Google Data Analytics Capstone

Bike-share company analysis

Case Study: Cyclistic Bike-Share

Creating a database for the analysis using BigQuery SQL.

Created a table with the columns that it will be used for the analysis:

RUN SAVE SHARE SCHEDULE MORE Query completed.

```sql
CREATE TABLE cyclisticdata-22.cyclistic_bike_share.rides_data (
  ride_id string,
  rideable_type string,
  started_at timestamp,
  ended_at timestamp,
  start_station_id string,
  end_station_id string,
  member_casual string
);
```

Combined the tables from June 2021 to May 2022 to the new table:

RUN SAVE SHARE SCHEDULE MORE This query will process 409.01 MB when run.

```sql
INSERT INTO `cyclisticdata-22.cyclistic_bike_share.rides_data`(ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual)
SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
FROM (
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2021_06`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2021_07`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2021_08`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2021_09`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2021_10`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2021_11`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2021_12`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2022_01`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2022_02`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2022_03`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2022_04`
  UNION ALL
  SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
  FROM `cyclisticdata-22.cyclistic_bike_share.2022_05`
)
```

It was created as a duplicate table for backup, in BigQuery there is an option to copy the table.

PROCESS

A glimpse of the data:

RUN SAVE SHARE SCHEDULE MORE

```
--Glimpse of the data:
SELECT *
FROM `cyclisticdata-22.cyclistic_bike_share.rides_data_copy` LIMIT 10;
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	ride_id	rideable_type	started_at	ended_at	start_station_id	end_station_
1	5EF4219CD9AAA5CD	docked_bike	2021-06-27 13:19:29 UTC	2021-06-27 15:35:09 UTC	TA1309000002	TA13050000
2	F3C710D149DC2498	docked_bike	2021-10-10 01:18:18 UTC	2021-10-10 01:56:44 UTC	TA1307000117	TA13070001
3	14F0B0C18ECEBF77	docked_bike	2021-08-21 11:02:44 UTC	2021-08-21 11:47:07 UTC	LF-005	13128
4	6E4465BE8C3D6E0D	docked_bike	2021-08-13 21:38:12 UTC	2021-08-13 23:02:36 UTC	SL-005	623
5	3EB77DC17350F9A4	docked_bike	2021-08-08 23:44:13 UTC	2021-08-09 00:31:22 UTC	TA1308000029	TA13080000
6	AF93BC9ECF2AFCB0	docked_bike	2021-09-21 13:13:44 UTC	2021-09-21 14:43:44 UTC	13008	13022
7	531B7CA1EAC7AC1C	docked_bike	2021-06-04 11:30:06 UTC	2021-06-04 14:03:45 UTC	E007	13300
8	A6932DC2CC4F3650	docked_bike	2021-06-21 16:21:43 UTC	2021-06-21 17:47:41 UTC	13124	13124
9	F805359564072C9E	docked_bike	2021-09-08 13:02:32 UTC	2021-09-08 13:55:10 UTC	13022	TA13080000
10	D2C391F07864E886	docked_bike	2021-09-30 13:43:15 UTC	2021-09-30 14:35:00 UTC	13008	TA13050000

Checking for duplicates by counting distinct rows and total rows:

RUN SAVE SHARE SCHEDULE MORE

```
SELECT
(
  SELECT COUNT(1)
  FROM (SELECT DISTINCT * FROM `cyclisticdata-22.cyclistic_bike_share.rides_data_copy`
)) AS distint_rows,
(
  SELECT COUNT(1)
  FROM `cyclisticdata-22.cyclistic_bike_share.rides_data_copy`
) AS total_rows
;
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	distint_rows	total_rows
1	5489527	5773569

It can be noticed that there are less distinct rows than total rows.

Continuing checking for duplicates by counting ride_id, started_at and ended_at that are more than once:

RUN SAVE SHARE SCHEDULE MORE

```
--Checking for duplicates:
SELECT ride_id, started_at, ended_at,
COUNT(*)
 FROM `cyclisticdata-22.cyclistic_bike_share.rides_data_copy`
 GROUP BY ride_id, started_at, ended_at
 HAVING COUNT(ride_id) >1
 ;
```

