



CONTINUOUS DATA REPLICATION INTO CLOUD STORAGE WITH ORACLE GOLDENGATE

Michael Rainey | ITOUG Tech Days

February 2019

ABOUT ME

Michael Rainey

Senior Solutions Architect at Snowflake Computing

Oracle ACE Director 

Twitter: [@mRainey](https://twitter.com/mRainey)

Email: michael.rainey@snowflake.com



3 YEARS IN STEALTH + 3 YEARS GA

Founded 2012 by industry veterans with over 120 database patents



Over \$850M in venture funding from leading investors

First customers 2014, general availability 2015



900+ employees
Over 2000 customers today

Fun facts:

Queries processed in Snowflake per day:

100 million

Largest single table:

68 trillion rows

Largest number of tables single DB:

200,000

Single customer most data:

> 40PB

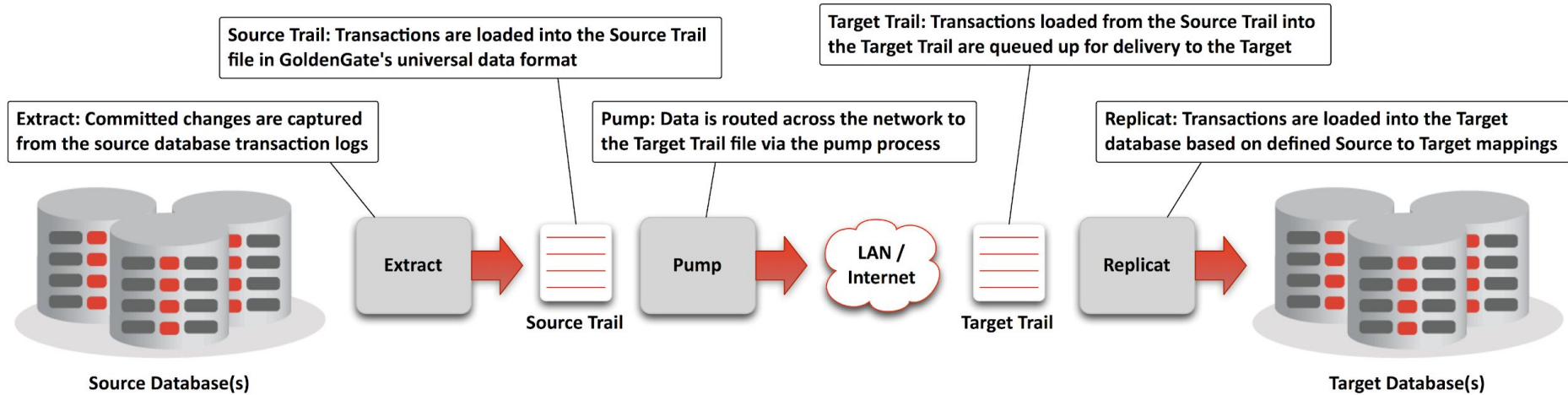
Single customer most users:

> 10,000





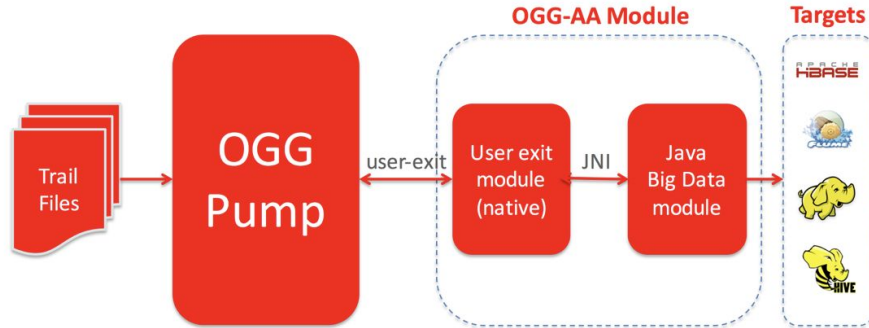
ORACLE GOLDENGATE



ORACLE GOLDENGATE FOR BIG DATA

In the beginning...

- Development of the “handler” was a manual effort using GoldenGate for Java adapter
- Very few targets available (HBase, Flume, Hive, etc)

