Querying data using SQL

- New York Taxi dataset
- Yellow Taxi Trip Records and Taxi Zone Lookup Table.

Create Database

```
%sql
CREATE DATABASE taxidata
```

```
%sql
USE taxidata
```

Data Preview

```
%sql

SELECT *
FROM yellow_tripdata
LIMIT 10;
```

```
%sql

SELECT VendorID, Trip_Distance
FROM yellow_tripdata
LIMIT 10;
```

Filtering data

```
%sql
SELECT *
FROM taxi_zone_lookup
WHERE borough = 'Queens';
```

```
%sql
SELECT *
FROM yellow_tripdata
WHERE VendorID IN (1,2);
```

Filtering data (cont.)

```
%sql
SELECT *
FROM taxi_zone_lookup
WHERE borough LIKE '%een%';
```

```
%sql
SELECT*
FROM taxi_zone_lookup
WHERE borough LIKE 'M%nha%n';
```

Filtering data (cont.)

```
%sql
SELECT *
FROM yellow_tripdata
WHERE vendorid BETWEEN 1 AND 5;
```

```
%sql
SELECT *
FROM yellow_tripdata
WHERE tpep_pickup_datetime BETWEEN
'2021-01-02' AND '2021-01-03';
```

Subqueries

```
%sql
SELECT *
FROM yellow_tripdata
WHERE pulocationid IN
  (SELECT locationid FROM taxi_zone_lookup WHERE zone = 'Flatlands');
```

```
%sql
SELECT *
FROM yellow_tripdata
WHERE trip_distance NOT BETWEEN 0 AND 10;
```

Joins and merges

```
%sql

SELECT
tz.Borough,
tz.Zone,
yt.tpep_pickup_datetime,
yt.tpep_dropoff_datetime
FROM
yellow_tripdata yt
LEFT JOIN taxi_zone_lookup tz
ON (yt.PULocationID = tz.LocationID);
```

Order

```
%sql
SELECT *
FROM taxi_zone_lookup
ORDER BY borough, zone;
```

```
%sql
SELECT *
FROM taxi_zone_lookup
ORDER BY borough DESC, zone ASC;
```

Functions

```
%sql
SELECT ROUND(SUM(trip_distance),2) AS rounded_trip_distance
FROM yellow_tripdata_2021_01_csv;
```

```
%sql
SELECT VendorID, SUM(fare_amount) total_amount
FROM yellow_tripdata
GROUP BY VendorID ORDER BY total_amount;
```

```
%sql
select md5('Pablito')
```

Windowing Functions

```
%sql

CREATE TABLE taxi_day_sum AS
SELECT dayofmonth(tpep_pickup_datetime) day, passenger_count, sum(fare_amount) total_fare_amount
FROM yellow_tripdata
GROUP BY dayofmonth(tpep_pickup_datetime), passenger_count;
```

Windowing Functions: Breakdown by day

```
%sql
SELECT
day,
passenger_count,
total_fare_amount,
round(sum(total_fare_amount) OVER (PARTITION BY day),2) day_total,
round(total_fare_amount/sum(total_fare_amount) OVER (PARTITION BY day) * 100,2) day_pct
FROM taxi_day_sum
where day = 1
ORDER BY day, passenger_count;
```

Views

```
%sql

CREATE VIEW borough_timespan_view AS
SELECT
tz.Borough,
tz.Zone,
yt.tpep_pickup_datetime,
yt.tpep_dropoff_datetime
FROM
yellow_tripdata yt
LEFT JOIN taxi_zone_lookup tz
ON (yt.PULocationID = tz.LocationID);
```

- Views are only **stored queries**, not materialized tables.

Delta Lake SQL

- Bridge between data lakes and traditional database territory.
- More data manipulation options with SQL (`UPDATE`, `DELETE`, `MERGE`).
- Tables need to be explicity created as Delta Lake tables.

Delta Lake SQL (cont.)

```
%sql
%sql
CREATE TABLE tzl_delta USING DELTA AS
SELECT LocationID, Borough, Zone,
service_zone FROM taxi_zone_lookup;
```

```
%sql
UPDATE tzl_delta SET zone = 'Unknown' WHERE locationid = 265;
DELETE FROM tzl_delta WHERE locationid = 265;
```

Delta Lake SQL (cont.)

```
%sql
DESCRIBE HISTORY tzl_delta;
```

```
%sql
SELECT * FROM tzl_delta VERSION AS OF 1
MINUS SELECT * FROM tzl_delta
VERSION AS OF 0;
```

Delta Lake SQL (cont.)

```
%sql
OPTIMIZE tzl_delta;
VACUUM tzl_delta RETAIN 200 HOURS;
```

Accessing metadata

```
%sql
DESCRIBE taxi_zone_lookup;
DESC detail taxi_zone_lookup;
```

```
%sql
SHOW DATABASES;
SHOW TABLES;
SHOW COLUMNS FROM tzl_delta;
```

```
%sql
SHOW ALL FUNCTIONS;
SHOW SYSTEM FUNCTIONS LIKE '*SU*';
SHOW USER FUNCTIONS LIKE '*TAX*';
```

Statistics and `EXPLAIN` plan

```
%sql
ANALYZE TABLE yellow_tripdata COMPUTE STATISTICS;

ANALYZE TABLE yellow_tripdata COMPUTE STATISTICS FOR COLUMNS tpep_pickup_datetime,tpep_dropoff_datetime, PULocationID DOLocationID
```

```
%sql
EXPLAIN SELECT * FROM yellow_tripdata;
```