

Setting Up Streaming ETL to Snowflake Striim White Paper



Setting Up Streaming ETL to Snowflake Table of Contents

Introduction	1
Streaming ETL to Snowflake	1
Approaches — ETL/CDC/ELT	1
Continuously Integrating Transactional Data into Snowflake	1
Data Flow	1
Steps 1, 2 and 3	2
Steps 4 and 5	3
Steps 6 and 7	4
Steps 8, 9 and 10	5
Steps 11 and 12	6
Steps 13 and 14	7
Steps 15 and 16	8
Step 17	9
Conclusion	9



STREAMING ETL FOR SNOWFLAKE

Snowflake, the data warehouse built for the cloud, is designed to bring power and simplicity to your cloud-based analytics solutions, especially when combined with a streaming ETL to Snowflake running in the cloud.

Snowflake helps you make better and faster business decisions using your data on a massive scale, fueling data-driven organizations. Just take a look at Snowflake's example use cases and you can see how companies are creating value from their data with Snowflake. There's just one key caveat — how do you get your data into Snowflake in the first place?

APPROACHES — **ETL/CDC/ELT**

There are plenty of options when it comes to using data integration technologies, including ETL to Snowflake.

Let's start with traditional ETL. Now a 50+ year old legacy technology, ETL was the genesis of data movement and enabled batch, disk-based transformations. While ETL is still used for advanced transformation capabilities, the high latencies and immense load on your source databases leave something to be desired.

Next, there was Change Data Capture (CDC). Pioneered by the founders of Striim at their previous company, GoldenGate Software (acquired by Oracle), CDC technology enabled use cases such as zero downtime database migration and heterogeneous data replication. However, CDC lacks transformational capabilities, forcing you into an ELT approach — first landing the data into a staging area such as storage, and then transforming to its final form. While this works, the multiple hops increase your end-to-end latency and architectural complexity.

CONTINUOUSLY INTEGRATING TRANSACTIONAL DATA INTO SNOWFLAKE

Enter, Striim. Striim is an evolution from GoldenGate and combines the real-time nature of CDC with many of the transformational capabilities of ETL into a next-generation streaming solution for ETL to Snowflake and other analytics platforms, on-premises or in the cloud. Enabling real-time data movement into Snowflake, Striim continuously ingests data from on-premises systems and other cloud environments to Snowflake. In this quick start guide, we will walk you through, step-by-step, how to use Striim for streaming ETL to Snowflake by loading data in real time, whether you run Snowflake on Azure or AWS.

DATA FLOW

We'll get started with an on-premises Oracle to Snowflake application with in-line transformations and denormalization. This guide assumes you already have Striim installed either on-premises or in the cloud, along with your Oracle database and Snowflake account configured.

After installing Striim, there are a variety of ways to create applications, or data pipelines, from a source to a target. Here, I'll focus on using our pre-built wizards and drag-and-drop UI, but you can also build applications with the drag-and-drop UI from scratch, or using a declarative language using the CLI.

We will show how you can set up the flow between the source and target, and then how you can enrich records using an in-memory cache that's preloaded with reference data.

1. In the Add App page, select Start with Template.

💲 🔳 🛛 Create a New App				⊘ Help ♀ Admin 0
		CREATE A NEW APP		
	S	elect a method to create your App		
		<u></u>	P	
	Select Template to build your App	Custom build your App with Flow Designer to take full control of the pipeline	Import an existing App file (you can drag & drop your file here)	
	Start with Template	Start From Scratch	Import Existing App	
				-

2. In the following App Wizard screen, search for Snowflake.

💲 🚍 App Wizard		⑦ Help	只 Admin 0
	CREATE NEW APP 💿		
	Select a template to auto-generate your App		
	Search Templates Snow		
Quick Links Build with Flow Designer Download Agent for 3.8.6 Contract Support Download JOBC drivers	Streaming Integration to Azure 2		
	Streaming Integration to HDInsight Cracle DC to HDInsight Hadoop Oracle CDC to HDInsight Hadoop Oracle CDC to HDInsight Hadoop		
	Streaming Integration to Acure Storage		
	Streaming Integration from Microsoft SQL Server		Q

3. For this example, we'll choose Oracle CDC to Snowflake.

💲 🚍 App Wizard		⑦ Help	ር Admin
	CREATE NEW APP 🛛 🕥		
	Select a template to auto-generate your App		
	Search Templates Target: Snowflake ×]		
	Oracle CDC to Scoutfake Dracle CDC to Scoutfake		

.

