
CREATE DELTA TABLE

```
CREATE OR REPLACE TABLE health_tracker_silver
USING DELTA
PARTITIONED BY (p_device_id)
LOCATION "/health_tracker/silver" AS (
SELECT
  value.name,
  value.heartrate,
  CAST(FROM_UNIXTIME(value.time) AS timestamp) AS time,
  CAST(FROM_UNIXTIME(value.time) AS DATE) AS dte,
  value.device_id p_device_id
FROM
  health_tracker_data_2020_01
)
```

APPEND FILES

```
INSERT INTO
  health_tracker_silver
SELECT
  value.name,
  value.heartrate,
  CAST(FROM_UNIXTIME(value.time) AS timestamp) AS time,
  CAST(FROM_UNIXTIME(value.time) AS DATE) AS dte,
  value.device_id p_device_id
FROM
  health_tracker_data_2020_02
```

PREPARE INSERTS

```
CREATE OR REPLACE TEMPORARY VIEW inserts AS (
SELECT
  value.name,
  value.heartrate,
  CAST(FROM_UNIXTIME(value.time) AS timestamp) AS time,
  CAST(FROM_UNIXTIME(value.time) AS DATE) AS dte,
  value.device_id p_device_id
FROM
  health_tracker_data_2020_02_late)
```

Prepare upserts

```
CREATE OR REPLACE TEMPORARY VIEW upserts
AS (
  SELECT * FROM updates
  UNION ALL
  SELECT * FROM inserts
)
```

Perform Upsert

```
MERGE INTO health_tracker_silver          -- the MERGE instruction is used to
perform the upsert
USING upserts
```

```
ON health_tracker_silver.time = upserts.time AND
  health_tracker_silver.p_device_id = upserts.p_device_id -- ON is used to describe the
MERGE condition
```

```
WHEN MATCHED THEN                          -- WHEN MATCHED describes the
update behavior
```

```
  UPDATE SET
    health_tracker_silver.heartrate = upserts.heartrate
```

```
WHEN NOT MATCHED THEN                      -- WHEN NOT MATCHED describes
the insert behavior
```

```
  INSERT (name, heartrate, time, dte, p_device_id)
  VALUES (name, heartrate, time, dte, p_device_id)
```

Write to Gold

```
DROP TABLE IF EXISTS health_tracker_gold;
```

```
CREATE TABLE health_tracker_gold
USING DELTA
LOCATION "/health_tracker/gold"
AS
SELECT
  AVG(heartrate) AS meanHeartrate,
  STD(heartrate) AS stdHeartrate,
```

```
MAX(heartrate) AS maxHeartrate  
FROM health_tracker_silver  
GROUP BY p_device_id
```

Optimizing

```
OPTIMIZE flights ZORDER BY (DayofWeek);
```