**Design an Anomalous Trade Detection System to detect Wash Trades**

-----------------------------------------------------------------------------------------------

This design explores using Oracle Cloud Infrastructure to build an anomalous trade detection system for detecting wash trades, where the same quantity of a security gets bought and sold(or vice-versa) in a short period of time, to create fake activity in the marketplace.

Anomalous Trade is the identification of rare events, items or observations which is suspicious because they are differ from standard or normal patterns of trade.

**Problem Statement**

To detect anomalous trades (wash trade) where a buy and sell transaction occurs for the same fund, security and trade date within a predefined time slot.

### Understanding the Business Use Case

In today’s world, security is a critical aspect for any digital transaction and trading where valuation of any fund can fluctuate by any anomalous activities and it can affect even many business needs. We need to stay ahead of advanced attacks, so our customers and financial partners have confidence that transactions are processed securely. When anomaly detection is in place, data sources are integrated into a centralised platform, giving you total visibility into performance and operations and revealing critical security vulnerabilities.

Previously, analytics team used to detect the anomaly trade manually, and in that case, there were only a few standards any organization needed to detect the trade data, and the datasets were limited and manageable for any analytics team.

For example, for market manipulation in which an investor simultaneously sells and buys the same financial instruments to create misleading & artificial activity in the marketplace.

Here, an investor will either place a buy & sell the same quantity of stock in a very short interval of time, or vice-versa. Without a robust & a fool-proof solution to quickly detect and document wash trades from a constant stream of trade information every day, it could prove very troublesome for the stock market to function legitimately.

### Understand the Solution

Modern trading systems need automated detection for anomalous trade data to provide accurate detection of fraudulent trades regardless of how much data is being streamed in a certain amount of time. An automated anomaly detection system includes detection and notification to the trader and keeps record of the anomalous trade datasets thereby eliminates the need for the analytics team to analyse large data.

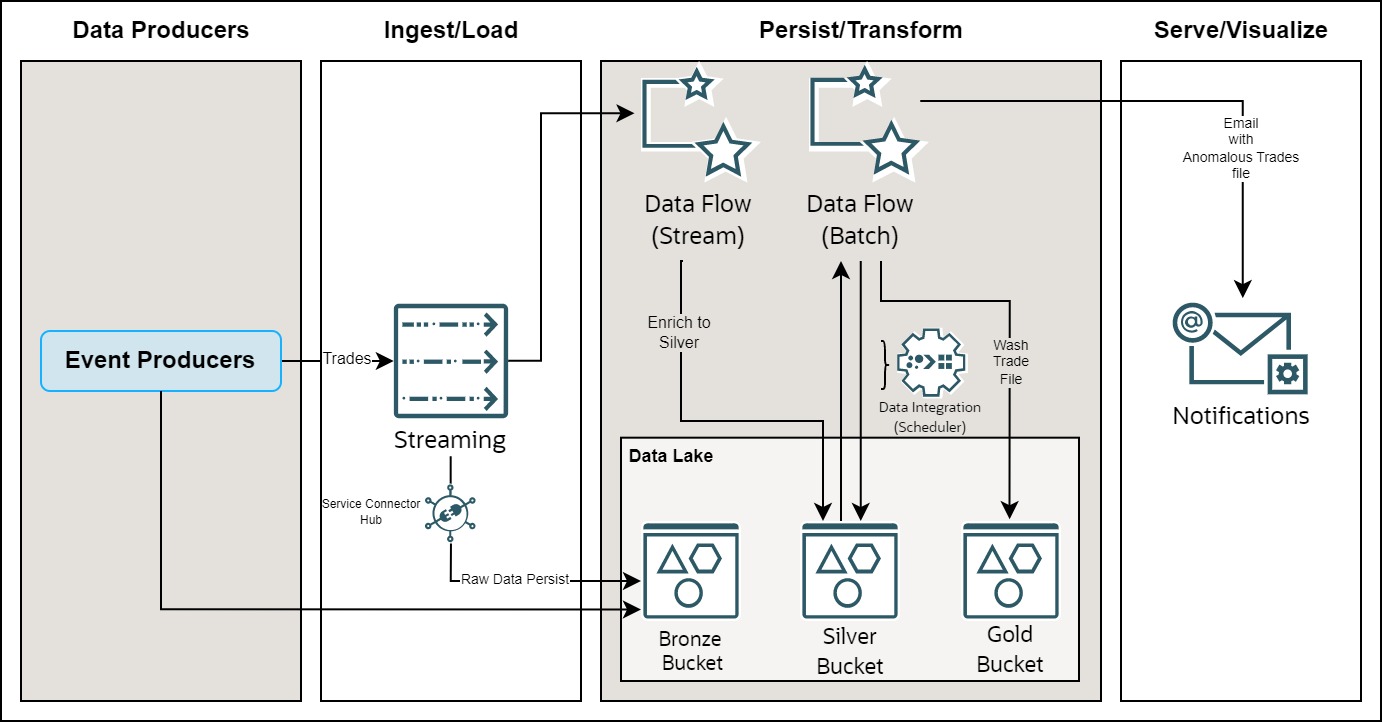
With the combination of OCI Streaming and OCI Data Flow, a constant stream of trade data can be continuously streamed and processed and then the detected wash trades can either be stored or be sent in a email which would act as an alert. OCI Data Integration can be coupled with OCI Data Storage to further refine raw trade data and to persist it if the need arises.

