databricks8.4 Lab - Delta Lab

Lab 4 - Delta Lab

Module 8 Assignment

In this lab, you will continue your work on behalf of Moovio, the fitness tracker company. You will be working with a new set of files that you must move into a "gold-level" table. You will need to modify and repair records, create new columns, and merge late-arriving data.

```
%run ../Includes/Classroom-Setup
```

Mounting course-specific datasets to **/mnt/training**...
Datasets are already mounted to **/mnt/training** from **s3a://databricks-corp-training/common**

```
res1: Boolean = true
```

```
res2: Boolean = false
```

Exercise 1: Create a table

Summary: Create a table from `json` files.

Use this path to access the data:
`"dbfs:/mnt/training/healthcare/tracker/raw.json/"`

Steps to complete:

- Create a table named `health_tracker_data_2020`

- Use optional fields to indicate the path you're reading from and epress that the schema should be inferred.

```
DROP TABLE IF EXISTS health_tracker_data_2020;

CREATE TABLE health_tracker_data_2020
USING json
OPTIONS (
  path "dbfs:/mnt/training/healthcare/tracker/raw.json",
  inferSchema "true"
  );
```

OK

Exercise 2: Preview the data

Summary: View a sample of the data in the table.

Steps to complete:

- Query the table with `SELECT *` to see all columns
- Sample 5 rows from the table

```
SELECT * FROM health_tracker_data_2020
LIMIT 5
```

	month	value
1	2020-05	{"device_id": 0, "heartrate": 54.7922842229, "name": "Deborah Powell", "time"
2	2020-05	{"device_id": 0, "heartrate": 56.1916535912, "name": "Deborah Powell", "time"
3	2020-05	{"device_id": 0, "heartrate": 56.491746118, "name": "Deborah Powell", "time":
4	2020-05	{"device_id": 0, "heartrate": 55.9563823115, "name": "Deborah Powell", "time"
5	2020-05	{"device_id": 0, "heartrate": 56.1483078922, "name": "Deborah Powell", "time"

Showing all 5 rows.

Exercise 3: Count Records

Summary: Write a query to find the total number of records

Steps to complete:

- Count the number of records in the table

Answer the corresponding question in Coursera

```
SELECT COUNT(*) FROM health_tracker_data_2020
```

	count(1)
1	18168

Showing all 1 rows.

Exercise 4: Create a Silver Delta table

Summary: Create a Delta table that transforms and restructures your table

Steps to complete:

- Drop the existing `month` column
- Isolate each property of the object in the `value` column to its own column
- Cast time as timestamp **and** as a date
- Partition by `device_id`
- Use Delta to write the table

```
%fs rm -r dbfs:/user/hive/warehouse/health_tracker_data_2020_bronze
```

```
res3: Boolean = true
```

```
DROP TABLE IF EXISTS health_tracker_data_2020_bronze;
CREATE TABLE health_tracker_data_2020_bronze
USING
  delta
PARTITIONED BY
  (device_id)
SELECT
  value.device_id device_id,
  value.heartrate heartrate,
  value.name name,
  CAST(FROM_UNIXTIME(value.time) AS timestamp) AS time,
  CAST(FROM_UNIXTIME(value.time) AS DATE) AS dte
FROM
health_tracker_data_2020;
```

Query returned no results

Exercise 5: Register table to the metastore

Summary: Register your Silver table to the Metastore Steps to complete:

- Be sure you can run the cell more than once without throwing an error
- Write to the location: `/health_tracker/silver`

```
DROP TABLE IF EXISTS health_tracker_data_2020_silver;
CREATE OR REPLACE TABLE health_tracker_data_2020_silver
USING DELTA
PARTITIONED BY (p_device_id)
LOCATION "/health_tracker/silver" AS (
SELECT
  value.device_id p_device_id,
  value.heartrate heartrate,
  value.name name,
  CAST(FROM_UNIXTIME(value.time) AS timestamp) AS time,
  CAST(FROM_UNIXTIME(value.time) AS DATE) AS dte
FROM
health_tracker_data_2020
);
```

Query returned no results

Exercise 6: Check the number of records

Summary: Check to see if all devices are reporting the same number of records

Steps to complete:

- Write a query that counts the number of records for each device
- Include your partitioned device id column and the count of those records