4. Name the application whatever you'd like — we'll choose oracleToSnowflake. Go ahead and use the default admin Namespace. Namespaces are used for both application organization and enable a microservices approach when you have multiple data pipelines. Click Save.

💲 🚍 App Wizard		Help	Ω Admin	•
	CREATE NEW APP 💿			
Note: Before proceeding to the				
Name Namespace	New Application oracletosnowflake admin 🛛 🗶 🗸			

5. Follow the wizards, entering first your on-premises Oracle configuration properties, and then your Snowflake connection properties. In this case I'm migrating an Oracle orders table. Click Save, and you'll be greeted by our drag and drop UI with the source and target pre-populated. If you want to just do a straight source-to-target migration, that's it! However, we'll continue this example with enrichment and denormalization, editing our application using the connectors located on the left-hand side menu bar.



6. In this use case, we'll enrich the Orders table with another table of the same on-premises

Oracle database. Locate the Enrichment tab on the left-hand menu bar, and drag and drop the DB Cache to your canvas.



7. First, name the cache whatever you'd like—I chose salesRepCache. Then, specify the Type of your cache. In this case, my enrichment table contains three fields: ID, Name, and Email. Specify a Key to map. This tells Striim's in-memory cache how to position the data for the fastest possible joins. Finally, specify your Oracle Username, the JDBC Connection URL, your password, and the tables that you want to use as a cache. Click Save.

💲 ≡ Flow			⑦ Help
B Flow: oracleToSnowflak	Created ~ 15 B A		ᢙ 🛞 😁 🔠 Metadata Browser
Search Sources > Enrichment ~ DB Cache File Cache	admin.oracleToSnowNake	New Cac	he salesRepCache III panedSteam_T × v HDE Type ID Int × v \$ ×
HDFS Cache	Socontiskeelviter	SALES_REP	NAME String × v \$ × EMAIL String × v \$ ×
Transformation > Targets > Base Components >		ADD FIELD	CANCEL SAVE
		QUERY PROPER	TIES
		Key to map ① Skip invalid ①	SALES_REP_ID × ~
		Replicas Cache Refresh	Type replicas count or select 'All' v
		ADAPTER ()	DatabaseReader View Documentation
	Mercane Log 🔥 💶		Cancel Save

8. Now we'll go ahead and join our streaming CDC source with the static Database Cache. Click the circular stream beneath your Oracle source, and click Connect next CQ component.

\$ ≡	Flow					0	Help 🎗 Admin 🚺
Flow: or	acleToSnowflake	>	Created ~	16 B A	. 1 B	∩ €	- B Metadata Browser
Search		admin.oracleToSnowflake			Connect ne	xt compone	nt
Sources	>		onPremOracle OracleReader		Conne	ct next CQ com	ponent
8			(Conne	ct next Window	component
DB Cache	File Cache				Conne	ct next Target o	component
æ			SnowflakeWriter				Store component
Processing	>						
Transformation	>		salesRepCache				
Base Compone	nts >		- Dationaleteauer				

9. Application logic in Striim is expressed using Continuous Queries, or CQs. You do so using standard SQL syntax and optional Java functionality for custom scenarios. Unlike a query on a database where you run one query and receive one result, a CQ is constantly running, executing the query on an event-by-event basis as the data flows through Striim. Data can be easily pre-formatted or denormalized using CQs.

10. In this example, we are doing a few simple transformations of the fields of the streaming Oracle CDC source, as well as enriching the source with the database cache — adding in the SALES_REP_NAME and SALES_REP_EMAIL fields where the SALES_REP_ID of the streaming CDC source equals the SALES_REP_ID of the static database cache. Specify the name of the stream you want to output the result to, and click Save. Your logic here may vary depending on your use case.

\$ ≡	Flow			⑦ Help
Flow: or	acleToSnowflake		Created ~	B B A 🖀 🕞 🐠 🔠 🏭 Metadata Browser
Search Sources Enrichment DB Cache HDFS Cache	> ~ File Cache	admin.oracleToSnowNake	ontremonade oracleiteader	New CQ Name enkchQ QUERY v C C C C C C C C C C C C
Processing Transformation Targets Base Compone	> > > nts >		akesRepCache Databaselteader	WILEN TO_STRING(WETA(c, 'Operationshee')) is "BOD3!' NAD TO_STRING(WETA(c, 'Operationshee')) is "COULT' ADD TO_STRING(WETA(c, 'Operationshee')) is "COULT' PressCPTAD(cotats(4)) is "coult is "coult is "COULT' PressCPTAD(cotats(4)) is "coult is "coul

11. Lastly, we have to configure our SnowflakeTarget to read from the enrichedStream, not the original CDC source. Click on your Snowflake target and change the Input Stream from the Oracle source stream to your enriched stream. Click Save.

