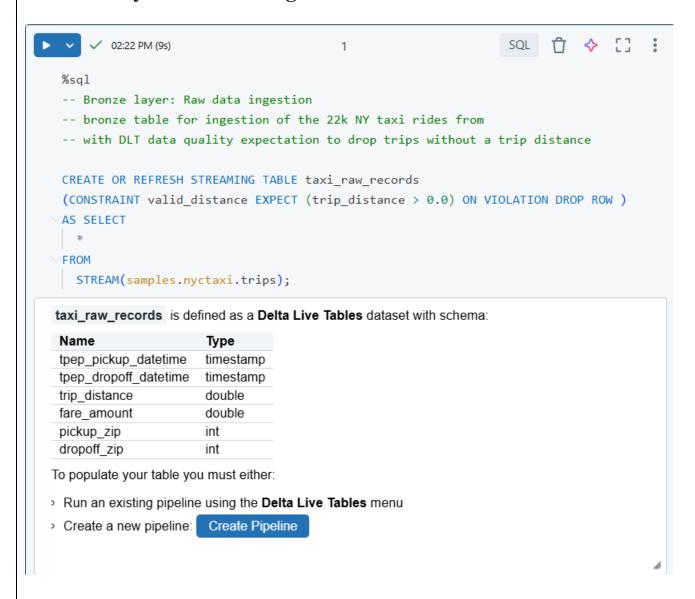
Delta Live Tables

Name: Aathirainathan P

Date: 03-12-2024

1. Bronze layer: Raw data ingestion:



2. Silver layer 1: Flagged rides:

```
SQL 🛈 ❖ [] :
    2 minutes ago (<1s)</p>
                                          2
 %sql
 -- Silver layer 1: Flagged rides
 -- silver layer: data transformations and cleansing
 -- we look into short trips or trips within the same zip code that cost more than
 $50
 CREATE OR REFRESH STREAMING TABLE flagged_rides
 AS SELECT
   date_trunc("week", tpep_pickup_datetime) as week,
   pickup_zip as zip,
   fare amount, trip distance
 FROM
   STREAM(LIVE.taxi_raw_records)
        ((pickup_zip = dropoff_zip_AND fare_amount > 50) OR
        (trip_distance < 5 AND fare amount > 50));
flagged_rides is defined as a Delta Live Tables dataset with schema:
 Name
              Type
 week
              timestamp
 zip
              int
 fare_amount double
 trip_distance double
To populate your table you must either:
> Run an existing pipeline using the Delta Live Tables menu
> Create a new pipeline: Create Pipeline
```

3. Silver layer 2: Weekly statistics

```
√ 02:22 PM (<1s)
</p>
                                        3
%sql
-- Silver layer 2: Weekly statistics
-- calculate avg fares and trip distances for each week
CREATE
OR REFRESH MATERIALIZED VIEW weekly_stats
AS SELECT
  date_trunc("week", tpep_pickup_datetime) as week,
 AVG(fare_amount) as avg_amount,
 AVG(trip_distance) as avg_distance
FROM
 live.taxi_raw_records
GROUP BY
 week
ORDER by week ASC;
```

weekly_stats is defined as a Delta Live Tables dataset with schema:

Name	Туре
week	timestamp
avg_amount	double
avg_distance	double

To populate your table you must either:

- > Run an existing pipeline using the Delta Live Tables menu
- > Create a new pipeline: Create Pipeline

4. Gold layer: Top N rides to investigate:

```
✓ 02:22 PM (<1s)</p>
                                           4
 %sql
 -- gold layer using materialized for downstream usage, e.g. BI
 -- join weely_stats with flagged_rides for top n rides to investigate
 -- display top n short distance and costly rides
 CREATE OR REPLACE MATERIALIZED VIEW top_n
 AS SELECT
   weekly_stats.week,
   ROUND(avg_amount, 2) as avg_amount,
   ROUND(avg distance, 3) as avg distance,
   fare_amount,trip_distance, zip
 FROM live.flagged_rides
 LEFT JOIN live.weekly_stats ON weekly_stats.week = flagged_rides.week
 ORDER BY fare amount DESC
 LIMIT 3;
top_n is defined as a Delta Live Tables dataset with schema:
 Name
               Type
 week
               timestamp
 avg_amount
              double
 avg_distance double
 fare_amount double
 trip_distance
              double
 zip
               int
To populate your table you must either:
> Run an existing pipeline using the Delta Live Tables menu
> Create a new pipeline: Create Pipeline
```

Submitted by: Aathirainathan P