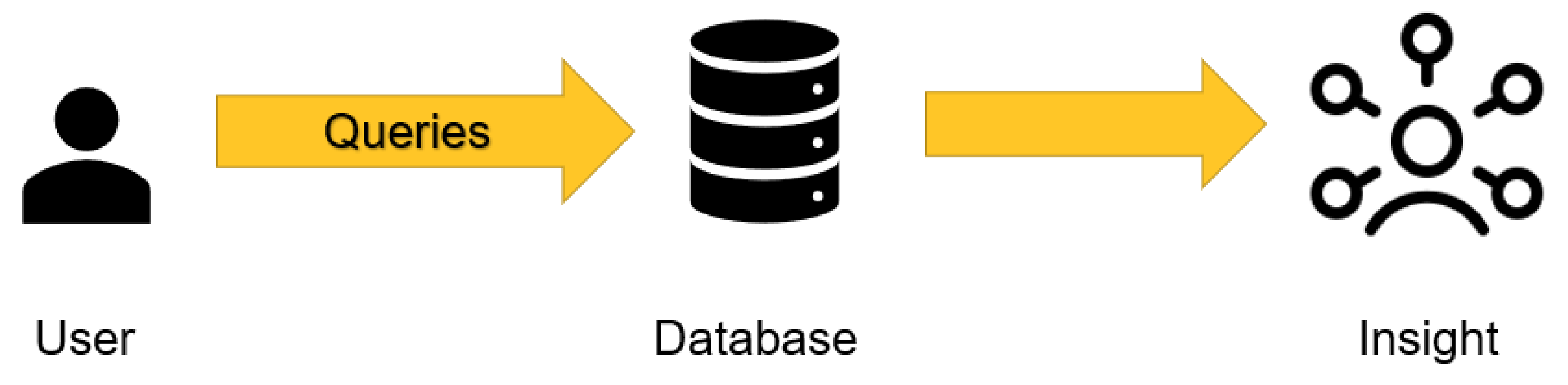
Auto-generation of Spatial SQL Queries

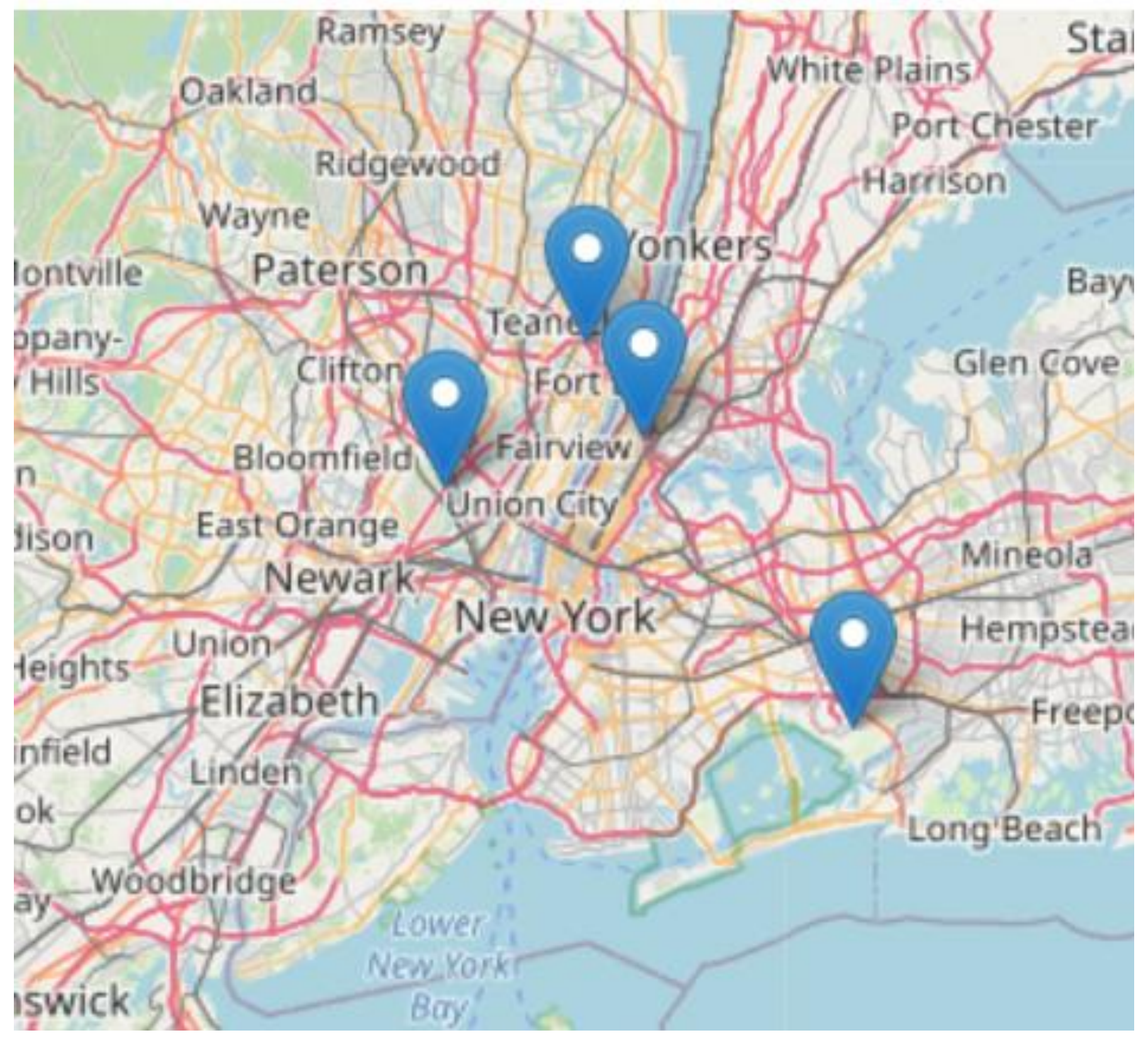
Krishna Prasad Sheshadri, Computer Science
Mentor: Professor Mohamed Sarwat Abdelghany Aly Elsayed; Collaborator: Vamsi Meduri
CIDSE, Arizona State University