💲 ☴ Flow			⑦ Hel	p 只 Admin 0
Riow: oracleToSnowflake	> Created ~ 🗈 🖒 🛧	1 B	∩ ©-	SB Metadata Browser
Sources > Enrichment > DB Cache File Cache	edmin.oracleToSnowflake	Snowfi Input Stream Type Key Field Q ₄ data	akeTarget	stream × • ewor fiter ream
HDFS Cache	Utat all fields in excitched	۹ metadat ۹ userdata	a admin a ≈ oracSo admin	urcestream util.HashMap ×
Transformation		e before		lang.Object v
Targets > Base Components >		ADAPTER ①	SnowflakeWrit	ler × v
		Password () Tables ()	View Documenta	tion
		Connection URL	0 jdbc:snowflake	://vc65648.east-us-2.azure.
		Username ① Show optional pro	striim perties	
				•
menanistanid(0)	Miraal or A		Ca	ncel Save

12. Now you're good to go! In the top menu bar, click on Created and press Deploy App.



13. The deployment page allows you to specify where you want specific parts of your data pipeline

to run. In this case I have a very simple deployment topology — I'm just running Striim on my laptop, so I'll choose the default option.

\$ ≡								ি) Help	Ω Admin 0
Flow: ora		>) Created 👻			Ť	0 <u>0</u> 0	0	.	Hetadata Browser
Search										
Sources	~		Deploy oracleToSnowflake	×						
	Valler		DEPLOYMENT GROUP							
			default Agents							
10			oracleToSnowflake	node 🗸						
				_						
	{}		×	DEPLOY						
	TEXT		SnowflakeWriter							
File	Free Form Text									
GG										
GoldenGate	HDFS									
Enrichment	>									
Processing	>									
Transformation	>									
Base Componer	nts >		Messare Log A							

14. Click the eye next to your enrichedStream to preview your data as it's flowing through, and press Start App in the top menu bar.

§ ≡ Flow								Help	Ω Admin 0
Flow: oracleToSno	wflake 🕻			•	eployed 🛩			∩ ©-	88 Metadata Browser
admin.oracle105nowflai	e		oracteseure Oracteseare Conceleteader Conceleteader envictor Uscali Reds in envict Uscali Reds in envict Uscali Reds in envict StoryfaketViriter StoryfaketViriter	Ned	deploy App rt App Getaba Allasdor				
Preview data enriched	Stream								Θ _μ ⁿ ×
ORDER_ID	ORDER_DATE	ORDER_MODE	CUSTOMER_ID	ORDER_STATUS	ORDER_TOTAL	SALES_REP_NAME	SALES_REP_EMAIL	PRO	IOTION_ID
				Message	107 - v				0

:

15. Now that the apps running, let's generate some data. In this case I just have a sample data generator that is connecting to my source Oracle on-premises database.

	Quick Access	<u></u>
oracleCDCUtility.jav	a 🛛	
43 44 String 45 System. 46 stmt.ad 47 } 48	<pre>SQL = "INSERT INTO ORDERS (ORDER_ID, ORDER_DATE, ORDER_MODE, CUSTOMER_ID, ORDER_STATUS, ORDER_TOTAL, SALES. .aut.println(SQL); kdBatch(SQL);</pre>	,RE
49 for(int i=0 50 Strin 51 System 52 stmt. 53 } 54 String 56 String 57 System	<pre>p;:<20;i++) { gs SQL = "UPDATE ORDERS SET ORDER_STATUS=1 where ORDER_ID=" + (1000+(100*j+i+90)); m.out.println(SQL); addBatch(SQL); b;:<5;i++) { SQL = "DELETE FROM ORDERS where ORDER_ID="+ (1000+(100*j+i+95)); out.println(SQL); </pre>	
🖹 Problems 📮 Cons	sole 🛛 💿 🗶 💥 🕞 🛃 💭 🖅 🚍 - 📬 💷 - 📬 - 🖿	
<terminated> oracleCD</terminated>	CUtility [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home/bin/java (Jan 7, 2019, 6:00:36 AM)	

