testcase30_csv_csv_3mill50cols_content

content comparison

1. Run Summary

Application Name	:sample
Protocol Name	:DBTablesDemo
Protocol File Path	:/app/test/testprotocol/testprotocol.xlsx
Testcase Name	:testcase30_csv_csv_3mill50cols_content
Testcase Type	:content
Test Environment	:dev
Start Time	:07-Nov-2023 08:54:43 UTC
End Time	:07-Nov-2023 09:07:19 UTC
Run Time	:0:12:36
Test Result	:Passed
Reason	:Content matched

2. Configuration Details

Compare Type	:s2tcompare
testquerygenerationmode	:Auto
Testcase Type	:content
Source Connection Type	:aws-s3
Source Format	:delimitedfile
Source Path	:/app/test/data/source/ar_properties_50.csv
Source Exclude Columns	:start_date, end_date,operation_type,description
Target Connection Type	:aws-s3
Target Format	:delimitedfile
Target Path	:/app/test/data/target/ar_properties_50.csv
Target Exclude Columns	:start_date, end_date,operation_type,description
S2T Path	:/app/test/s2t/s2t_30_csv_csv_3mill50cols.xlsx
Primary Keys	:id,created_on

3. Content Summary

Test Result	:Passed
No. of matched columns	:47
No. of columns compared	:45
No. of cols in Source but not in Target	:0
No. of cols in Target but not in Source	:0
No. of rows in Source	:3,380,202
No. of distinct rows in Source	:1,803,912
No. of duplicate rows in Source	:1,576,290
No. of rows in Target	:3,380,202
No. of distinct rows in Target	:1,803,912
No. of duplicate rows in Target	:1,576,290
No. of matched rows	:1,803,912
No. of mismatched rows	:0
No. of rows in Source but not in Target	:0
No. of rows in Target but not in Source	:0

4. SQL Queries

4.1 Source Query

```
readdatadf=spark.read.format('delimitedfile').option('delimiter',,).option('header','true').load('/app/test/data/source/ar_properties_50.csv')

readdatadf.createOrReplaceTempView('dataview')

spark.sql("SELECT src.ad_type as ad_type, src.adtype1 as adtype1, src.adtype2 as adtype2, src.adtype3 as adtype3, src.adtype4 as adtype4, src.adtype5 as adtype5, src.bathrooms as bathrooms, src.baths as baths, src.bedrooms as bedrooms, src.built_up as built_up, src.carpet_area as carpet_area, src.children as children, src.created_on as created_on, src.currency as currency, src.denomination as denomination, src.dining as dining, src.discount_price as discount_price, src.end_date as end_date, src.id as id, src.l1 as l1, src.l2 as l2, src.l3 as l3, src.l4 as l4, src.l5 as l5, src.l6 as l6, src.lat as lat, src.living as living, src.lockin_period as
lockin_period, src.lon as lon, src.master as master, src.price as price, src.price_period as price_period, src.property_type as property_type, src.proptype1 as proptype1, src.proptype2 as proptype2, src.proptype3 as proptype3, src.proptype4 as proptype4, src.proptype5 as proptype5, src.rate_card as rate_card, src.retail_price as retail_price, src.rooms as rooms, src.super_built as super_built, src.surface_covered as surface_covered, src.surface_total as surface_total, src.title as title, src.walltowall as walltowall, src.washrooms as washrooms FROM dataview src  ")
```

4.2 Target Query

```
readdatadf=spark.read.format('delimitedfile').option('delimiter',,).option('header','true').load('/app/test/data/target/ar_properties_50.csv')

readdatadf.createOrReplaceTempView('dataview')

spark.sql("SELECT ad_type, adtype1, adtype2, adtype3, adtype4, adtype5, bathrooms, baths, bedrooms, built_up, carpet_area, children, created_on, currency, denomination, dining, discount_price, end_date, id, l1, l2, l3, l4, l5, l6, lat, living, lockin_period, lon, master, price, price_period, property_type, proptype1, proptype2, proptype3, proptype4, proptype5, rate_card, retail_price, rooms, super_built, surface_covered, surface_total, title, walltowall, washrooms FROM dataview tgt ")
```

5. Sample Mismatches 8 rows

5.1 Keys in source but not in target

None

5.2 Keys in target but not in source

None

5.3 Keys having one or more unequal column values

None

6. Columnwise Mismatch Summary

None

7. Columnwise Mismatch Details

None