Research question:
The objective of this research is to develop creative ways of generating spatial SQL queries .We implement generation of two varieties of spatial queries for two different applications namely next query prediction and cardinality estimation.

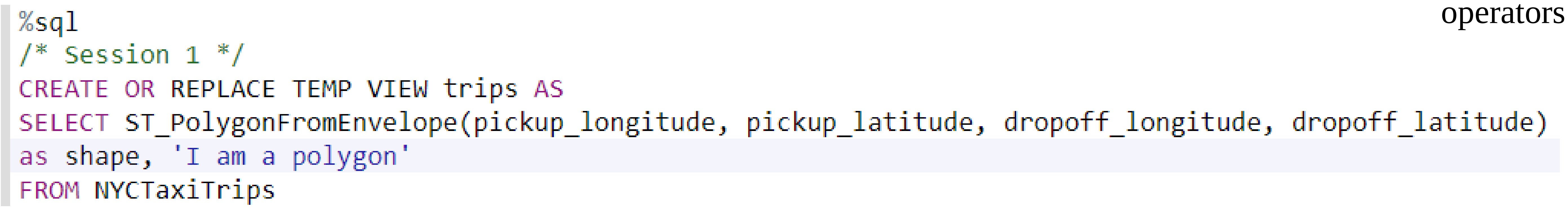
1.Next query prediction:
An Interactive Data Exploration session primarily consists of a user interacting with a dataset to draw interesting insights/ patterns by issuing a sequence of queries.



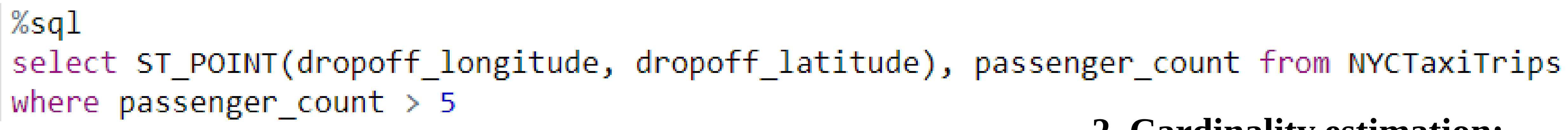
Human intent is where the queries become progressively more complex and converge towards a specific insight.
For example, a session of spatial queries issued by the user may look like this:

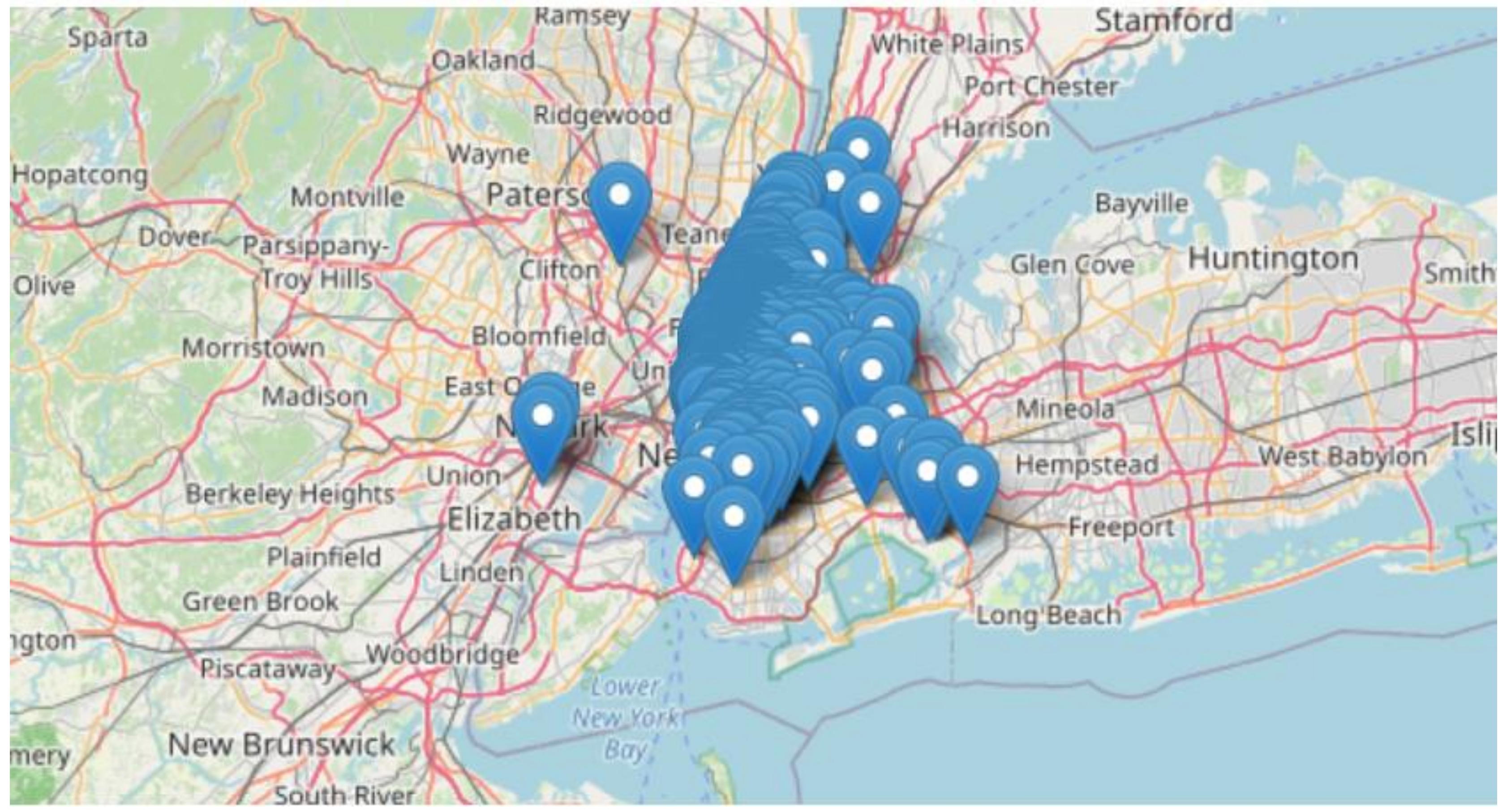


QUERY 1:

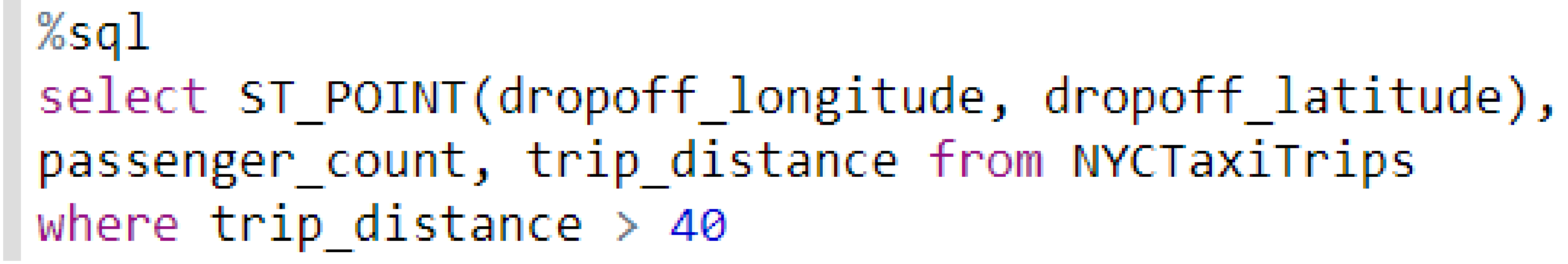

```
%sql
/* Session 1 */
CREATE OR REPLACE TEMP VIEW trips AS
SELECT ST_PolygonFromEnvelope(pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude)
as shape, 'I am a polygon'
FROM NYCTaxiTrips
```

