# User Guide Of Data Validation Service

* [Background](#UserGuideOfDataValidationService-Backgr)
* [Architecture](#UserGuideOfDataValidationService-Archit)
* [Use Case](#UserGuideOfDataValidationService-UseCas)
  + [Reporting Team - Cross Check](#UserGuideOfDataValidationService-Report)
  + [Fraud Team - Detailed metrics report](#UserGuideOfDataValidationService-FraudT)
  + [Merch Team - feature consistency](#UserGuideOfDataValidationService-MerchT)
* [Steps](#UserGuideOfDataValidationService-Steps)
  + [Step1 - Create metrics rule](#UserGuideOfDataValidationService-Step1-)
  + [Step2 - Create Validation rule](#UserGuideOfDataValidationService-Step2-)
    - [Select Quality - Data Validation in left navigation bar → Click Add button](#UserGuideOfDataValidationService-Select)
    - [Fill in basic information](#UserGuideOfDataValidationService-Fillin)
    - [Add Accuracy rule](#UserGuideOfDataValidationService-AddAcc)
    - [Add Consistency rule](#UserGuideOfDataValidationService-AddCon)
    - [Add triggers parameters](#UserGuideOfDataValidationService-Addtri)
    - [Add notifications](#UserGuideOfDataValidationService-Addnot)
    - [Select extra settings](#UserGuideOfDataValidationService-Select)
* [Examples](#UserGuideOfDataValidationService-Exampl)
  + [Data Consistency](#UserGuideOfDataValidationService-DataCo)
  + [Data Accuracy](#UserGuideOfDataValidationService-DataAc)

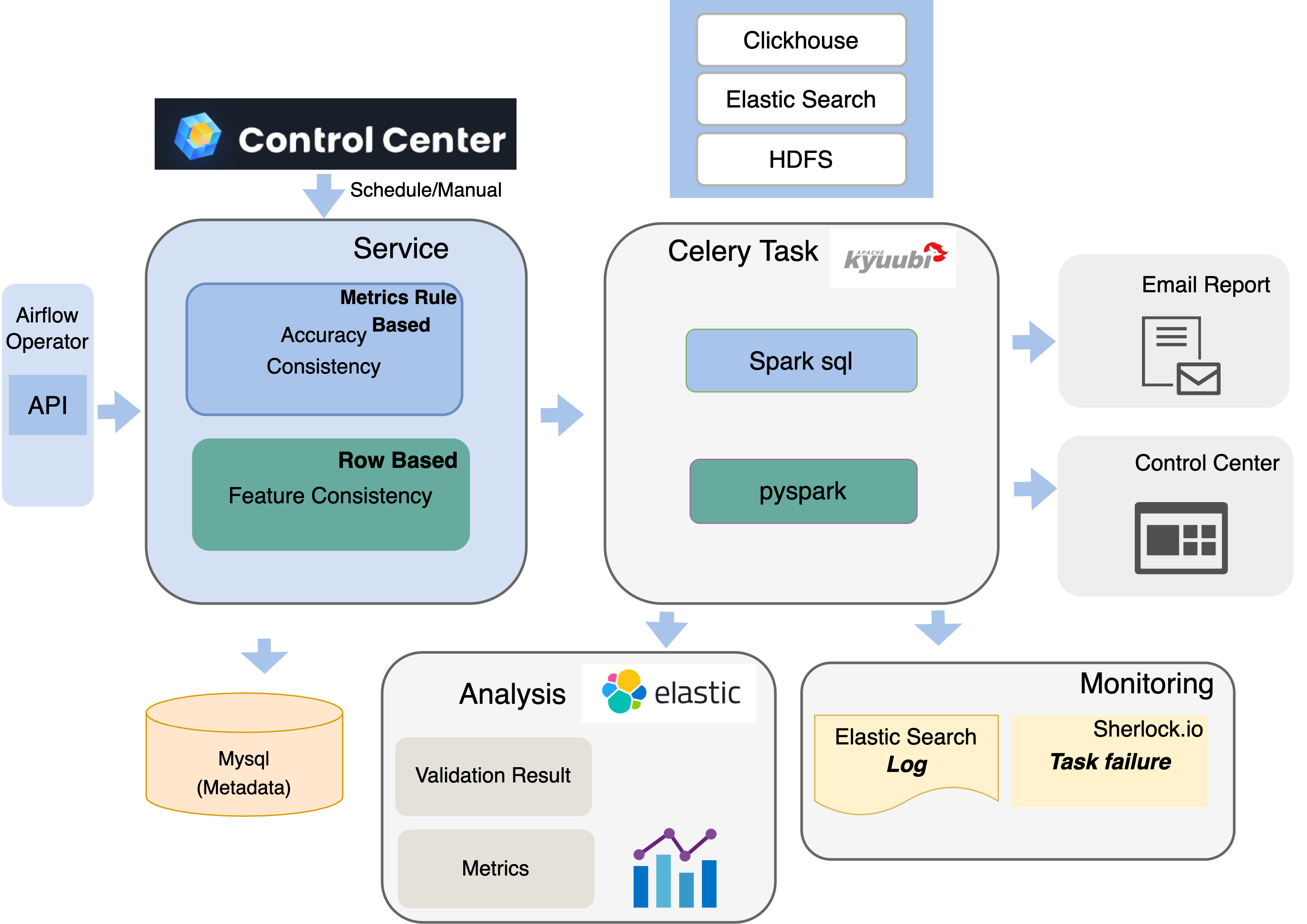
# Background

As ads is a highly data driven system, data validation is a general requirement across teams. For example:

* Validate fraud clicks count between batch and real-time
* Validate feature consistency between online and offline
* Validate data consistency among different data sets in data pipeline, like validate metrics between clickhouse and hdfs(DSS)

In order to make it more convenient and flexible, we released self-service in Control Center.