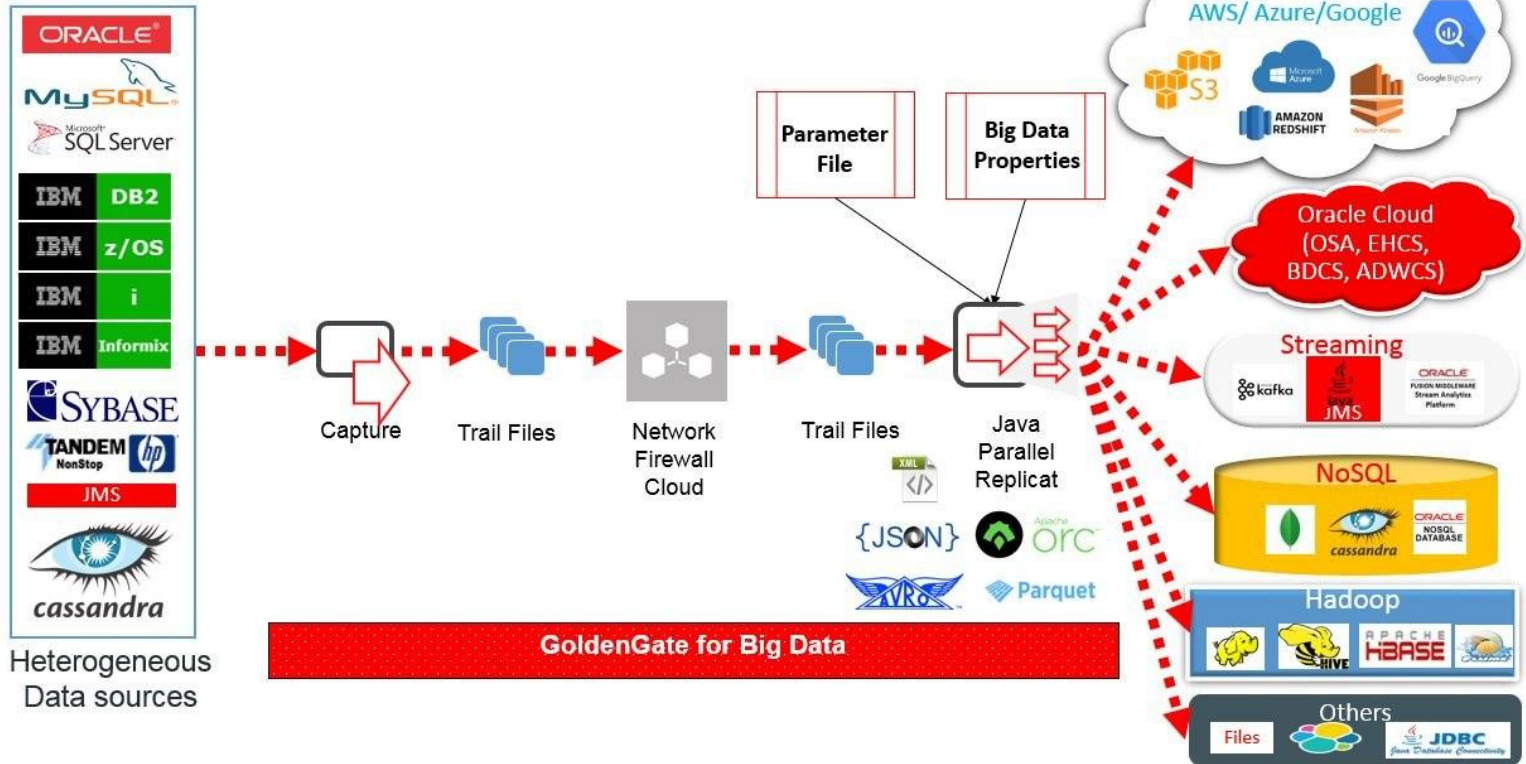


Over the years...

- More targets, more handlers, more automated!



ORACLE GOLDENGATE FOR BIG DATA



WRITE FILES WITH GOLDENGATE FOR BIG DATA



GOLDENGATE FOR BIG DATA HANDLERS

File Writer Handler

- Data is formatted and staged locally
- Maintain state - previously, all was lost
- Templated strings (substitution variables) can be used throughout properties file for naming

Event Handlers

- Transforms files written by File Writer Handler to target format (Parquet, ORC, etc)
- Connects to 3rd party application APIs
- Loads files to 3rd party applications (HDFS, S3, etc)

Pluggable Formatters

- Provide format options and metadata options



GOLDENGATE FOR BIG DATA SETUP

Extracting data from the source remains the same, no change

Install and setup GoldenGate for Big Data to handle replicat functionality

Create replicat parameter file and properties file

```
REPLICAT hrcsv

getEnv (JAVA_HOME)
SETENV(LD_LIBRARY_PATH = '/home/oracle/java/jdk1.8.0_131/jre/lib/amd64/server:/u01
/app/oracle/product/12.2/db_1/lib:/u01/app/oracle/product/12.2/db_1/jdk/jre/lib/am
d64/server:/gghome/oggd')

TARGETDB LIBFILE libggjava.so SET property=dirprm/hrcsv.properties

GROUPTRANSOPS 10000
MAP orcl.hr.*, TARGET *.*;
```



GOLDENGATE FOR BIG DATA PROPERTIES

Contains parameters required for...

