'(1) 135-5555', N'(1) 135-4892' );

--- CDC Insert Statements


INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'CENTC', N'Centro comercial Moctezuma', N'Francisco Chang', N'Marketing Manager', N'Sierras de Granada 9993',
N'México D.F.', NULL, N'05022', N'Mexico', N'(5) 555-3392', N'(5) 555-7293' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'CHOPS', N'Chop-suey Chinese', N'Yang Wang', N'Owner', N'Hauptstr. 29', N'Bern', NULL, N'3012', N'Switzerland',
N'0452-076545', NULL );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'COMMI', N'Comércio Mineiro', N'Pedro Afonso', N'Sales Associate', N'Av. dos Lusíadas, 23', N'São Paulo', N'SP',
N'05432-043', N'Brazil', N'(11) 555-7647', NULL );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'CONSH', N'Consolidated Holdings', N'Elizabeth Brown', N'Sales Representative', N'Berkeley Gardens 12  Brewery ',
N'London', NULL, N'WX1 6LT', N'United Kingdom', N'(171) 555-2282', N'(171) 555-9199' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'DELME', N'Delete Me', NULL, N'Goodbye', NULL, NULL, NULL, NULL, NULL, NULL, NULL );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'DRACD', N'Drachenblut Delikatessen', N'Sven Ottlieb', N'Order Administrator', N'Walserweg 21', N'Aachen', NULL,
N'52066', N'Germany', N'0241-039123', N'0241-059428' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'DUMON', N'Du monde entier', N'Janine Labrune', N'Owner', N'67, rue des Cinquante Otages', N'Nantes', NULL,
N'44000', N'France', N'40.67.88.88', N'40.67.89.89' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'EASTC', N'Eastern Connection', N'Ann Devon', N'Sales Agent', N'35 King George', N'London', NULL, N'WX3 6FW',
N'United Kingdom', N'(171) 555-0297', N'(171) 555-3373' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'ERNSH', N'Ernst Handel', N'Roland Mendel', N'Sales Manager', N'Kirchgasse 6', N'Graz', NULL, N'8010', N'Austria',
N'7675-3425', N'7675-3426' );
INSERT sales.demo_customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country,
Phone, Fax) VALUES (N'FAMIA', N'Familia Arquibaldo', N'Aria Cruz', N'Marketing Assistant', N'Rua Orós, 92', N'São Paulo', N'SP', N'05442-
030', N'Brazil', N'(11) 555-9857', NULL );


--CDC Update Statements

UPDATE `sales`.`demo_customers` SET `ContactName` = 'Maria Jones-Drew1' WHERE (`CustomerID` = 'ALFKI');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Ms. Ana Trujillo-Sanchez' WHERE (`CustomerID` = 'ANATR');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Antonio Black' WHERE (`CustomerID` = 'ANTON');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Thomas Harding' WHERE (`CustomerID` = 'AROUT');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Christina Berglun' WHERE (`CustomerID` = 'BERGS');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Hanna Moo' WHERE (`CustomerID` = 'BLAUS');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Frédérique Citeaux-Vaughn' WHERE (`CustomerID` = 'BLONP');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Martín Sommer-Free' WHERE (`CustomerID` = 'BOLID');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Laurence Jones' WHERE (`CustomerID` = 'BONAP');
UPDATE `sales`.`demo_customers` SET `ContactName` = 'Elizabeth Happy' WHERE (`CustomerID` = 'BOTTM');
```

3. Create Source Tables for Compose 2nd generation solution methodology.

```
CREATE TABLE sales.customers(
        CustomerID varchar(5) NOT NULL,
        CompanyName varchar(50) NULL,
        ContactName varchar(30) NULL,
        ContactTitle varchar(30) NULL,
        Address varchar(60) NULL,
        City varchar(15) NULL,
        Region varchar(15) NULL,
        PostalCode varchar(10) NULL,
        Country varchar(15) NULL,
        Phone varchar(24) NULL,
        Fax varchar(24) NULL,
PRIMARY KEY
(
            CustomerID
)
) ;
```

4. Insert an Initial sample set of records and then apply new inserts and update statements for CDC. (Labeled CDC in SQL comment.)