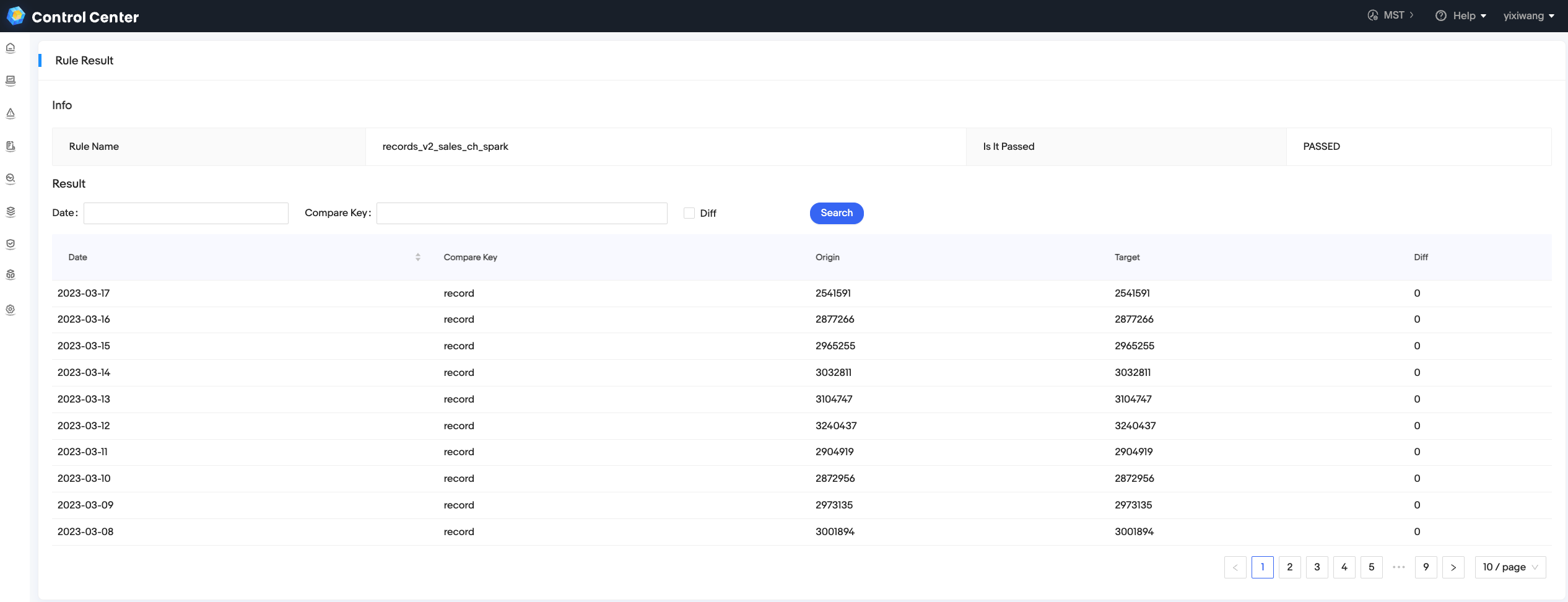
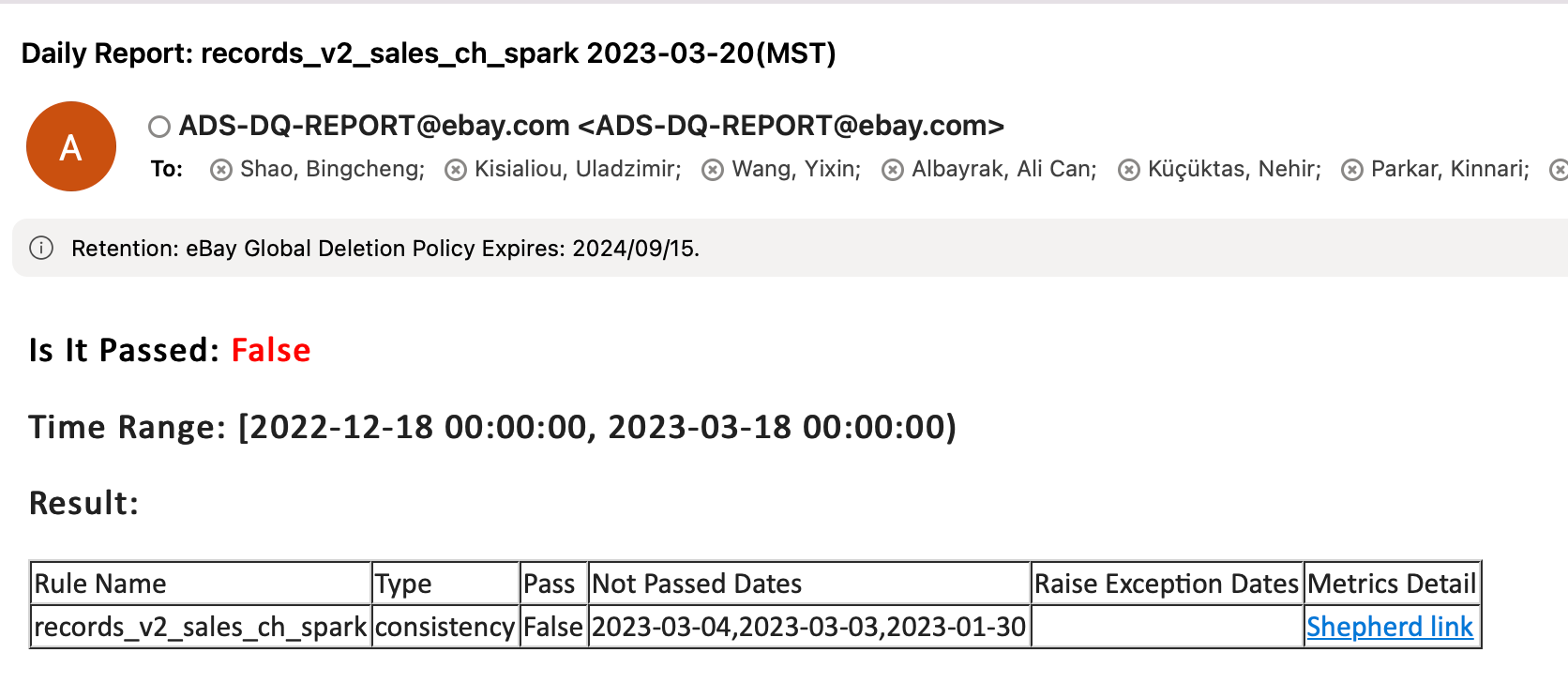
# Architecture