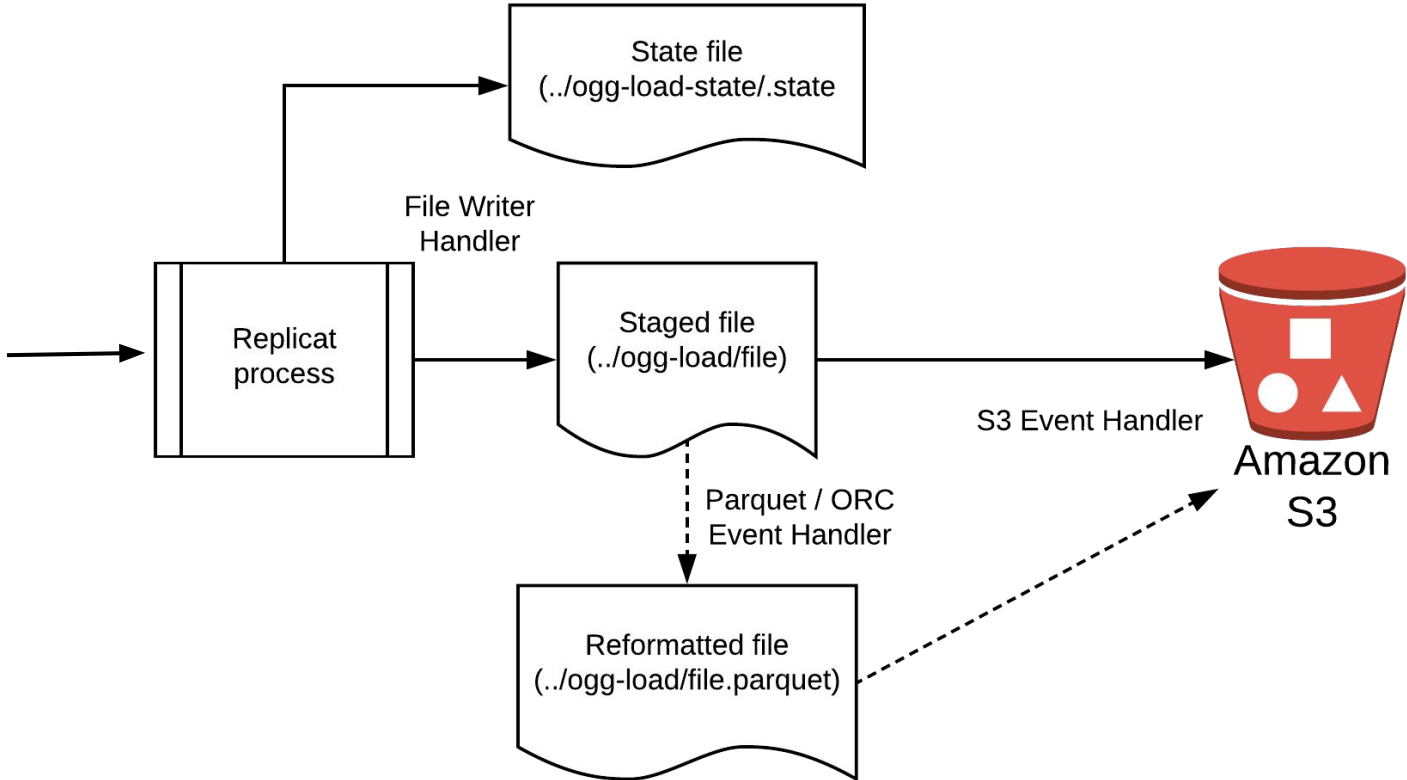
- ...creating the file and where it will be created
- ...file format and metadata options
- ...connection to final target and additional parameters required

Create with name that matches replicat parameter file name

- Example: payroll.prm / payroll.properties



REPLICAT TO TARGET PROCESS



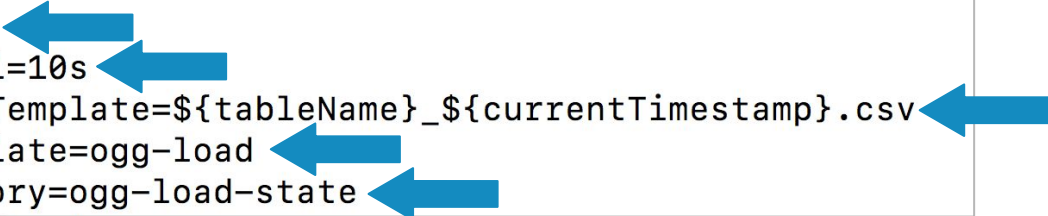
FILE WRITER HANDLER STATE

```
#File Writer Handler
#Tue Jan 29 08:22:47 EST 2019
fw.active=true
fw.offset=659
fw.firstwrite=2019-01-29 08\:22\:47.624
fw.cntoperations=4
fw.lastwrite=2019-01-29 08\:22\:47.626
fw.datafilename=EMPLOYEE_2019-01-29_08-22-47.624.csv
fw.tablename=HR.EMPLOYEE
fw.datafiledir=ogg-load
fw.lastwrite.ms=1548768167626
fw.format=bytes
fw.firstwrite.ms=1548768167624
fw.uuid=12b0cabd-86f2-44b7-920a-2cba7775bbfa
fw.cntupdates=4
12b0cabd-86f2-44b7-920a-2cba7775bbfa.state (END)
```



GOLDENGATE FOR BIG DATA PROPERTIES

```
gg.handlerlist=filewriter  
gg.handler.filewriter.type=filewriter  
gg.handler.filewriter.fileRollInterval=10s  
gg.handler.filewriter.fileNameMappingTemplate=${tableName}_${currentTimestamp}.csv  
gg.handler.filewriter.pathMappingTemplate=ogg-load  
gg.handler.filewriter.stateFileDirectory=ogg-load-state
```