This whole architecture with the help of some OCI services would act as a low-cost and modern solution to swiftly detect anomalous transactions occurring during a whole day of trade activity.

### Example Architecture

The Architecture below shows the flow of the anomalous trade detection system initially the stream of trade data is ingested by a data flow application of the type STREAM and is also converted into a static batch file and stored into the silver storage bucket. This is done for the trades to get timestamped, so that the wash trades can be identified by finding the trades coming in the selected time interval. The stream data is simultaneously also stored in the bronze storage bucket as raw data. The silver bucket files then go through another data flow application, of the type BATCH, and it processes the file and stores the wash trade file into Gold bucket, As well as takes the same processed file and attaches it into an Email, Writes up a body for that email and then sends it to the respective recipients.

Data integration is used to schedule the batch data flow application to run automatically every 30 minutes. Further refinement of the data (aggregation and/or data grouping) is also done by the data flow applications and stored in the data lake for future use.



Here is a brief of all the services used to create this system in a cloud infrastructure: -

* **OCI Streaming**

The Oracle Cloud Infrastructure Streaming service provides a fully managed, scalable, and durable solution for ingesting and consuming high-volume data streams in real-time. It can be used for Messaging, Metric and log ingestion, web or mobile activity data ingestion and Infrastructure and apps event processing.

* **OCI Storage**

It is an ideal storage platform to store very large amounts of data. It is two-tier architecture: -

Standard Tier: Gives the most recent copy of the data when retrieved & the Data retrieval is instantaneous.

Archive Tier: Rarely accessed data is stored and the minimum retention period is 90 days. Data retrieval is not instantaneous.

* **OCI Data Flow**

Oracle Cloud Infrastructure (OCI) Data Flow is a fully managed Apache Spark service that performs processing tasks on extremely large datasets—without infrastructure to deploy or manage. Developers can also use Spark Streaming to perform cloud ETL on their continuously produced streaming data.

* **Spark Streaming**

Spark Streaming is used to process real-time data. Streaming applications require continuous execution for a long period of time that often extends beyond 24 hours, and might be as long as weeks or even months. In case of unexpected failures, streaming applications must restart from the point of failure without producing incorrect computational results. Data Flow relies on Spark structured streaming check-pointing to record the processed offset which can be stored in your Object Storage bucket.

* **Data Integration**

Data Integration is a fully managed, multi-tenant service that helps data engineers and ETL developers with common extract, transform, and load (ETL) tasks such as ingesting data from a variety of data assets; cleansing, transforming, and reshaping that data; and efficiently loading it to target data assets

* **Data Lake**

A data lake is a centralized, highly flexible storage repository that stores large amounts of structured and unstructured data in its raw, original, and unformatted form.

It offers several benefits, such as:

Data consolidation, Data flexibility, Cost saving and also support for a wide variety of data science and machine learning use cases.