# Use Case

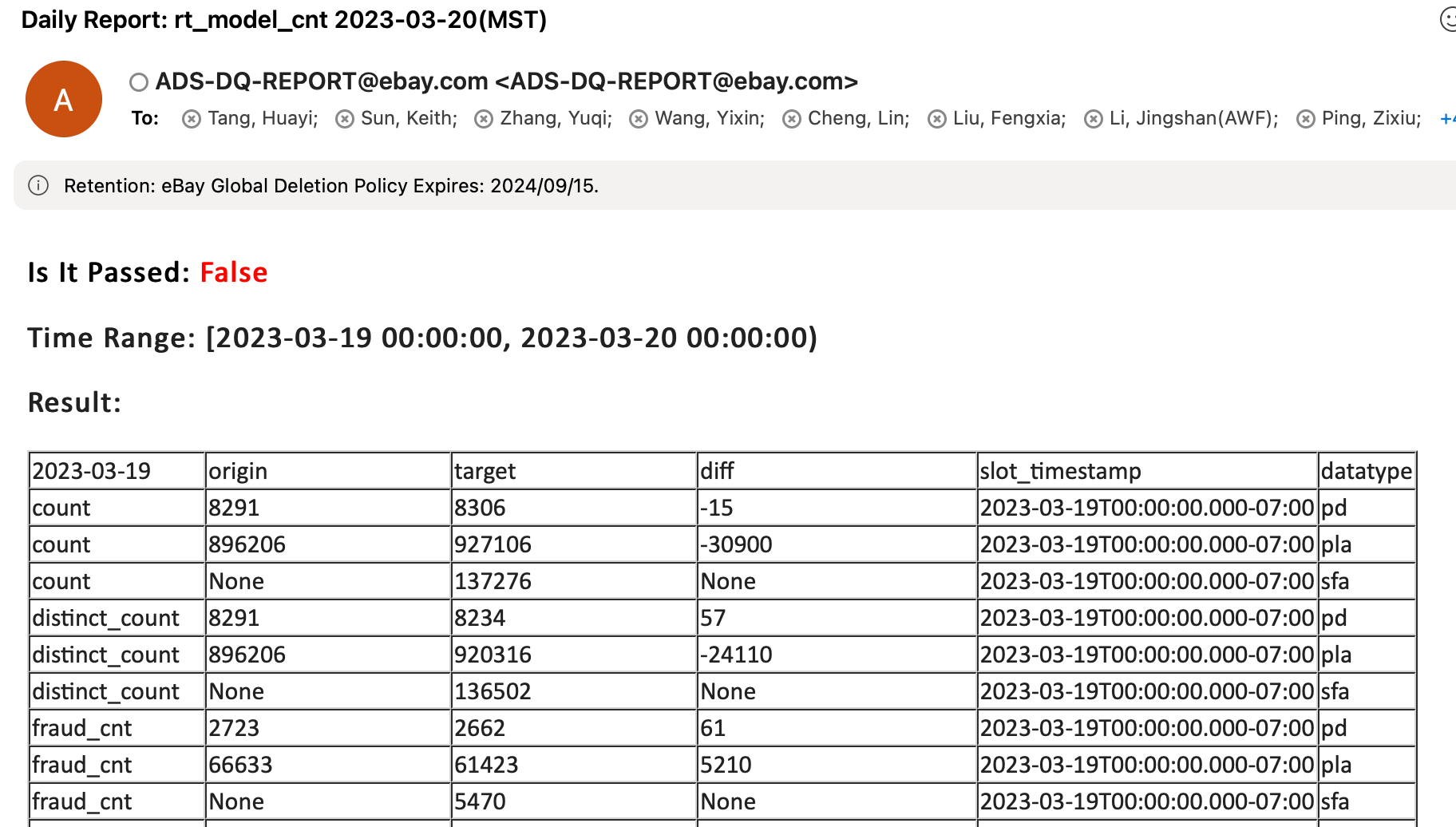
### Reporting Team - Cross Check

* Scheduled task, duration: 90 days
* Data source: Clickhouse, Hdfs



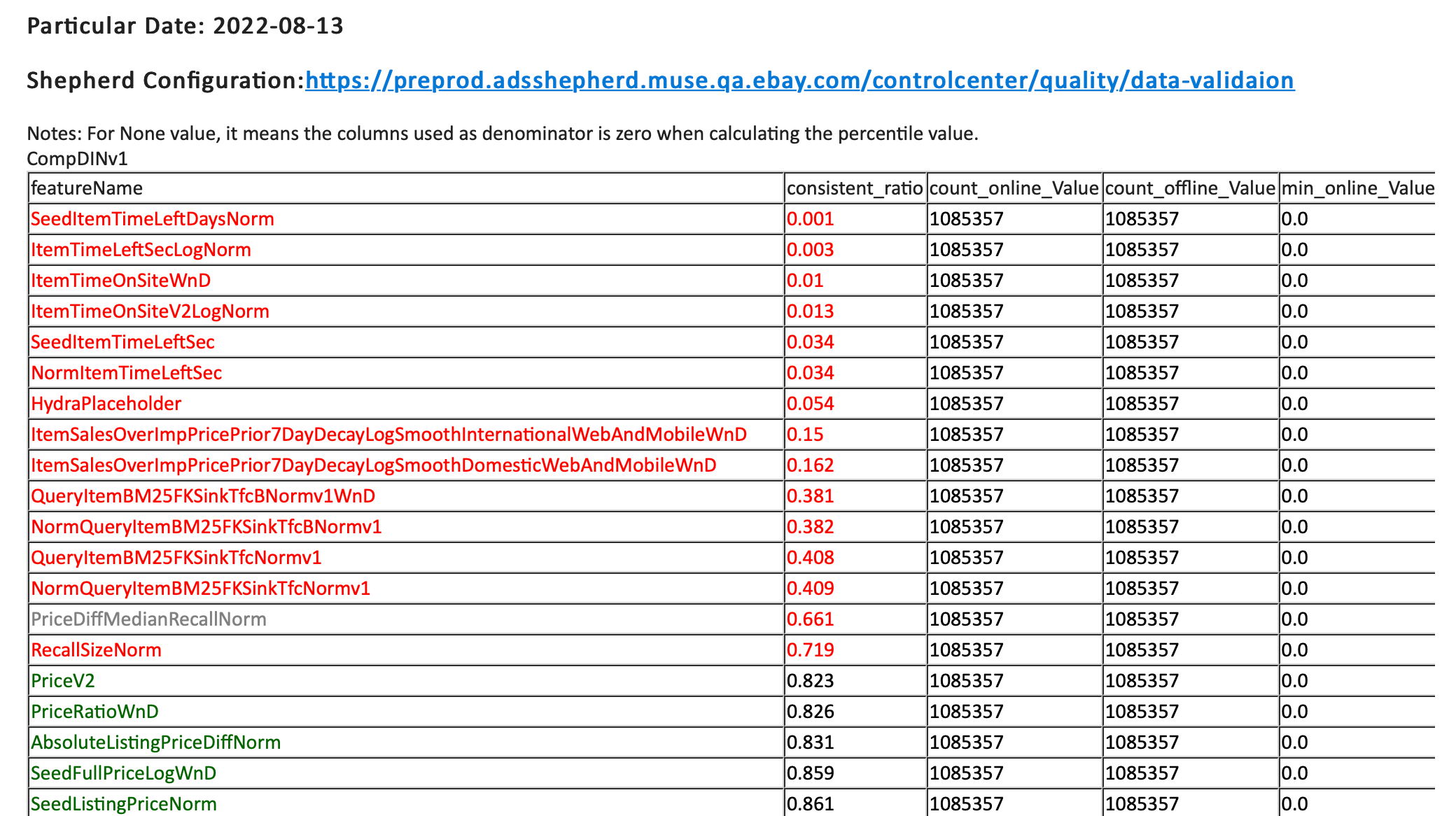
### Fraud Team - Detailed metrics report