Answer the corresponding question in Coursera

```
SELECT
*
FROM
health_tracker_data_2020_silver;
```

	p_device_id	heartrate	name	time
1	1	67.0409201203	Kristin Vasser	2020-05-01T00:00:00.000+0000
2	1	65.7129616583	Kristin Vasser	2020-05-01T01:00:00.000+0000
3	1	66.4715664581	Kristin Vasser	2020-05-01T02:00:00.000+0000
4	1	66.4433111984	Kristin Vasser	2020-05-01T03:00:00.000+0000
5	1	66.5503786953	Kristin Vasser	2020-05-01T04:00:00.000+0000
6	1	65.8812904671	Kristin Vasser	2020-05-01T05:00:00.000+0000
7	1	111.9467759173	Kristin Vasser	2020-05-01T06:00:00.000+0000

Truncated results, showing first 1000 rows.

```
SELECT
DISTINCT p_device_id,
COUNT(*) TOTAL
FROM
health_tracker_data_2020_silver
GROUP BY p_device_id;
```

	p_device_id	TOTAL
1	0	3648
2	1	3648
3	2	3648
4	3	3648
5	4	3576

Showing all 5 rows.

Exercise 7: Plot records

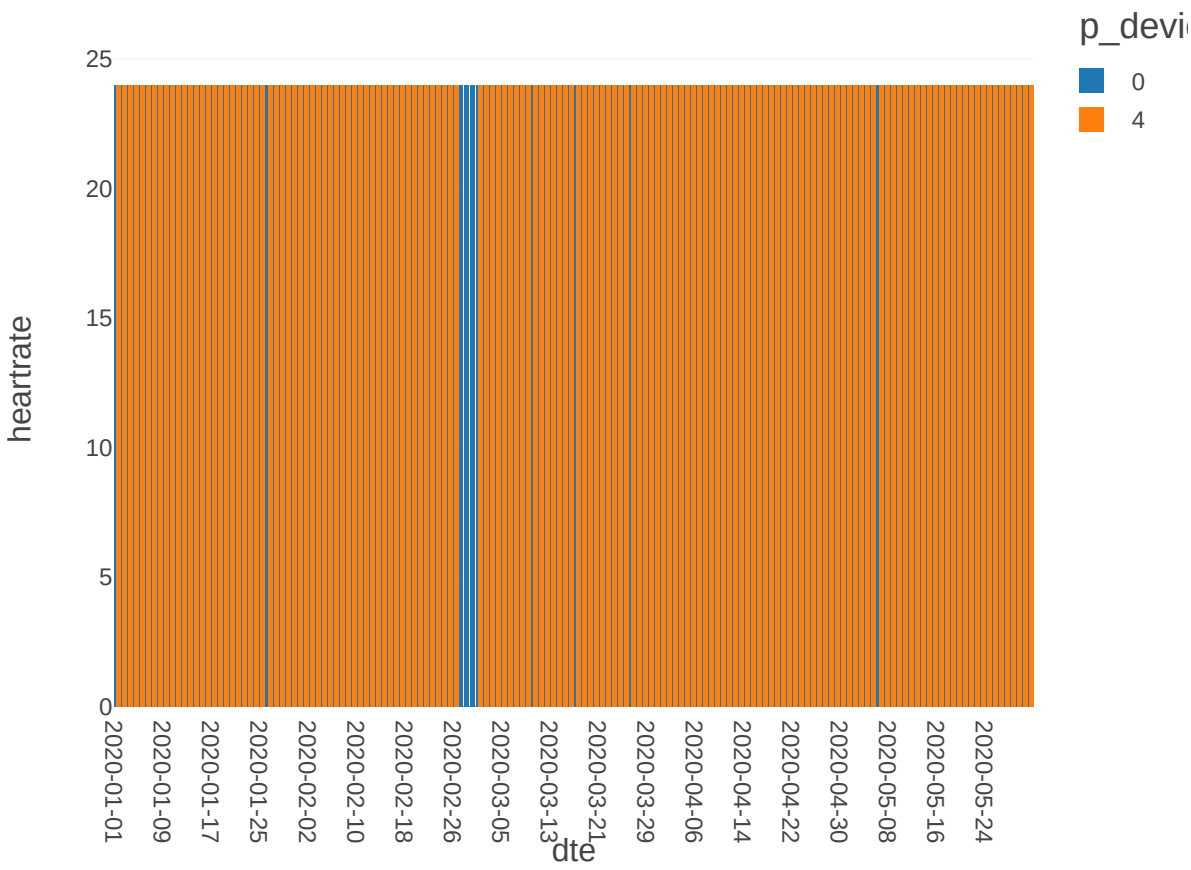
Summary: Attempt to visually assess which dates may be missing records