GOLDENGATE FOR BIG DATA PROPERTIES

```
gg.handlerlist=filewriter
gg.handler.filewriter.type=filewriter
gg.handler.filewriter.fileRollInterval=10s
gg.handler.filewriter.fileNameMappingTemplate=${tableName}_${currentTimestamp}.csv
gg.handler.filewriter.pathMappingTemplate=ogg-load
gg.handler.filewriter.stateFileDirectory=ogg-load-state
```

```
gg.handler.filewriter.format=delimitedtext
gg.handler.filewriter.format.fieldDelimiter=,
gg.handler.filewriter.format.lineDelimiter=CDATA[\n]
gg.handler.filewriter.format.wrapStringsInQuotes=true
gg.handler.filewriter.format.pkUpdateHandling=update
gg.handler.filewriter.format.includePosition=false
gg.handler.filewriter.format.includeTableName=false
gg.handler.filewriter.format.iso8601Format=false
```



Pluggable
Formatter



GOLDENGATE FOR BIG DATA PROPERTIES

```
gg.handlerlist=filewriter
gg.handler.filewriter.type=filewriter
gg.handler.filewriter.fileRollInterval=10s
gg.handler.filewriter.fileNameMappingTemplate=${tableName}_${currentTimestamp}.csv
gg.handler.filewriter.pathMappingTemplate=ogg-load
gg.handler.filewriter.stateFileDirectory=ogg-load-state
```

```
gg.handler.filewriter.format=delimitedtext
gg.handler.filewriter.format.fieldDelimiter=,
gg.handler.filewriter.format.lineDelimiter=CDATA[\n]
gg.handler.filewriter.format.wrapStringsInQuotes=true
gg.handler.filewriter.format.pkUpdateHandling=update
gg.handler.filewriter.format.includePosition=false
gg.handler.filewriter.format.includeTableName=false
gg.handler.filewriter.format.iso8601Format=false
```

```
gg.handler.filewriter.finalizeAction=rename ←
gg.handler.filewriter.fileRenameMappingTemplate=${tableName}_${currentTimestamp}.csv
gg.handler.filewriter.eventHandler=s3 ←
goldengate.userexit.writers=javawriter
```



GOLDENGATE FOR BIG DATA PROPERTIES

```
gg.eventhandler.s3.type=s3
gg.eventhandler.s3.region=us-west-2
gg.eventhandler.s3.bucketMappingTemplate=oggcsv
gg.eventhandler.s3.pathMappingTemplate=${tableName}_${currentTimestamp}
#gg.handler.s3.customMessageGrouper=oracle.goldengate.handler.s3.s3JsonTxMessageGrouper
gg.classpath=/gghome/oggd/dirprm/:/home/oracle/aws-java-sdk-1.11.395/lib/aws-java-sdk-1.11.395.jar:/home/oracle/aws-java-sdk-1.11.395/lib/*:/home/oracle/aws-java-sdk-1.11.395/third-party/lib/*:/u01/userhome/oracle/aws-java-sdk-1.11.395/third-party/lib/jackson-annotations-2.6.0.jar
gg.log=log4j
gg.log.level=DEBUG
javawriter.bootoptions=-Xmx512m -Xms32m -Djava.class.path=.:ggjava/ggjava.jar -Daws.accessKeyId=
-Daws.secretKey=
```



LESSONS LEARNED

Cloud storage security (S3)

- User/role must be able to list buckets and create buckets, along with ability to write files

Properties file syntax

- Not many examples besides in the documentation...which can be incorrect as well!

The client ID and secret can be set as Java properties in the Java Adapter properties file as follows:

```
javawriter.bootoptions=-Xmx512m -Xms32m
-Djava.class.path=ggjava/ggjava.jar
-Daws.accessKeyId=your_access_key
-Daws.secretKey=your_secret_key
```

Add all required properties

- For example, `goldengate.userexit.writers=javawriter` is required, but not in the docs



USING THE DATA IN CLOUD STORAGE



AWS GLUE

A fully managed extract, transform, and load (ETL) service

Data Catalog

- Central repository to store structural and operational metadata for all your data assets