* Scheduled task, duration: 1 day
* Data source: Hdfs



### Merch Team - feature consistency

* Integrate into Ads scheduler, use data validation as an operator
* Compare feature consistency between online and offline model, find out outliers to regulate model
* Supported metrcis:  count, min, max, diff\_percentile, diff\_ratio\_percentile, consistency\_ratio



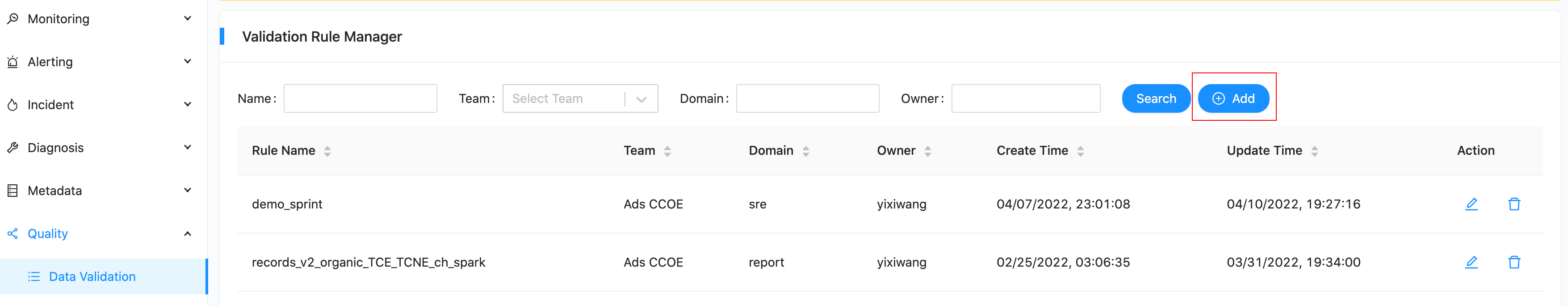
# Steps

### Step1 - Create metrics rule

[How to create a metrics rule in Control Center?](https://shepherd-api-adsshepherd.vip.ebay.com/docs/#/guide/monitoring_metrics_manager)

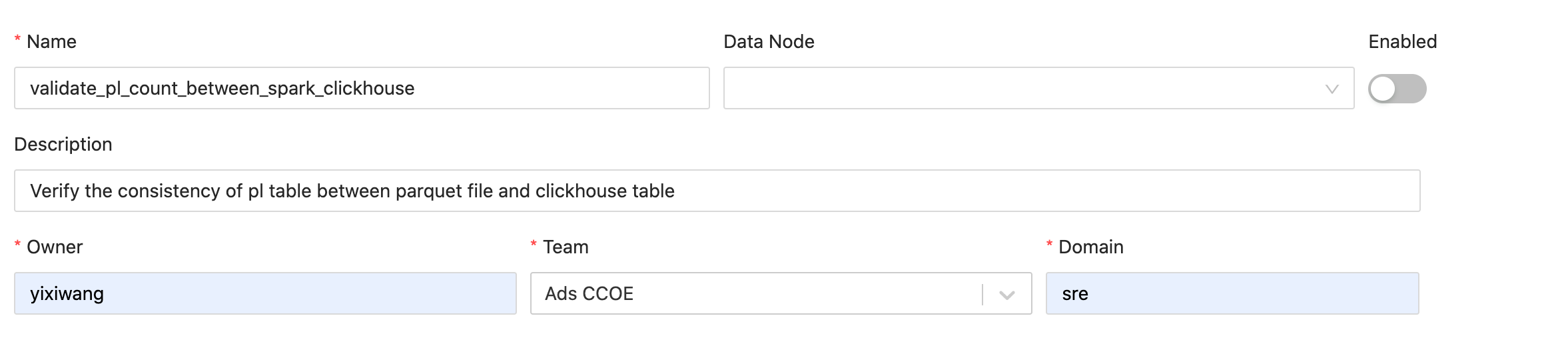
### Step2 - Create Validation rule

#### Select ****Quality - Data Validation**** in left navigation bar → Click ****Add**** button



#### Fill in basic information

            Name, Owner, Team and Domain are necessary in this part.