Steps to complete:

- Write a query that will return records from one devices that is **not** missing records as well as the device that seems to be missing records
- Plot the results to visually inspect the data
- Identify dates that are missing records

Answer the corresponding question in Coursera

```
SELECT * FROM health_tracker_data_2020_silver WHERE p_device_id IN (0,4)
```



Exercise 8: Check for Broken Readings

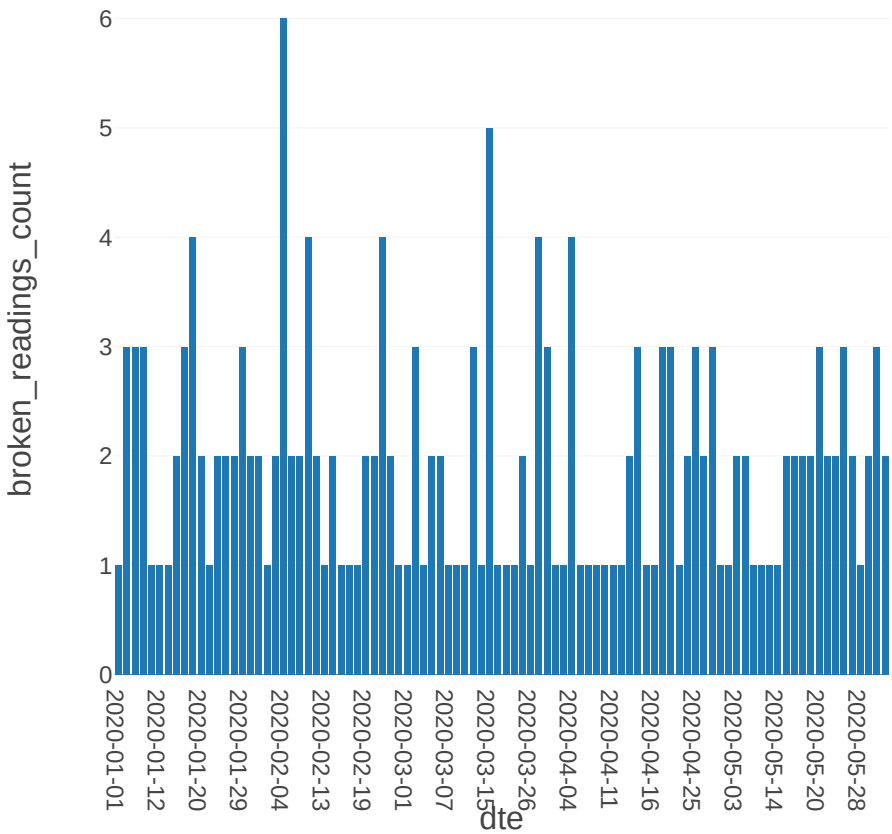
Summary: Check to see if your data contains records that would indicate a device has misreported data Steps to complete:

- Create a view that contains all records reporting a negative heartrate
- Plot/view that data to see which days include broken readings

```
CREATE OR REPLACE TEMPORARY VIEW broken_readings
AS (
  SELECT COUNT(*) as broken_readings_count, dte FROM
health_tracker_data_2020_silver
  WHERE heartrate < 0
  GROUP BY dte
  ORDER BY dte
)
```

OK

```
SELECT * FROM broken_readings;
```



Exercise 9: Repair records

Summary: Create a view that contains interpolated values for broken readings

Steps to complete:

- Create a temporary view that will hold all the records you want to update.
- Transform the data such that all broken readings (where heartrate is reported as less than zero) are interpolated as the mean of the the data points immediately surrounding the broken reading.
- After you write the view, count the number of records in it.