QUERY 2


```
%sql
select ST_POINT(dropoff_longitude, dropoff_latitude), passenger_count from NYCTaxiTrips
where passenger_count > 5
```



QUERY 3


```
%sql
select ST_POINT(dropoff_longitude, dropoff_latitude),
passenger_count, trip_distance from NYCTaxiTrips
where trip_distance > 40
```

Spatial predicate operators: ST_Contains

ST_Intersects

ST_Within

ST_Equals

ST_Crosses

ST_Touches

ST_Overlaps

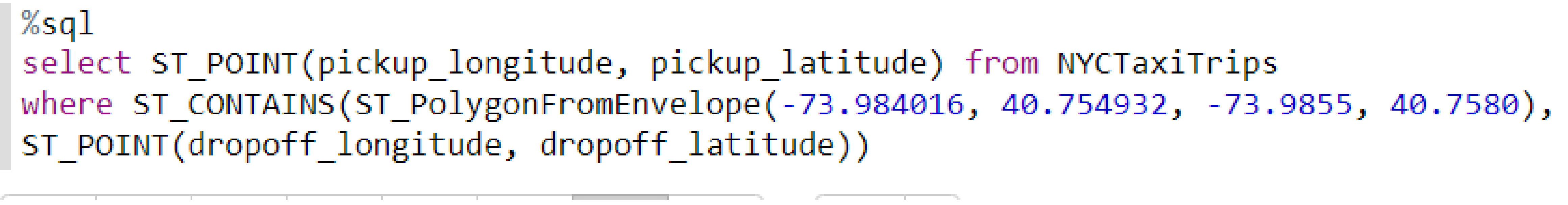
2. Cardinality estimation:
Cardinality estimation has been an active topic of interest in recent years and among the useful and important types of metadata has been the number of distinct values in a column. We shall generate several independent spatial queries with each query having a different purpose.

There are three different types of queries we synthesize :

1.Proximity query: The number of points that lie within a certain X miles from a specific latitude, longitude point. Spatial predicates such as KNN JOIN are used.

2.Point-polygon Spatial overlap query: The number of points that overlap with a given polygon. Spatial predicates such as ST_Contains, ST_Overlaps are used.

3.Polygon-polygon Spatial overlap query : The number of polygons that overlap with a given polygon. Spatial predicates such as ST_Intersects, ST_Contains are used.


```
%sql
select ST_POINT(pickup_longitude, pickup_latitude) from NYCTaxiTrips
where ST_CONTAINS(ST_PolygonFromEnvelope(-73.984016, 40.754932, -73.9855, 40.7580),
ST_POINT(dropoff_longitude, dropoff_latitude))
```

References:
1. Venkata Vamsikrishna Meduri, Kanchan Chowdhury, and Mohamed Sarwat. 2021. Evaluation of Machine Learning Algorithms in Predicting the Next SQL Query from the Future. ACM Trans. Database Syst. 46, 1, Article 4 (March 2021), 46 pages. https://doi.org/10.1145/3442338
2.Venkata Vamsikrishna Meduri, Kanchan Chowdhury, Mohamed Sarwat: Recurrent Neural Networks for Dynamic User Intent Prediction in Human-Database Interaction. EDBT 2019: 654-657
3. Hazar Harmouch and Felix Naumann. Cardinality Estimation: An Experimental Survey. PVLDB, 11(4): 499 - 512, 2017. DOI: https://doi.org/10.1145/3164135.3164145
4. DataFrame/SQL. (n.d.). Retrieved April 13, 2021, from http://sedona.apache.org/api/viz/sql/

FURI

Ira A. Fulton Schools of Engineering
Arizona State University