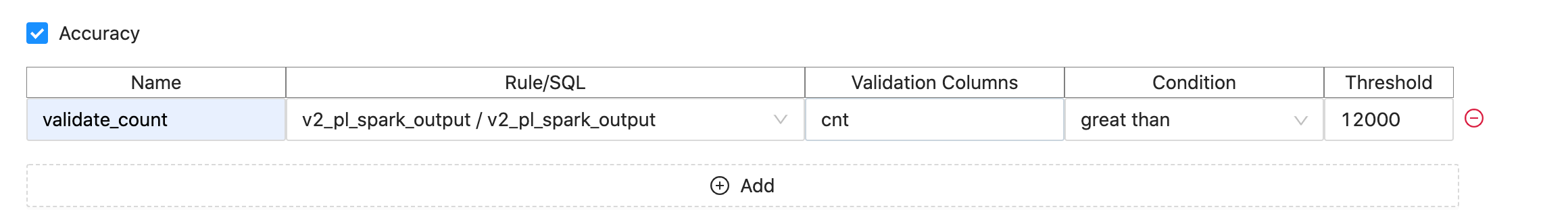


#### Add Accuracy rule

           To verify one table columns' accuracy, the rule will compare the metrics from hive/clickhouse/parquet file/es\_index which is supported by metrics rule with corresponding thresholds.

           In order to avoid duplicate rule creation, you can create multiple accuracy rules in the same validation rule.

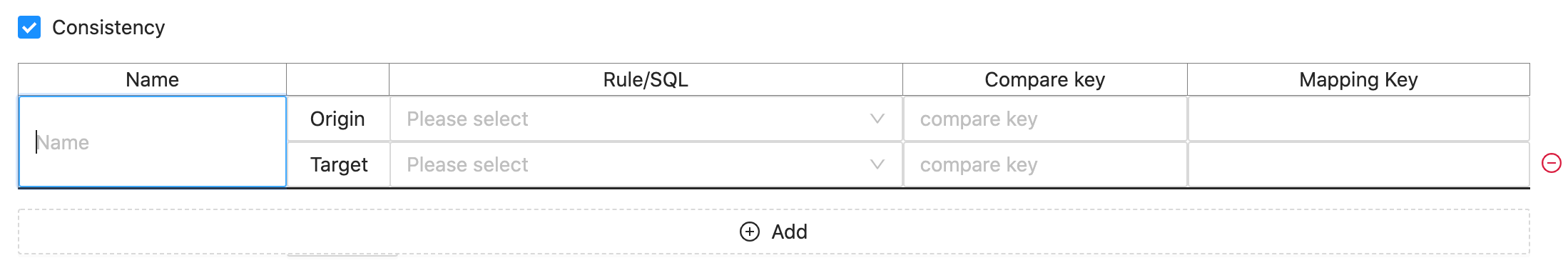
* + **Name**: The name of accuracy rule
  + **Rule/SQL**: Select the rule and sql name which you created in step 1
  + **Validation Columns**: Fill in the metric name which is selected in "Rule/SQL" stage
  + **Condition**: There are six options, equals, does not equals, less than, less than or equals, great than, great than or equals
  + **Threshold**: Threshold condition



#### Add Consistency rule

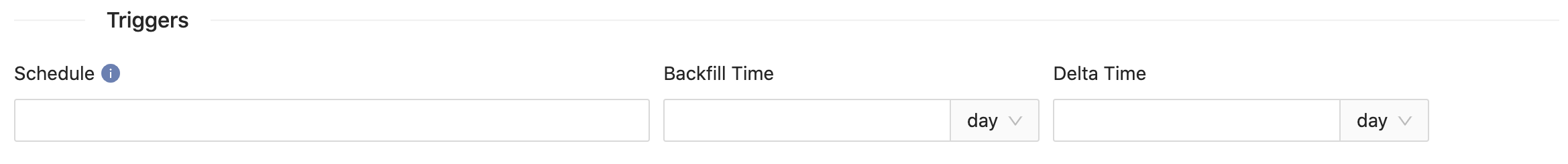
           To verify cross table column's consistency, the rule will compare source table columns with target, V1 supports clickhouse, es, hive, parquet file.

* + **Name**: the name of consistency rule
  + **Origin**: Original table
  + **Target**: Target table
  + **Rule/SQL**: The rule and sql name which you created in step 1
  + **Compare Key**: The column that need to be verified in original and target table
  + **Mapping Key**: Primary key that used to specify unique data



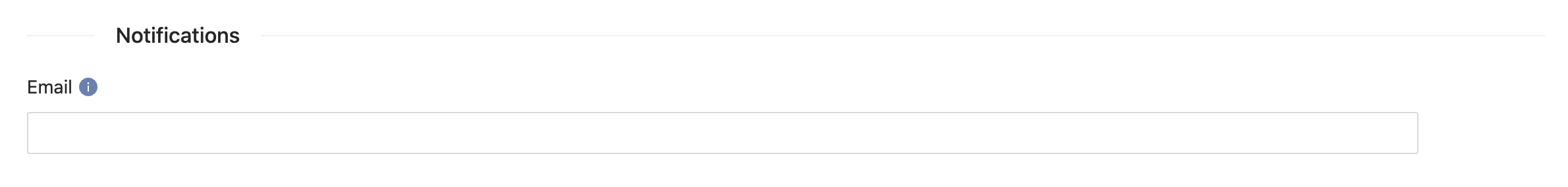
#### Add triggers parameters

* + **Schedule**: Crontab format string, [Crontab Generator](https://crontab.guru/)
  + **Backfill** **Time**: Time range
  + **Delta** **time**: The time duration from today



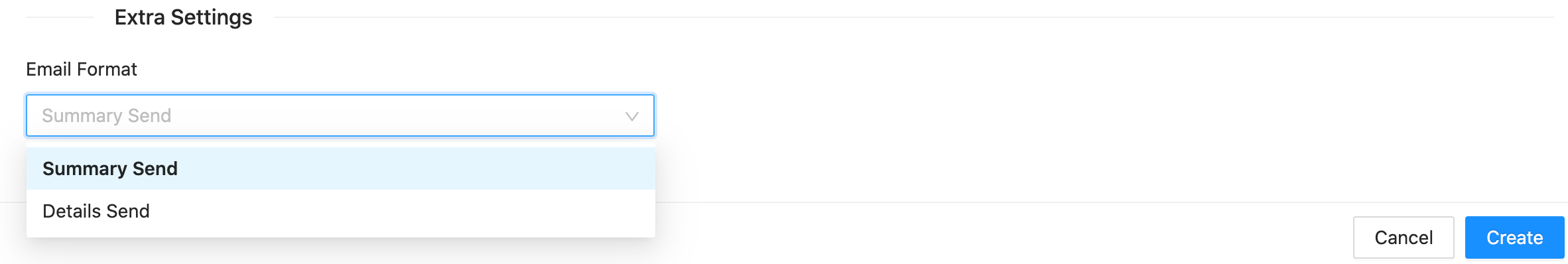
#### Add notifications

* + **Email**: Email prefix



#### Select extra settings

* + **Email** **format**: Summary send(the email only send the final result of this rule(pass/deny), does not contain metrics details), Details send(the email send the final result and the metrics details)

