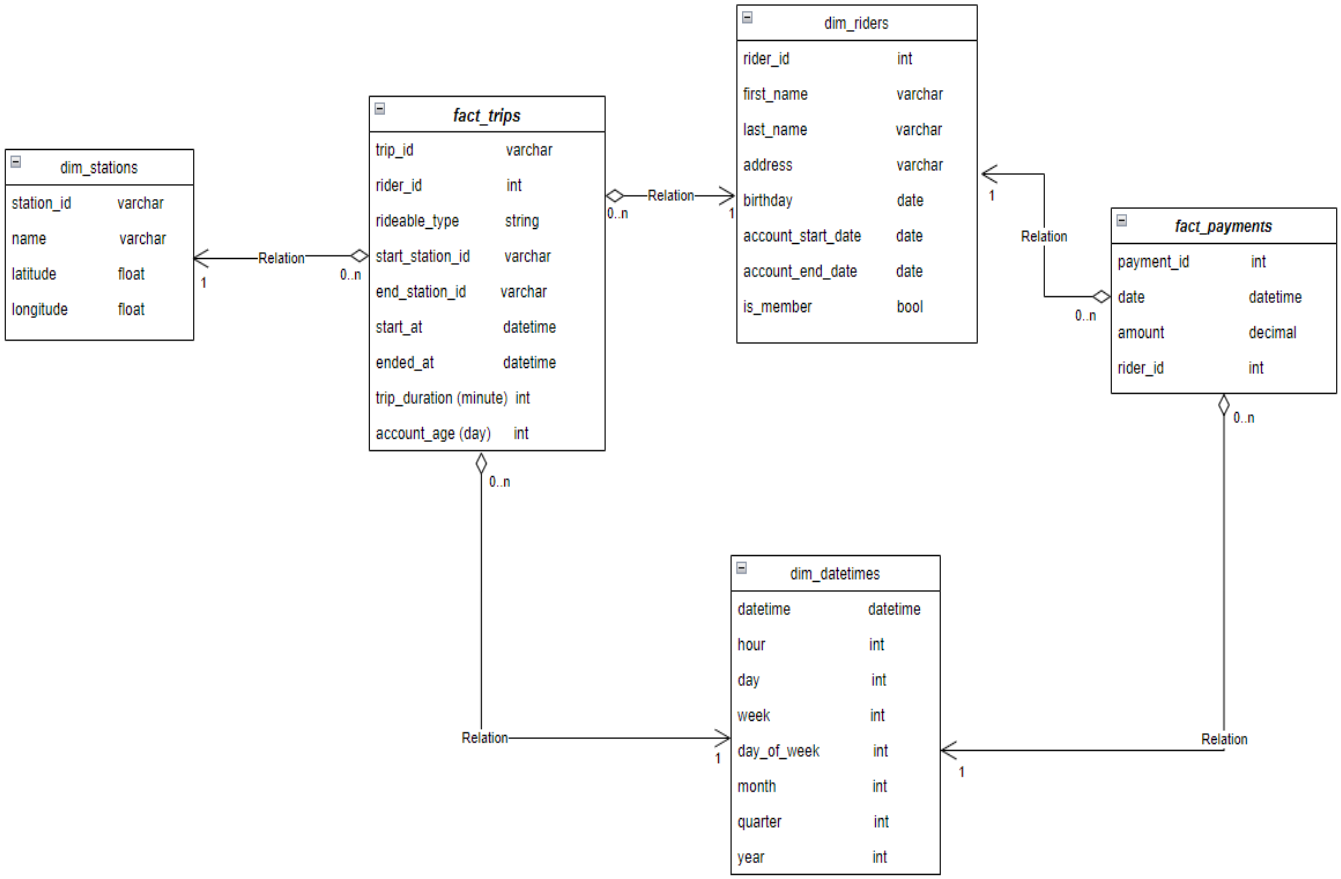
This is the Star Schema that I defined for this project:



I had 5 tables: 2 fact tables and 3 dimension tables (below is fields and data sample informations for these tables):

(See each page)

1. fact_trips: trip_id, rider_id, rideable_type, start_station_id, end_station_id, start_at, ended_at, trip_duration (minute), account_age (day).

```
display(fact_trips)
```

(2) Spark Jobs

fact_trips: pyspark.sql.dataframe.DataFrame = [trip_id: string, rider_id: integer ... 7 more fields]

Table Data Profile

	trip_id	rider_id	rideable_type	start_station_id	end_station_id	start_at	ended_at	trip_duration_minute	account_age_day
1	000002EBE159AE82	31773	electric_bike	638	13033	2021-06-22T17:25:15.000+0000	2021-06-22T17:31:34.000+0000	6	15793
2	0000080D43BAA9E4	21335	classic_bike	624	SL-008	2021-08-29T15:38:05.000+0000	2021-08-29T16:24:03.000+0000	46	8346
3	00000CAE95438C9D	71748	classic_bike	13022	TA1305000003	2021-07-20T15:40:46.000+0000	2021-07-20T17:38:17.000+0000	118	8096
4	00000EBBC119168C	44951	classic_bike	KA1503000069	TA1309000037	2021-10-31T11:30:37.000+0000	2021-10-31T11:39:27.000+0000	9	9099
5	000019B7F053D461	66662	classic_bike	13193	TA1309000033	2021-08-13T19:57:28.000+0000	2021-08-13T20:02:56.000+0000	5	5331
6	00001A81D056B01B	14172	classic_bike	13432	TA1306000015	2021-04-14T08:10:11.000+0000	2021-04-14T08:19:14.000+0000	9	9111
7	00001B4F79D102B5	43153	classic_bike	TA1309000049	13325	2021-07-28T07:58:27.000+0000	2021-07-28T08:05:00.000+0000	7	10742

Truncated results, showing first 1000 rows.