Crawlers

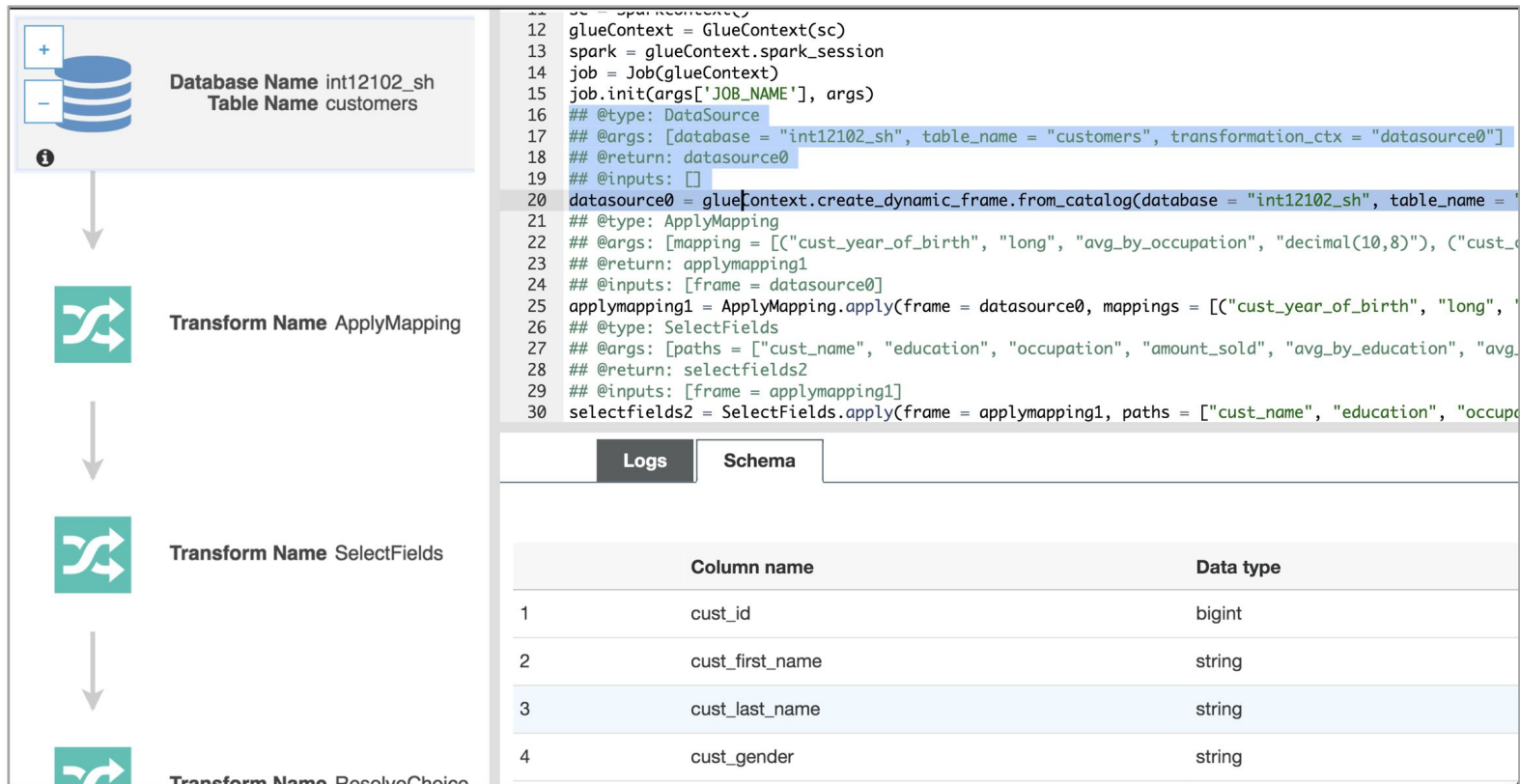
- Connects to data stores, determines the data structures, and writes tables into the Data Catalog

Jobs

- Business logic that performs the extract, transform, and load (ETL) work
- Developed using PySpark or Scala as scripts, generated by AWS Glue
- Built-in transforms used to process data



AWS GLUE JOB



AMAZON ATHENA

Serverless query engine for reporting and analytics

Uses Presto DB as the underlying query engine

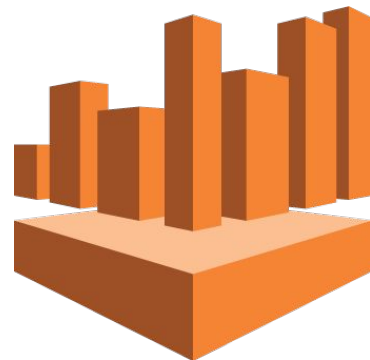
Supports CSV, JSON, Gzip files and columnar formats like Apache Parquet

- Use Athena's own catalog or the centralized AWS Glue catalog

Serverless query engine

- Performance scales “automatically” based on query profiling
- Pay as you query - based on data scanned
- SELECT only - no DML
- Follow best practices for query optimization

Integrates with BI tools like Tableau, Looker, AWS QuickSight, etc. for advanced reports and visualizations