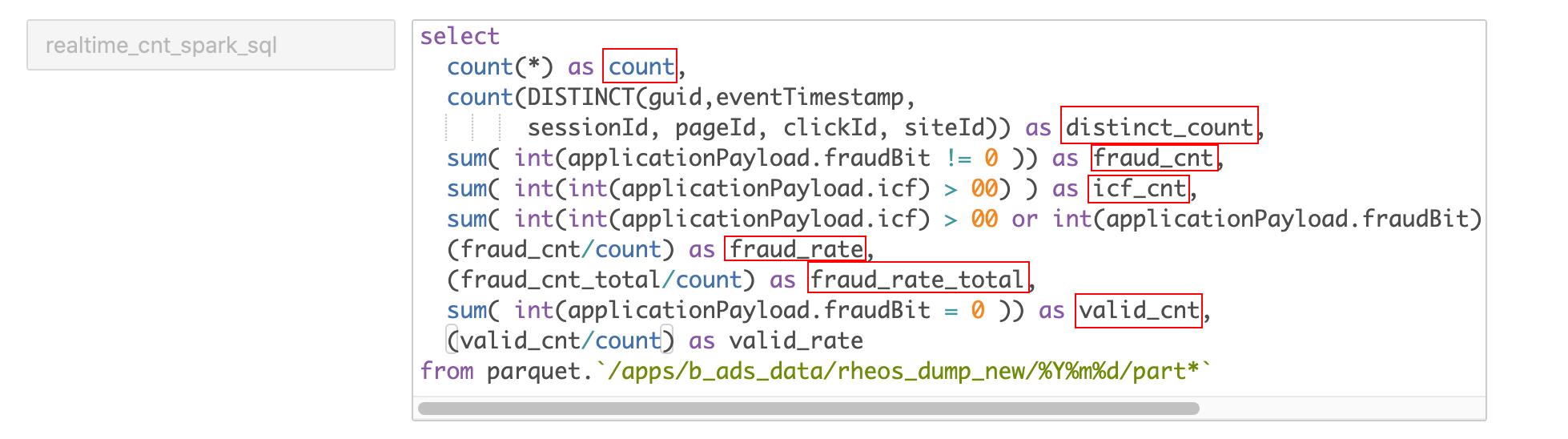


# Examples

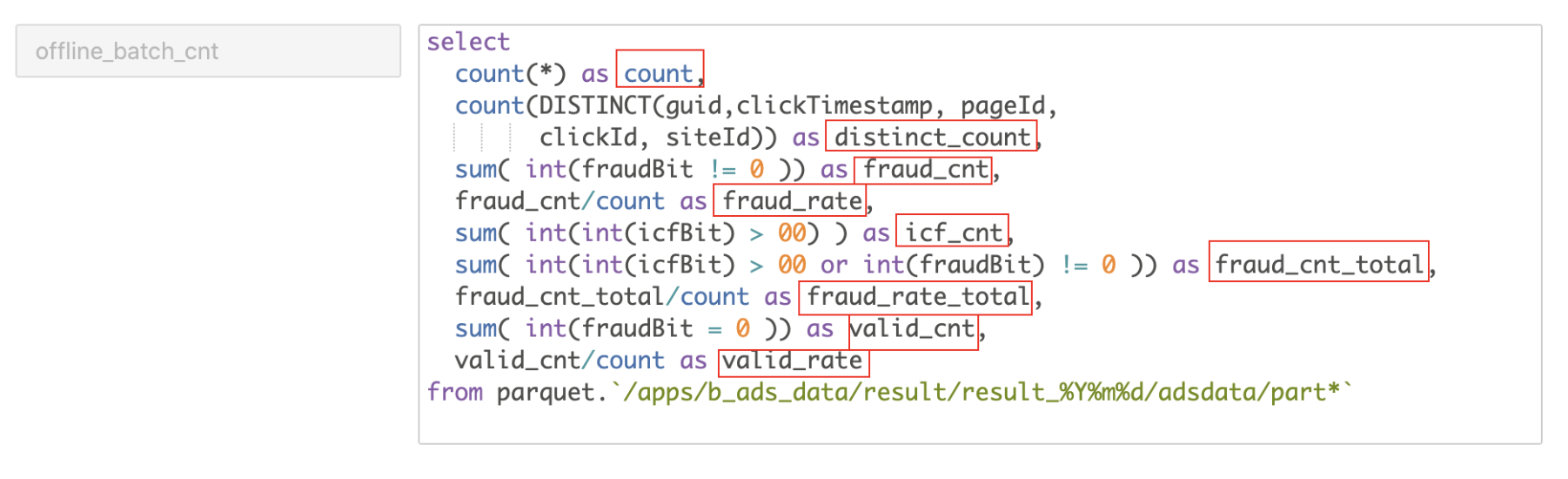
## Data Consistency

In ClickFraud team, to daily validate the model offline and real-time metrics consistency, They create multiply validation rule in Control Center. There is an example which is verifying batch\_cnt across offline and realtime table.

* Step1: Create metrics rules from offline and real-time parquet file. From the sql we can see metrics **count, distinct\_count, fraud\_cnt, icf\_cnt, fraud\_rate, fraud\_rate\_total, valid\_cnt** will be generated with a default key **slot\_timestamp** to indicate time info.