Press Alt+F1 for Accessibility Optior

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	ride_id	started_at	ended_at	f0_
1	3AC0837738FFA812	2022-03-17 18:14:58 UTC	2022-03-17 18:27:03 UTC	2
2	933A28A1B4CDBC21	2022-03-01 16:13:52 UTC	2022-03-01 16:26:26 UTC	2
3	76F5EF69AFD6978C	2022-03-06 12:31:06 UTC	2022-03-06 12:43:21 UTC	2
4	A55B41F8D0F1BAE8	2022-03-15 19:09:27 UTC	2022-03-15 19:21:45 UTC	2
5	B4A752138D4A653A	2022-03-29 08:25:21 UTC	2022-03-29 08:38:02 UTC	2
6	44AEE725021E1C0F	2022-03-20 12:07:23 UTC	2022-03-20 12:19:45 UTC	2
7	8872062C3148D5CA	2022-03-09 10:44:39 UTC	2022-03-09 10:57:02 UTC	2
8	1CEFFB0D4F51C84C	2022-03-03 23:43:26 UTC	2022-03-03 23:55:54 UTC	2
9	A17548F9C249DF7C	2022-03-09 19:41:44 UTC	2022-03-09 19:53:45 UTC	2
10	DDE9A9A92FB78E8E	2022-03-03 13:20:23 UTC	2022-03-03 13:33:13 UTC	2
11	F84650ECC1DC29F9	2022-03-29 00:47:52 UTC	2022-03-29 01:00:04 UTC	2
12	92BCC9CAAACD855C	2022-03-20 12:27:33 UTC	2022-03-20 12:39:56 UTC	2
13	281AE8080D76F7B4	2022-03-20 15:55:01 UTC	2022-03-20 16:07:22 UTC	2
14	44FBE2C6E2285727	2022-03-17 16:49:38 UTC	2022-03-17 17:02:24 UTC	2
15	247D5076C1215224	2022-03-05 13:37:33 UTC	2022-03-05 13:49:34 UTC	2
16	0184F4E5AE6D6F7F	2022-03-17 14:28:01 UTC	2022-03-17 14:40:56 UTC	2

Results per page: 50 1 – 50 of 284042

It was found the duplicates have the ‘started_at’ after the ‘ended_at’. As there is a backup table, it can confidently delete the duplicates.

Deleting the ‘started_at’ after the ‘ended_at’:

RUN SAVE SHARE SCHEDULE MORE

```
DELETE
FROM `cyclisticdata-22.cyclistic_bike_share.rides_data_copy`
WHERE started_at > ended_at
;
```

Press Alt+F1 for Accessibility Opti

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS EXECUTION DETAILS

This statement removed 141 rows from rides_data_copy. GO TO TABLE

Selecting the distinct ride_id:

```
SELECT AS VALUE ANY_VALUE(t)
FROM `cyclisticdata-22.cyclistic_bike_share.rides_data_copy` AS t
GROUP BY ride_id
;
```

Query results

Row	ride_id	rideable_type	started_at	ended_at	start_station_id	end_stati
1	3B5C5DFE347BA212	electric_bike	2022-05-07 14:23:24 UTC	2022-05-07 14:32:44 UTC	13247	13157
2	F95CCA47D5E7D598	electric_bike	2021-12-03 20:33:10 UTC	2021-12-03 20:43:09 UTC	null	null
3	6AD6DB63F4DF37C9	electric_bike	2022-05-30 19:53:11 UTC	2022-05-30 20:02:50 UTC	13071	TA13070
4	31FBD6EDCC9CD5F4	electric_bike	2021-11-04 17:43:05 UTC	2021-11-04 17:53:04 UTC	TA1305000011	15534
5	280FD31929F02A35	electric_bike	2021-10-11 11:42:12 UTC	2021-10-11 11:52:01 UTC	13146	null
6	5E34A9CC9CD686B4	electric_bike	2021-08-22 22:41:20 UTC	2021-08-22 22:50:40 UTC	TA1307000001	TA13070
7	149C80864A8C885C	electric_bike	2021-08-03 11:18:05 UTC	2021-08-03 11:27:34 UTC	13430	13341
8	565C0CC022A430A5	electric_bike	2022-03-17 22:33:51 UTC	2022-03-17 22:43:12 UTC	null	null
9	3B944876DEBB2908	electric_bike	2021-10-03 14:03:01 UTC	2021-10-03 14:12:07 UTC	null	null
10	C4E391F65B6DA8B3	electric_bike	2021-11-14 10:44:50 UTC	2021-11-14 10:54:26 UTC	TA1306000012	TA13070
11	F1AD251F94EFC41A	electric_bike	2021-09-26 17:38:59 UTC	2021-09-26 17:48:21 UTC	TA1307000117	KA15030
12	C647C3CD4CB58DA7	electric_bike	2021-07-27 15:10:02 UTC	2021-07-27 15:19:47 UTC	13434	KP17050
13	AC247EF3C27612BA	electric_bike	2021-11-13 16:09:10 UTC	2021-11-13 16:18:29 UTC	TA1309000042	null
14	3655D69E19B0B366	electric_bike	2021-09-23 00:16:39 UTC	2021-09-23 00:26:29 UTC	null	TA13070
15	B009FD8617CDEA25	electric_bike	2021-06-15 08:39:07 UTC	2021-06-15 08:48:21 UTC	13146	TA13060
16	0577FAF21FA618A1	electric_bike	2021-07-29 22:51:51 UTC	2021-07-29 23:00:55 UTC	16906	16906
17	29033A837EF7CCA5	electric_bike	2022-05-29 22:58:18 UTC	2022-05-29 23:07:26 UTC	null	TA13050

Results per page: 50 1 – 50 of 5489388

Checking the number of rows can be observed that after removing the 'started_at' after 'ended_at' , the number of rows is less than the distinct number of rows counted earlier.