AMAZON ATHENA

Results



	OP_TS	CURRENT_TS	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
1	2019-01-10 02:32:59.000000	2019-01-10 02:33:02.179000	172	Elizabeth	Bates	EBATES	011.44.1343.529268	1999-03-24 00:00:00	SA_REP	7300.0	148	80
2	2019-01-10 02:32:59.000000	2019-01-10 02:33:02.179001	168	Lisa	Ozer	LOZER	011.44.1343.929268	1997-03-11 00:00:00	SA_REP	11500.0	148	80
3	2019-01-10 02:33:06.000000	2019-01-10 02:33:11.197000	173	Sundita	Kumar	SKUMAR	011.44.1343.329268	2000-04-21 00:00:00	SA_REP	6100.0	148	80
4	2019-01-10 02:33:06.000000	2019-01-10 02:33:11.198000	174	Ellen	Abel	EABEL	011.44.1644.429267	1996-05-11 00:00:00	SA_REP	11000.0	149	80
5	2019-01-10 02:33:06.000000	2019-01-10 02:33:11.198001	173	Sundita	Kumar	SKUMAR	011.44.1343.329268	2000-04-21 00:00:00	SA_REP	6100.0	148	80
6	2019-01-10 02:33:06.000000	2019-01-10 02:33:11.199000	164	Mattea	Marvins	MMARVINS	011.44.1346.329268	2000-01-24 00:00:00	SA_REP	7200.0	147	80
7	2019-01-10 02:32:03.000000	2019-01-10 02:32:09.024000	122	Payam	Kaufling	PKAUFLIN	650.123.3234	1995-05-01 00:00:00	ST_MAN	7900.0	100	50
8	2019-01-10 02:32:03.000000	2019-01-10 02:32:09.027000	123	Shanta	Vollman	SVOLLMAN	650.123.4234	1997-10-10 00:00:00	ST_MAN	6500.0	100	50
9	2019-01-10 02:32:03.000000	2019-01-10 02:32:09.027001	124	Kevin	Mourgos	KMOURGOS	650.123.5234	1999-11-16 00:00:00	ST_MAN	5800.0	100	50
10	2019-01-10 02:32:03.000000	2019-01-10 02:32:09.030000	125	Julia	Nayer	JNAYER	650.124.1214	1997-07-16 00:00:00	ST_CLERK	3200.0	120	50



AWS QUICKSIGHT

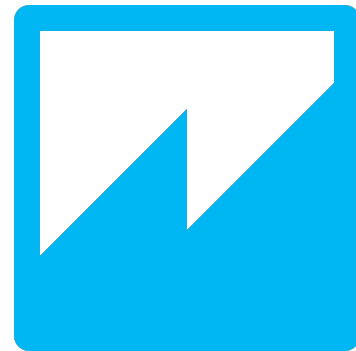
BI service used to create data visualizations and interactive dashboards for insightful analysis

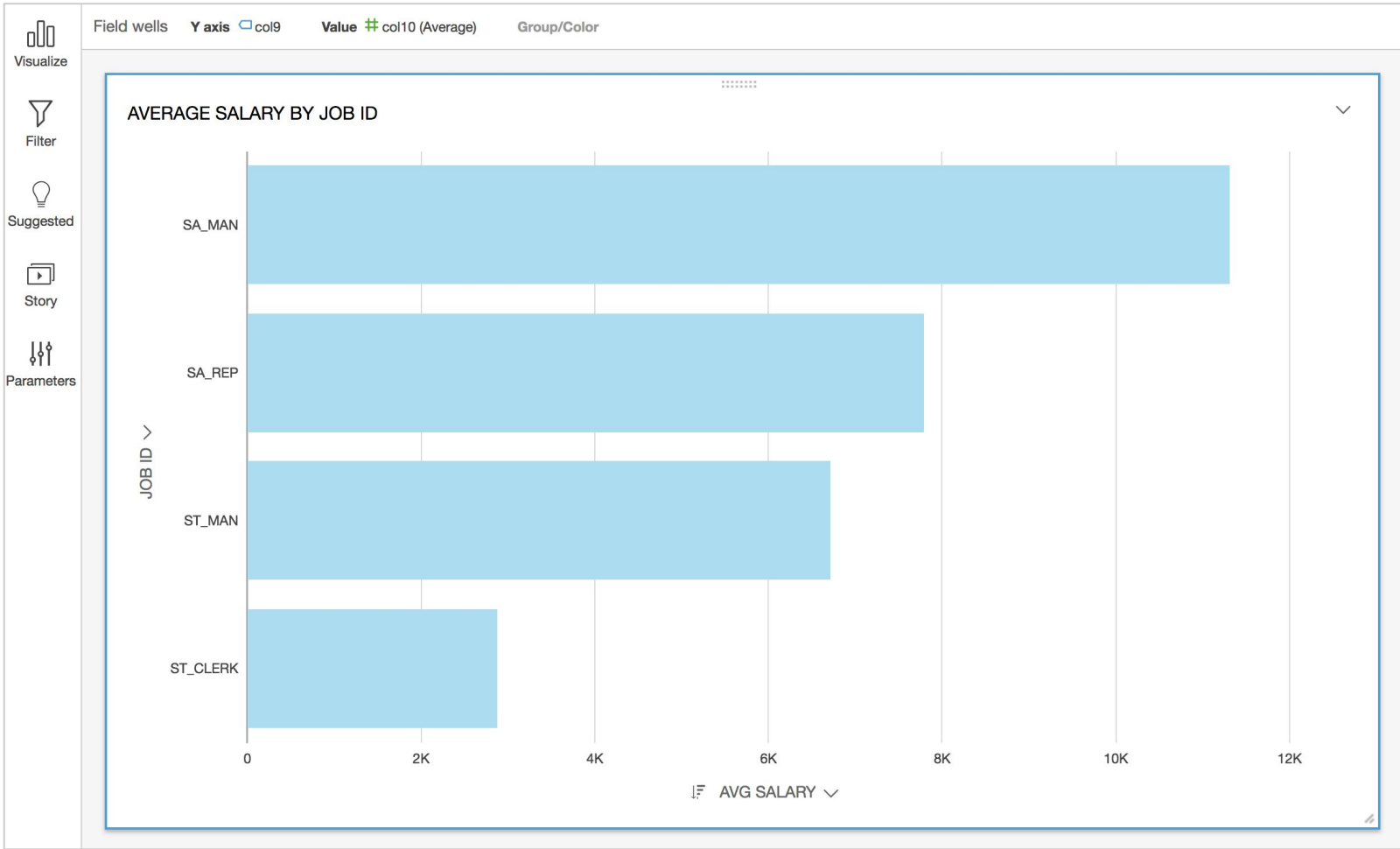
Connects to AWS sources (Athena, S3, etc), SQL databases, and SaaS applications (Salesforce, etc)