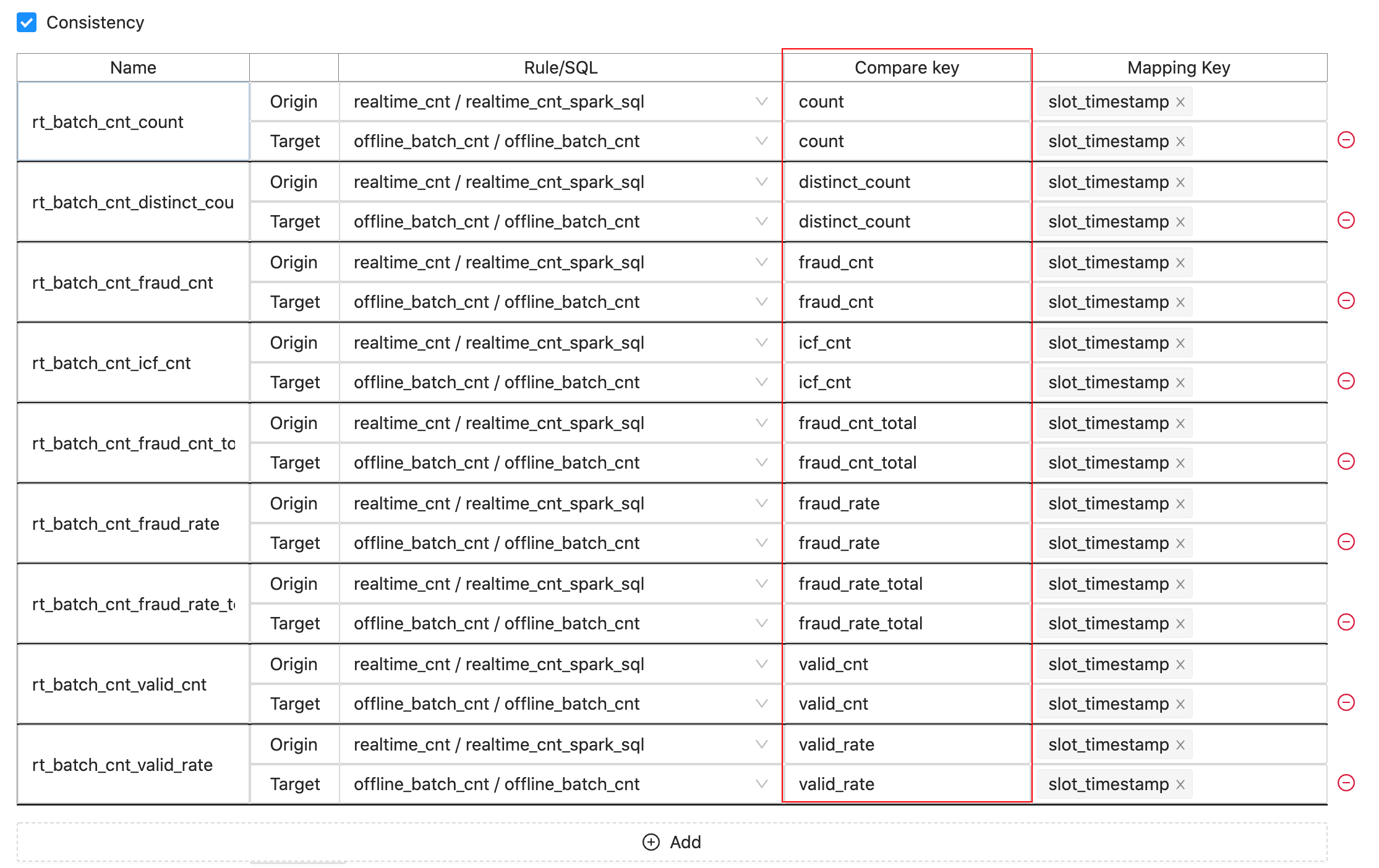
                                                 Real-time metrics rule



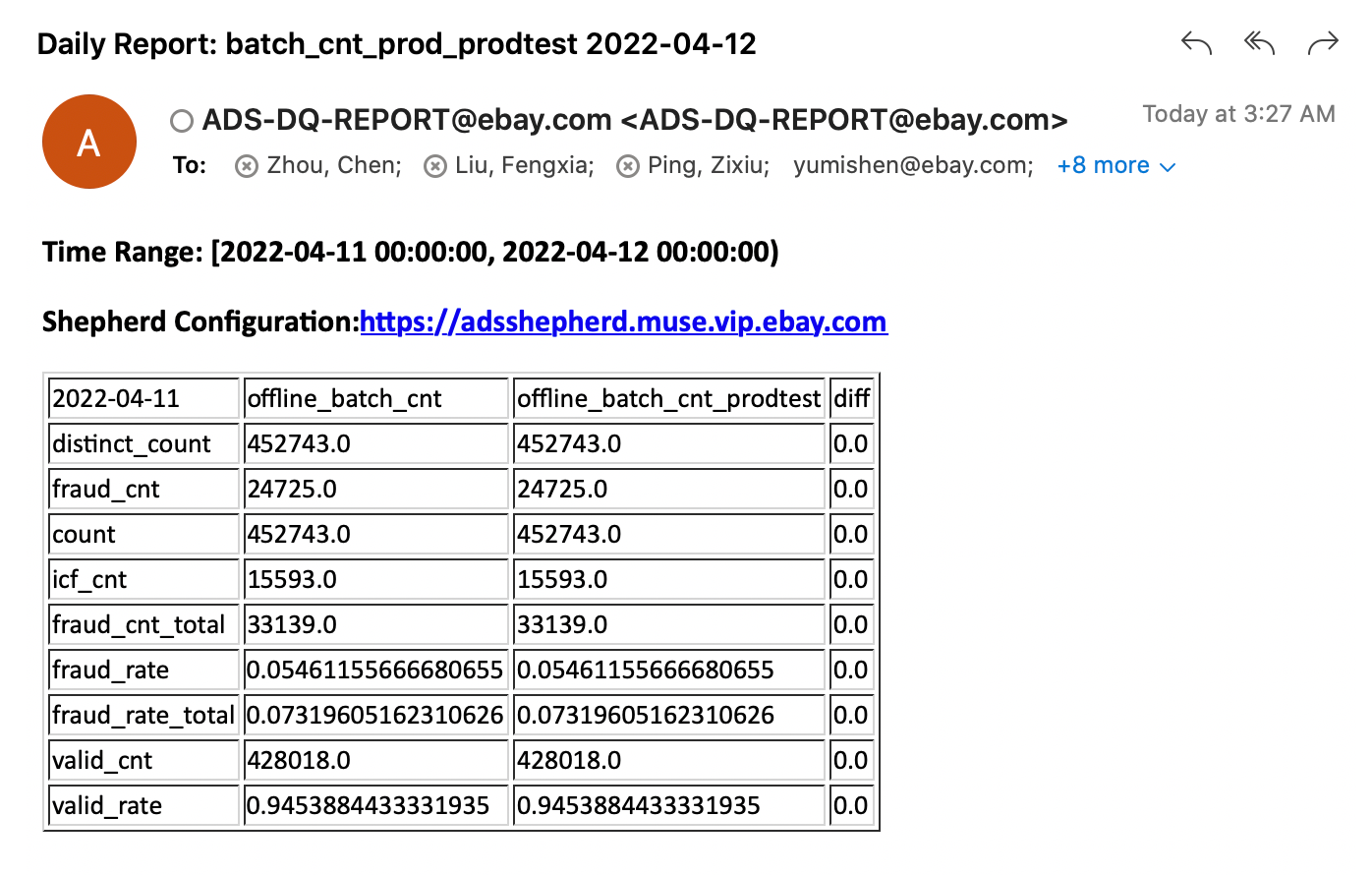
                                                 Offline metrics rule