Answer the corresponding question in Coursera

```
describe health_tracker_data_2020_silver
```

	col_name	data_type	comment
1	p_device_id	bigint	
2	heartrate	double	
3	name	string	
4	time	timestamp	
5	dte	date	
6			
7	# Partitioning		

Showing all 8 rows.

```
CREATE OR REPLACE TEMPORARY VIEW updates
AS (
  SELECT name, (prev_amt+next_amt)/2 AS heartrate, time, dte, p_device_id
  FROM (
    SELECT *,
    LAG(heartrate) OVER (PARTITION BY p_device_id, dte ORDER BY p_device_id,
dte) AS prev_amt,
    LEAD(heartrate) OVER (PARTITION BY p_device_id, dte ORDER BY p_device_id,
dte) AS next_amt
    FROM health_tracker_data_2020_silver
  )
  WHERE heartrate < 0
)
```

OK

```
DESCRIBE updates
```

	col_name	data_type	comment
1	name	string	null
2	heartrate	double	null
3	time	timestamp	null
4	dte	date	null
5	p_device_id	bigint	null

Showing all 5 rows.

```
SELECT COUNT(*) FROM updates
```

	count(1)
1	182

Showing all 1 rows.

Exercise 10: Read late-arriving data

Summary: Read in new late-arriving data

Steps to complete:

- Create a new table that contains the late arriving data at this path: `"dbfs:/mnt/training/healthcare/tracker/raw-late.json"`
- Count the records

Answer the corresponding question in Coursera

```
DROP TABLE IF EXISTS health_tracker_data_2020_late;
CREATE TABLE health_tracker_data_2020_late
USING json
OPTIONS (
  path "dbfs:/mnt/training/healthcare/tracker/raw-late.json",
  inferSchema "true"
  );

SELECT COUNT(*) FROM health_tracker_data_2020_late;
```

	count(1)
1	72

Showing all 1 rows.

Exercise 11: Prepare inserts

Summary: Prepare your new, late-arriving data for insertion into the Silver table

Steps to complete:

- Create a temporary view that holds the new late-arriving data
- Apply transformations to the data so that the schema matches our existing Silver table

```
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_late
)
```

OK

Exercise 12: Prepare upserts

Summary: Prepare a view to upsert to our Silver table

Steps to complete:

- Create a temporary view that is the `UNION` of the views that hold data you want to insert and data you want to update
- Count the records

Answer the corresponding question in Coursera

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

OK

```
SELECT COUNT(*) FROM upserts
```

	count(1)
1	254

Showing all 1 rows.

Exercise 13: Perform upserts

Summary: Merge the upserts into your Silver table

Steps to complete:

- Merge data on the time and device id columns from your Silver table and your upserts table
- Use `MATCH` conditions to decide whether to apply an update or an insert

```
MERGE INTO health_tracker_data_2020_silver                              -- the
MERGE instruction is used to perform the upsert
USING upserts

ON health_tracker_data_2020_silver.time = upserts.time AND
   health_tracker_data_2020_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_data_2020_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)
```

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserte
1	254	182	0	72

Showing all 1 rows.

Exercise 14: Write to gold

Summary: Create a Gold level table that holds aggregated data

Steps to complete:

- Create a Gold-level Delta table
- Aggregate heartrate to display the average and standard deviation for each device.
- Count the number of records

```
DROP TABLE IF EXISTS health_tracker_data_2020_gold;

CREATE TABLE health_tracker_data_2020_gold
USING DELTA
LOCATION "/health_tracker/gold"
AS
SELECT
  AVG(heartrate) AS meanHeartrate,
  STD(heartrate) AS stdHeartrate,
  MAX(heartrate) AS maxHeartrate
FROM health_tracker_data_2020_silver
GROUP BY p_device_id
```

Query returned no results

```
SELECT * FROM health_tracker_data_2020_gold
```

	meanHeartrate	stdHeartrate	maxHeartrate
1	82.52207869094194	23.375608427849777	186.708851863
2	85.08801733820111	27.41188410264521	191.7364805027
3	84.19046229191886	24.61892380223707	192.1828472326
4	84.5435245360994	25.57932106284926	193.5343947448
5	82.7775300853638	25.54242733866919	193.5233172661

Showing all 5 rows.

Cleanup

Run the following cell to clean up your workspace.

```
%run .Includes/Classroom-Cleanup
```

```
  Notebook not found: .Includes/Classroom-Cleanup. Notebooks can be specified
   via a relative path (./Notebook or ../folder/Notebook) or via an absolute p
  ath (/Abs/Path/to/Notebook). Make sure you are specifying the path correctl
  y.

  Stacktrace:
    /SQLDA/Module-8/8.4 Lab - Delta Lab: sql
```