16. Data is flowing through the Striim platform, and you can see the enriched Sales Rep Name and Emails.

\$:	Flow								⑦ Help	ቢ Admin	n O
D FI	iow: oracleToSnowflake	>			O Runn	ing ~		A 🖶 🖻 🗚	\$ -	88 Metadata	Browser
admin.	oracle ToSnowflake			endedbourd OnciciReader ConcisR	imp(A) 5 estektegCar DatabaseRe d	ser 1000,0 met/4 he der					
Preview data enrichedStream											
	ORDER_ID	ORDER_DATE	ORDER_MODE	CUSTOMER_ID	ORDER_STATUS	ORDER_TOTAL	SALES_REP_NAME	SALES_REP_EMAIL	PRO	MOTION_ID	
.00	1099	1546913752759	Amazon	1099	ī	66274.49	Cassie Ferguson	Cassie@striim.com	7758	5	
9	1098	1546913692759	Amazon	1098	1	99518.09	Laurence Valencia	Laurence@striim.com	1344	8	
8	1097	1546913632759	Amazon	1097	7	48634.12	Randell Maldonado	Randell@striim.com	3748.	2	
7	1096	1546913572759	In-Store	1096	1	99140.65	Darrell Snyder	Darrell@striim.com	6204	í .	
6	1095	1546913512759	In-Store	1095	7 Message Log	~	Arline Gould	Arline@striim.com	6884	5	

17. Lastly, let's go to our Snowflake warehouse and just do a simple select * query. Data is now being continuously written to Snowflake.

*snowflake	Databases	Shares	Warehouses	> Worksheets	Q History						Partner Connect	? STRIIM SYSADMIN ~		
< Vorksheet 1	+ •											> [•		
Find database objects	¢ «	► Bun	All Queries Sa	aved a few second	s ago					Context: 11 S	(SADMIN MI + COMPUTE WHICH	DEMO DB 🛠 STAR 👻 ····		
Starting with										-				
DEMO_DB		i serec	t * Tron order	rs;										
INFORMATION_SCHEMA														
PUBLIC														
► Tables														
No Views in this Schema														
> Tables														
No Views in this Schema														
SNOWFLAKE_SAMPLE_DATA														
INFORMATION_SCHEMA														
No Tables in this Schema Views														
TPCDS_SF100TCL														
 Tables 														
No Views in this Schema														
TPCDS_SF10TCL														
TPCH_SF001														
TPCH SF10														
 Tables 														
No Views in this Schema														
TPCH_SF100														
TPCH_SF1000														
* TPCH_SF10000		Results Data	a Preview									+ Open History		
Tables		✓ Query ID	SQL 1.64	5	75,552 rows									
No Views in this Schema					4 Сору							Columns v 🖉		
		Row	0	RDER_ID ORD	ER_DATE	ORDER_MODE	CUSTOMER_ID	ORDER_STATUS	ORDER_TOTAL	SALES_REP_NAME	SALES_REP_EMAIL	PROMOTION_ID		
		1		1000 1545	149295077	Online	1000	1	78236.1	Amulfo Reeves	Amulfo@striim.com	65529		
		2		1795 1545	154995077	Amazon	1795	1	65442.69	Chandra Briggs	Chandra@striim.com	33999		
		3		1799 1545	155235077	Amazon	1799	1	43301.14	Francine Ray	Francine@striim.com	16880		
		4		1798 1545	155175077	In-Store	1798	1	6537.44	Silas Hernandez	Silas@striim.com	39286		
		5		1797 1545	155115077	Amazon	1797	1	69940.11	Emilia Grant	Emilia@striim.com	93205		
		6		1796 1545	155055077	Amazon	1796	1	63038.01	Bret Mckay	Bret@striim.com	45960		
		7		1795 1545	154995077	Amazon	1795	1	65442.69	Chandra Briggs	Chandra@striim.com	33999		
		8		1794 1544	806360791	Walmart	1794	1	15047.03	Pansy Parks	Pansy@strim.com	79538		

That's it. Without any coding, you now have set up streaming ETL to Snowflake to load data continuously, in real time.

Using real-time data delivered in a consumable format, Snowflake customers can further accelerate their time-to-insight. By streaming enterprise data to Snowflake with built-in scalability, security, and reliability, Striim simplifies adopting a modern, cloud data warehouse in the cloud for time-sensitive, operational decision making.

Copyright © 2019 Striim, Inc. All rights reserved. All product and company names are trademarks or registered trademarks of their respective holders.



Connect with us:

Swww.striim.com/blog/

- in www.linkedin.com/company/striim
- www.facebook.com/striim
- y www.twitter.com/striimteam
- www.striim.com/youtube

For more information, or to schedule a free trial, please contact us at **info@striim.com** or at **+1 (650) 241-0680**

www.striim.com