* Step2: Create consistency validation rule.

           Take the rt\_batch\_cnt\_count rule in the below figure as an example, it will compare the count field in the realtime\_cnt/realtime\_cnt\_spark\_sql table with the count field in the offline\_batch\_cnt/offline\_batch\_cnt table.



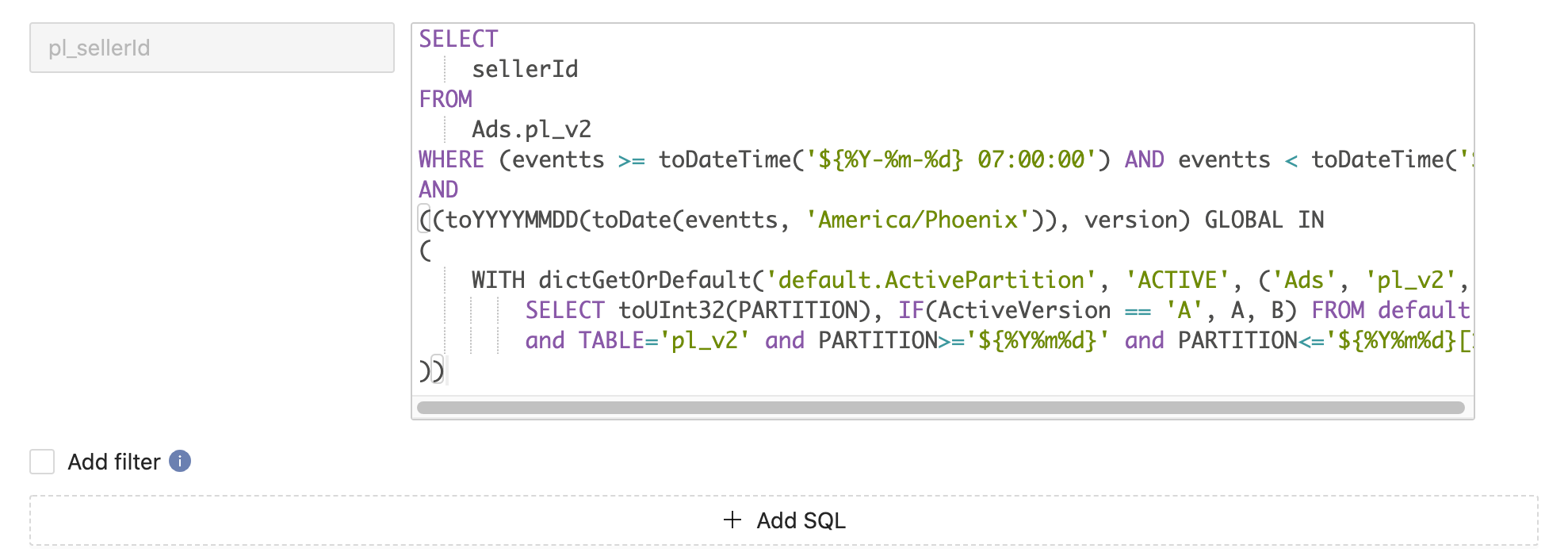
* Step3: As schedule was set up. Daily email report will be send out. You can also trigger validation by API to validate a certain date or date range.



## Data Accuracy

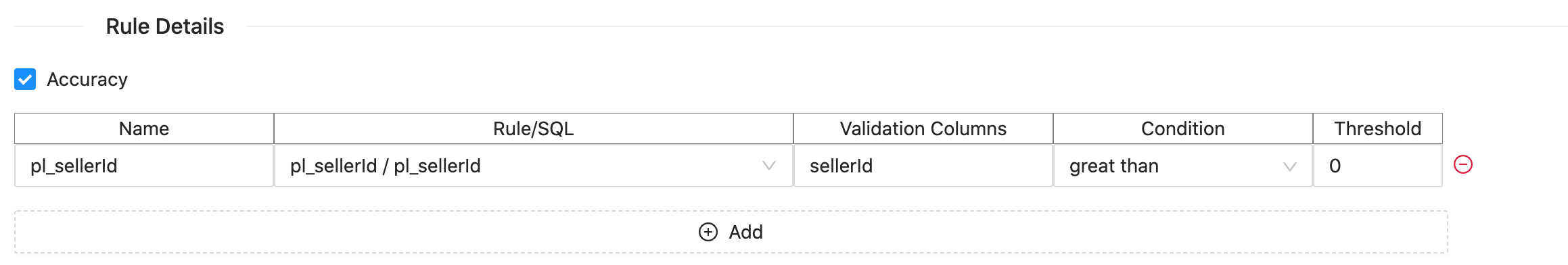
There is a case to verify the quality of sellerId. In the ebay system, it is impossible for sellerId to have a negative value.

* Step1: Create metrics rules from pl\_v2 clickhouse table. From the sql we can see metrics **sellerId** will be generated with a default key **slot\_timestamp** to indicate time info.



* Step2: Create accuracy rule

          Fill in the accuracy rule name, select the metrics rule and sql\_name which you created in step1. Fill in validation columns which is selected in "Rule/SQL" stage. Condition and threshold fields are validation conditions.



* Step3: As schedule was set up. Daily email report will be send out. You can also trigger validation by API to validate a certain date or date range.