The result was saved as a new table data_bike_share. Also, it was observed that start_station_id and end_station_id contain NULL values.

Checking and deleting NULL values:

```
--Checking for Null and NOT NULL values:
SELECT *
FROM `cyclisticdata-22.cyclistic_bike_share.data_bike_share`
WHERE
ride_id IS NULL
OR rideable_type IS NULL
OR started_at IS NULL
OR ended_at IS NULL
OR start_station_id IS NULL
OR end_station_id IS NULL
OR member_casual IS NULL
OR ride_duration IS NULL
;
```

Query results

Row	ride_id	rideable_type	started_at	ended_at	start_station_id	end_stati
1	F95CCA47D5E7D598	electric_bike	2021-12-03 20:33:10 UTC	2021-12-03 20:43:09 UTC	null	null
2	280FD31929F02A35	electric_bike	2021-10-11 11:42:12 UTC	2021-10-11 11:52:01 UTC	13146	null
3	565C0CC022A430A5	electric_bike	2022-03-17 22:33:51 UTC	2022-03-17 22:43:12 UTC	null	null

Deleting the NULL values:

RUN SAVE SHARE SCHEDULE MORE Query completed

```
--Deleting the NULL values:
DELETE
FROM `cyclisticdata-22.cyclistic_bike_share.data_bike_share`
WHERE end_station_id IS NULL AND end_station_id IS NULL;
```

Press Alt+F1 for Accessibility Optior

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS EXECUTION DETAILS

This statement removed 803,031 rows from data_bike_share. GO TO TABLE

Add a new column with ride_duration:

RUN SAVE SHARE SCHEDULE MORE

```
--Adding ride duration:
SELECT
DATETIME_DIFF (ended_at, started_at, MINUTE) AS ride_duration
FROM `cyclisticdata-22.cyclistic_bike_share.data_bike_share`
```

Press Alt+F1 for Accessibility Opti

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	ride_duration
1	2817
2	100
3	82
4	204
5	49
6	133
7	46

Finding the day of the week and month for further analysis:

RUN SAVE SHARE SCHEDULE MORE

```
-- Checking the days of the week:

SELECT ride_id,
FORMAT_DATE('%A', started_at) AS day_of_week
FROM `cyclisticdata-22.cyclistic_bike_share.bike_share_data`
;
```

Press Alt+F1 for Accessibility Options

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	ride_id	day_of_week
1	8F1D61606CBDB963	Friday
2	1A3A82E492AB2AF5	Friday
3	835D81CE60919B67	Friday
4	936C6EB376DDCE9B	Friday

RUN SAVE SHARE SCHEDULE MORE

```
SELECT ride_id,
FORMAT_DATE('%B', started_at) AS month
FROM `cyclisticdata-22.cyclistic_bike_share.cyclist_data`
;
```

Row	ride_id	month
1	5D38DA914C5550DF	May
2	BD45846B8AA2D34F	May
3	6B2AA7CA89D05932	May
4	4E9F735FCAAE8B0A	May
5	0F388DF2BE08BE6D	May
6	1617838494FDC116	May
7	95B324D70DF6D5C9	May
8	F3AAEA1DFD5BC3F0	May
9	93FF36669A9F91AA	May
10	CB425DF72152B190	May
11	F3C0A17ACEB6F1D3	May
12	8DA325C2A0325845	May

Joining the day of the week and month to the table:

RUN SAVE SHARE SCHEDULE MORE

```
SELECT
data_cyclistic.ride_id,
data_cyclistic.started_at,
data_cyclistic.ended_at,
data_cyclistic.start_station_id,
data_cyclistic.end_station_id,
data_cyclistic.rideable_type,
data_cyclistic.ride_duration,
data_cyclistic.member_casual,
data_cyclistic.day_of_week,
month_data.month
FROM `cyclisticdata-22.cyclistic_bike_share.cyclist_data` AS data_cyclistic
JOIN `cyclisticdata-22.cyclistic_bike_share.month` AS month_data ON data_cyclistic.ride_id = month_data.ride_id
;
```

Press Alt+F1 for Accessibility Opti

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

ow	station_id	rideable_type	ride_duration	member_casual	day_of_week	month
1	:05000029	docked_bike	72	casual	Saturday	December
2	2	docked_bike	101	casual	Sunday	May
3	1	docked_bike	60	casual	Sunday	July
4	9	docked_bike	45	casual	Sunday	September

Rechecking the data:

RUN SAVE SHARE SCHEDULE MORE

```
--Rechecking the data:
SELECT
(
  SELECT COUNT(1)
  FROM (SELECT DISTINCT * FROM `cyclisticdata-22.cyclistic_bike_share.bike_share_data`
)) AS distint_rows,
(
  SELECT COUNT(1)
  FROM `cyclisticdata-22.cyclistic_bike_share.bike_share_data`
) AS total_rows
```

Press Alt+F1 for Accessibility Options

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	distint_rows	total_rows
1	4334541	4334541

It was observed that the rows are distinct.