Click to re-execute with maximum result limits.

Command took 4.51 seconds -- by student_10f0d7tol6mgb9b9_00826808@vocareumvocareum.onmicrosoft.com at 01:42:16, 12/8/2022 on demo

2. fact_payments: payment_id, date, amount, rider_id.

```
display(payments)
```

(1) Spark Jobs

payments: pyspark.sql.dataframe.DataFrame = [payment_id: integer, date: timestamp ... 2 more fields]

Table Data Profile

	payment_id	date	amount	rider_id
1	1	2019-05-01T00:00:00.000+0000	9	1000
2	2	2019-06-01T00:00:00.000+0000	9	1000
3	3	2019-07-01T00:00:00.000+0000	9	1000
4	4	2019-08-01T00:00:00.000+0000	9	1000
5	5	2019-09-01T00:00:00.000+0000	9	1000
6	6	2019-10-01T00:00:00.000+0000	9	1000
7	7	2019-11-01T00:00:00.000+0000	9	1000

Truncated results, showing first 1000 rows.

Click to re-execute with maximum result limits.

Command took 0.57 seconds -- by student_10f0d7tol6mgb9b9_00826808@vocareumvocareum.onmicrosoft.com at 01:42:16, 12/8/2022 on demo

3. dim_riders: rider_id, first_name, last_name, address, birthday, account_start_date, account_end_date, is_member.

```
2 display(riders)
```

(1) Spark Jobs

riders: pyspark.sql.dataframe.DataFrame = [rider_id: integer, first_name: string ... 6 more fields]

Table Data Profile

	rider_id	first_name	last_name	address	birthday	account_start_date	account_end_date	is_member
1	1000	Diana	Clark	1200 Alyssa Squares	1989-02-13T00:00:00.000+0000	2019-04-23T00:00:00.000+0000	null	true
2	1001	Jennifer	Smith	397 Diana Ferry	1976-08-10T00:00:00.000+0000	2019-11-01T00:00:00.000+0000	2020-09-01T00:00:00.000+0000	true
3	1002	Karen	Smith	644 Brittany Row Apt. 097	1998-08-10T00:00:00.000+0000	2022-02-04T00:00:00.000+0000	null	true
4	1003	Bryan	Roberts	996 Dickerson Turnpike	1999-03-29T00:00:00.000+0000	2019-08-26T00:00:00.000+0000	null	false
5	1004	Jesse	Middleton	7009 Nathan Expressway	1969-04-11T00:00:00.000+0000	2019-09-14T00:00:00.000+0000	null	true
6	1005	Christine	Rodriguez	224 Washington Mills Apt. 467	1974-08-27T00:00:00.000+0000	2020-03-24T00:00:00.000+0000	null	false
7	1006	Alicia	Taylor	1137 Angela Locks	2004-01-30T00:00:00.000+0000	2020-11-27T00:00:00.000+0000	2021-12-01T00:00:00.000+0000	true

Truncated results, showing first 1000 rows.

Click to re-execute with maximum result limits.

Command took 0.39 seconds -- by student_10f0d7tol6mgb9b9_0082608@vocareumvocareum.onmicrosoft.com at 01:42:16, 12/8/2022 on demo

4. dim_stations: station_id, name, latitude, longitude.

```
2 display(stations)
```

(1) Spark Jobs

stations: pyspark.sql.dataframe.DataFrame = [station_id: string, name: string ... 2 more fields]

Table Data Profile

	station_id	name	latitude	longitude
1	525	Glenwood Ave & Touhy Ave	42.012701	-87.66605799999999
2	KA1503000012	Clark St & Lake St	41.88579466666667	-87.63110066666668
3	637	Wood St & Chicago Ave	41.895634	-87.672069
4	13216	State St & 33rd St	41.8347335	-87.6258275
5	18003	Fairbanks St & Superior St	41.89580766666667	-87.62025316666669
6	KP1705001026	LaSalle Dr & Huron St	41.894877	-87.632326
7	13253	Lincoln Ave & Waveland Ave	41.948797	-87.675278

Showing all 838 rows.

Command took 0.40 seconds -- by student_10f0d7tol6mgb9b9_0082608@vocareumvocareum.onmicrosoft.com at 01:42:16, 12/8/2022 on demo

5. dim_datetimes: datetime, hour, day, week, weekday (day_of_week), month, quarter, year.

```
15 display(dim_datetimes)
```

(4) Spark Jobs

time_df: pyspark.sql.dataframe.DataFrame = [value: timestamp]

dim_datetimes: pyspark.sql.dataframe.DataFrame = [datetime: timestamp, hour: integer ... 6 more fields]

Table Data Profile

	datetime	hour	day	week	day_of_week	month	quarter	year
1	2021-02-12T18:36:17.000+0000	18	12	6	6	2	1	2021
2	2021-02-27T10:35:38.000+0000	10	27	8	7	2	1	2021
3	2021-02-12T07:41:13.000+0000	7	12	6	6	2	1	2021
4	2021-02-27T14:33:11.000+0000	14	27	8	7	2	1	2021
5	2021-02-23T11:22:10.000+0000	11	23	8	3	2	1	2021
6	2021-02-11T11:42:52.000+0000	11	11	6	5	2	1	2021
7	2021-02-28T11:47:54.000+0000	11	28	8	1	2	1	2021

Truncated results, showing first 1000 rows.

Click to re-execute with maximum result limits.

Command took 1.97 minutes -- by student_10f0d7tol6mgb9b9_00826808@vocareumvocareum.onmicrosoft.com at 01:57:24, 12/8/2022 on demo