```
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'ALFKI', N'Alfreds Futterkiste', N'Maria Jones', N'Sales Representative', N'Obere Str. 57', N'Berlin', NULL, N'12209', N'Germany', N'030-0074321', N'030-0076545' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'ANATR', N'Ana Trujillo Emparedados y helados', N'Ms. Ana Trujillo', N'Owner', N'Avda. de la Constituciï¿½n 2222', N'Mï¿½xico D.F.', NULL, N'05021', N'Mexico', N'(5) 555-4729', N'(5) 555-3745' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'ANTON', N'Antonio Moreno Taquería', N'Antonio Moreno', N'Owner', N'Mataderos  2312', N'México D.F.', NULL, N'05023', N'Mexico', N'(5) 555-3932', NULL );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'AROUT', N'Around the Horn', N'Thomas Hardy', N'Sales Representative', N'120 Hanover Sq.', N'London', NULL, N'WA1 1DP', N'United Kingdom', N'(171) 555-7788', N'(171) 555-6750' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'BERGS', N'Berglunds snabbköp', N'Christina Berglund', N'Order Administrator', N'Berguvsvägen  8', N'Luleå', NULL, N'S-958 22', N'Sweden', N'0921-12 34 65', N'0921-12 34 67' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'BLAUS', N'Blauer See Delikatessen', N'Hanna Moos', N'Sales Representative', N'Forsterstr. 57', N'Mannheim', NULL, N'68306', N'Germany', N'0621-08460', N'0621-08924' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'BLONP', N'Blondel père et fils', N'Frédérique Citeaux', N'Marketing Manager', N'24, place Kléber', N'Strasbourg', NULL, N'67000', N'France', N'88.60.15.31', N'88.60.15.32' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'BOLID', N'Bólido Comidas preparadas', N'Martín Sommer', N'Owner', N'C/ Araquil, 67', N'Madrid', NULL, N'28023', N'Spain', N'(91) 555 22 82', N'(91) 555 91 99' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'BONAP', N'Bon app''', N'Laurence Lebihan', N'Owner', N'12, rue des Bouchers', N'Marseille', NULL, N'13008', N'France', N'91.24.45.40', N'91.24.45.41' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'BOTTM', N'Bottom-Dollar Markets', N'Elizabeth Lincoln', N'Accounting Manager', N'23 Tsawassen Blvd.', N'Tsawassen', N'BC', N'T2F 8M4', N'Canada', N'(604) 555-4729', N'(604) 555-3745' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'BSBEV', N'B''s Beverages', N'Victoria Ashworth', N'Sales Representative', N'Fauntleroy Circus', N'London', NULL, N'EC2 5NT', N'United Kingdom', N'(171) 555-1212', NULL );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'CACTU', N'Cactus Comidas para llevar', N'Patricio Simpson', N'Sales Agent', N'Cerrito 333', N'Buenos Aires', NULL, N'1010', N'Argentina', N'(1) 135-5555', N'(1) 135-4892' );



---CDC Insert Statements
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'ERNSF', N'Ernst Handel', N'Roland Mendel', N'Sales Manager Sr', N'Kirchgasse 6', N'Graz', NULL, N'8010', N'Austria', N'7675-3425', N'7675-3426' );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'FAMIL', N'Familia Arquibaldo', N'Aria Cruz', N'Marketing Assistant II', N'Rua Orós, 92', N'São Paulo', N'SP', N'05442-030', N'Brazil', N'(11) 555-9857', NULL );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'CBMIL', N'ACME', N'John Doe', N'Marketing Assistant II', N'Rua Orós, 92', N'São Paulo', N'SP', N'05442-030', N'Brazil', N'(11) 555-9857', NULL );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'ABMIL', N'ACME', N'Jane Doe', N'Marketing Assistant I', N'Rua Orós, 92', N'São Paulo', N'SP', N'05442-030', N'Brazil', N'(11) 555-9857', NULL );
INSERT sales.customers (CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax) VALUES (N'FBMIL', N'ACME', N'Will Fox', N'Sales Representative', N'Rua Orós, 92', N'São Paulo', N'SP', N'05442-030', N'Brazil', N'(11) 555-9857', NULL );
```

```
-- CDC Update Statements
UPDATE `sales`.`customers` SET `PostalCode` = '68307' WHERE (`CustomerID` = 'BLAUS');
UPDATE `sales`.`customers` SET `PostalCode` = '67001' WHERE (`CustomerID` = 'BLONP');
UPDATE `sales`.`customers` SET `PostalCode` = '28024' WHERE (`CustomerID` = 'BOLID');
UPDATE `sales`.`customers` SET `PostalCode` = '13009' WHERE (`CustomerID` = 'BONAP');
UPDATE `sales`.`customers` SET `PostalCode` = 'EC2 5N' WHERE (`CustomerID` = 'BSBEV');
```

Qlik LEAD WITH DATA

About Qlik

Qlik's vision is a data-literate world, where everyone can use data and analytics to improve decision-making and solve their most challenging problems. Qlik provides an end-to-end, real-time data integration and analytics cloud platform to close the gaps between data, insights and action. By transforming data into active intelligence, businesses can drive better decisions, improve revenue and profitability, and optimize customer relationships. Qlik does business in more than 100 countries and serves over 50,000 customers around the world.

qlik.com