Uses 'pay-per-session' pricing for cost-effective user access

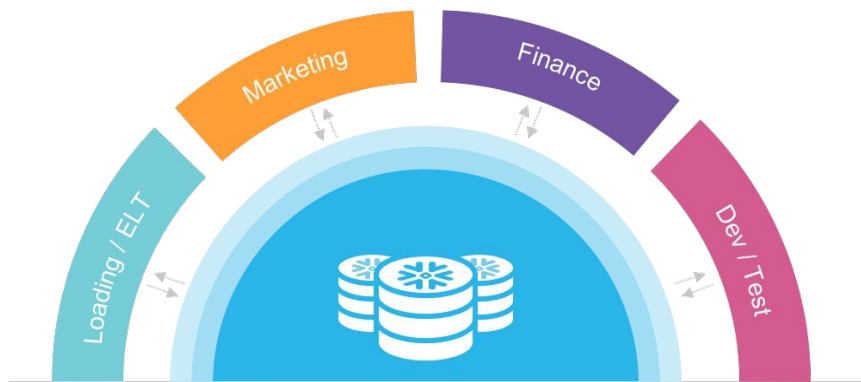
Create and publish interactive dashboards, share reports via email, or embed analytics into customer applications

Powered by super-fast, parallel, in-memory calculation engine (SPICE)





SNOWFLAKE DATA WAREHOUSE



Storage separated from compute

- Centralized, scale-out storage that expands and contracts automatically

Resize compute instantly

- Scale up/down or turn off when not in use

Multiple clusters access data without contention

- ETL, reporting, data science, and applications all running at the same time without performance impact.

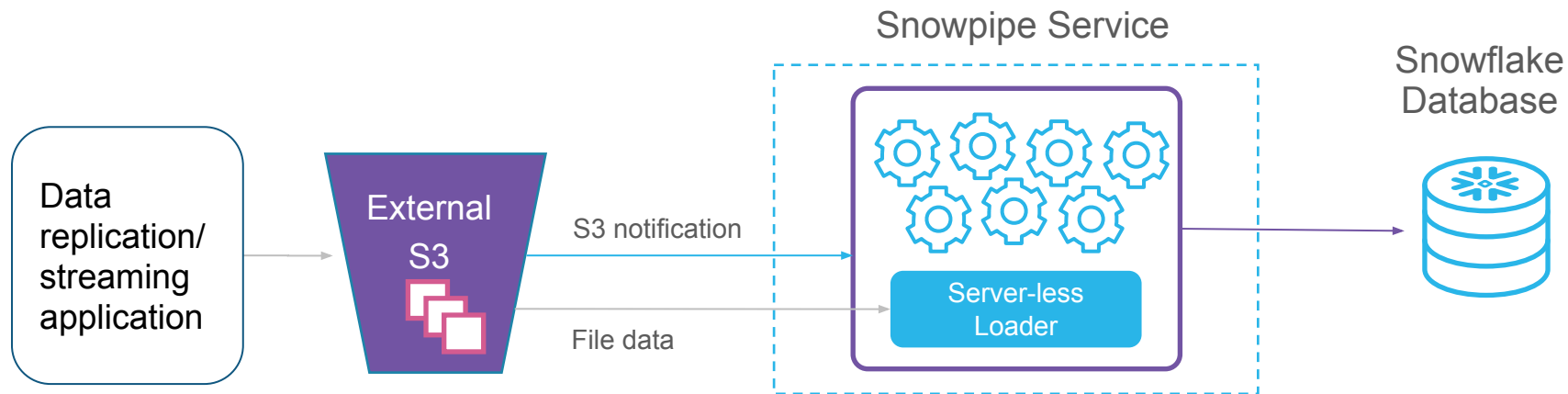
Centralized management

- Separate metadata from storage and compute

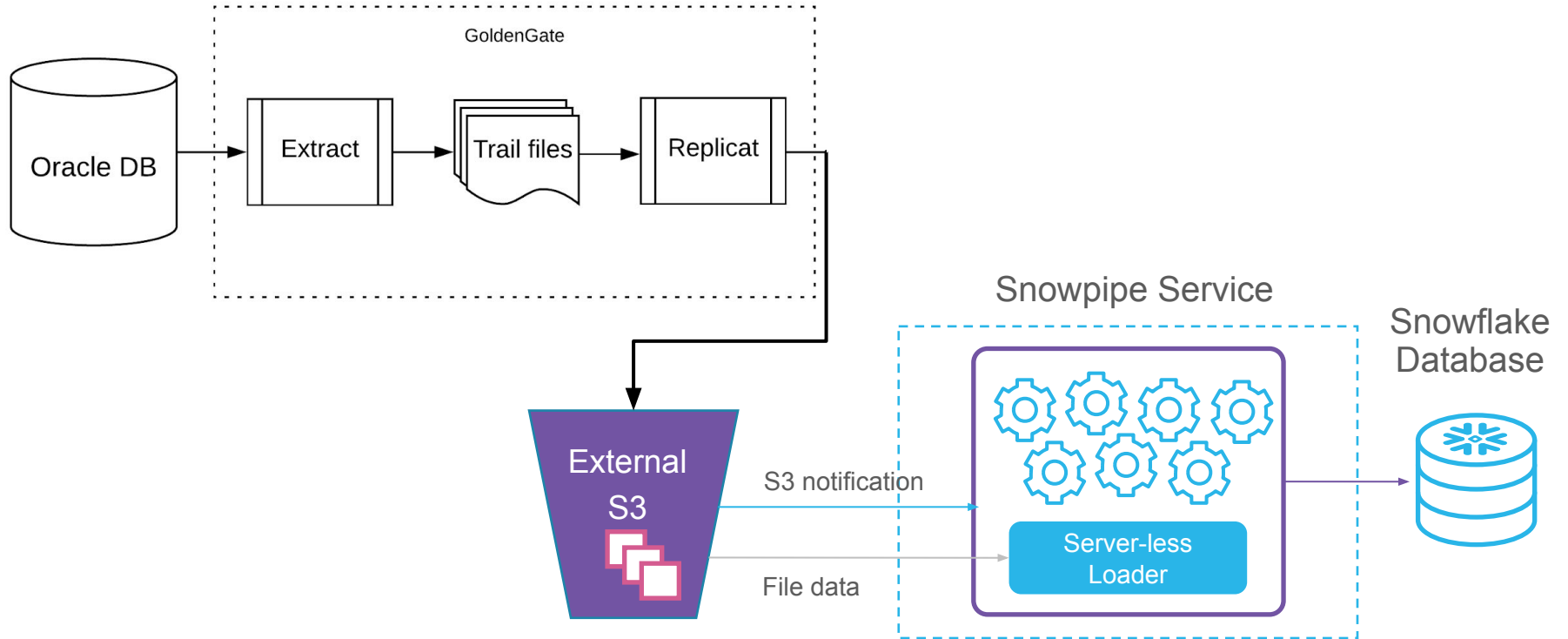
Full transactional consistency (ACID)



AUTO-INGEST WITH SNOWPIPE



CONTINUOUS REPLICATION TO SNOWFLAKE



MERGE REPLICATED DATA

```
MERGE INTO csvtarget tgt
USING (
  WITH v AS (
    SELECT x.*
           ,ROW_NUMBER() OVER (PARTITION BY employee_id ORDER BY current_ts DESC) AS row_rank
    FROM PUBLIC.oggcsvtarget x
  )
  SELECT *
  FROM v
  WHERE row_rank = 1
  ORDER BY employee_id
) src
ON tgt.employee_id = src.employee_id
```



MERGE REPLICATED DATA

```
WHEN MATCHED
  AND src.op_type = 'D'
  THEN
    DELETE
```



MERGE REPLICATED DATA

```
WHEN MATCHED
  AND src.op_type = 'U'
  THEN
    UPDATE
    SET tgt.batch_ts = src.batch_ts
      ,tgt.employee_id = src.employee_id
      ,tgt.last_name = src.last_name
      ,tgt.first_name = src.first_name
      ,tgt.email = src.email
      ,tgt.phone_number = src.phone_number
      ,tgt.department_id = src.department_id
      ,tgt.manager_id = src.manager_id
      ,tgt.job_id = src.job_id
      ,tgt.salary = src.salary
      ,tgt.commission_pct = src.commission_pct
      ,tgt.hire_date = src.hire_date
```



MERGE REPLICATED DATA

```
WHEN NOT MATCHED
  THEN
    INSERT (
      batch_ts
      ,employee_id
      ,last_name
      ...
      ,hire_date
    )
  VALUES (
    src.batch_ts
    ,src.employee_id
    ,src.last_name
    ...
    ,src.hire_date
  );
```





MORE INFORMATION

GoldenGate for Big Data docs:

<https://docs.oracle.com/goldengate/bd123210/gg-bd/index.html>

Oracle Data Integration blog:

<https://blogs.oracle.com/dataintegration/data-integration>

Continuous Data Replication into Snowflake with Oracle Goldengate blog post:

<https://www.snowflake.com/blog/continuous-data-replication-into-snowflake-with-oracle-goldengate/>





THANK YOU



DISCOVER THE PERFORMANCE, CONCURRENCY, AND SIMPLICITY OF SNOWFLAKE

As easy as 1-2-3!

- 01 Visit Snowflake.com
- 02 Click “Try for Free”
- 03 Sign up & register

Snowflake is the only data warehouse built for the cloud. You can automatically scale compute up, out, or down—independent of storage. Plus, you have the power of a complete SQL database, with zero management, that can grow with you to support all of your data and all of your users. With Snowflake On Demand™, pay only for what you use.

Sign up and receive
\$400 worth of free
usage for 30 days!

