

Processamento de Streams

Análise a Corridas de Táxis

André Lopes *Departamento de Informática*
Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
 Almada, Portugal

Nelson Coquenim *Departamento de Informática*
Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
 Almada, Portugal

I. INTRODUÇÃO

II. Frequent Routes

O objectivo deste *query* é achar o top 10 das rotas mais frequentes durante um período de 30 minutos. Um rota é representada por uma *cell* inicial e uma *cell* final.

Na Figura 1 observa-se a estrutura desta *query*. Primeiramente efetua-se uma janela deslizante de 30 minutos sobre o input. De seguida, selecciona-se os 10 resultados com a frequência mais alta.

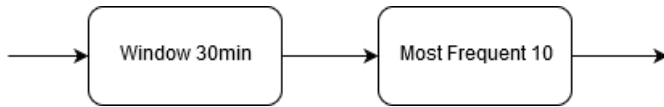


Fig. 1. Diagrama da *query* Frequent Routes.

O código siddhi que implementa esta *query* é o seguinte:

```
from TaxiSecStr#window.time(30 minutes)
select pickup_grid_x, pickup_grid_y,
       dropoff_grid_x, dropoff_grid_y,
       count(*) as frequency
group by pickup_grid_x, pickup_grid_y,
         dropoff_grid_x, dropoff_grid_y
order by frequency DESC
insert into RouteFrequencyStr;

from RouteFrequencyStr#window.length(10)
select *
insert into TopFreqRoutesStr;
```

III. Profitables Areas

Nesta *query* pretende-se identificar, de forma contínua, as áreas que são mais lucrativas para os taxistas. Para tal, o lucro de uma área é definido pelas receitas geradas nessa área a dividir pelo número de táxis vazios também nessa área.

A receita gerada numa área é a média das *fare + tip* de todas as corridas que originaram nessa área e que acabaram nos 15 minutos seguintes.

O número de táxis vazios é a soma dos táxis que efetuaram uma *dropoff* nessa área mas que após 30 minutos ainda não efetuaram uma *pickup*.

Na Figura 2 pode-se observar um diagrama que demonstra o fluxo desta *query*.

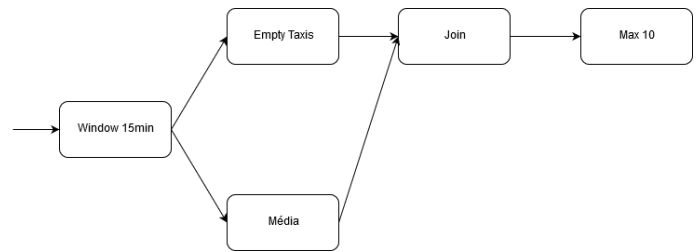


Fig. 2. Diagrama da *query* Profitables Areas.

O código siddhi que implementa esta *query* é o seguinte:

```
from TaxiSecStr#window.time(15 min)
select avg(FareTrip)
group by pickup_grid_x, pickup_grid_y,
         dropoff_grid_x, dropoff_grid_y
insert into ProfitStr;
```

```
verificar esta query no paper esta diferente
from e1 = TaxiSecStr ->
    TaxiSecStr[e1.medallion == medallion
    and pickup_datetime - e1.dropoff_datetime > 30]
select *
insert into EmptyTaxisStr
```

IV. IDLE TAXIS

```
from TaxiSecStr#window.time(1 hour)
select *
insert into AvailableTaxisStr;
```

```
from e1 = TaxiSecStr ->
    e2 = TaxiSecStr[e2.medallion == e1.medallion]
select e1.medallion as taxi,
       (e2.picked_datetime
        - e1.dropoff_datetime) as idle_time
insert into IdleTimeTaxisStr;
```

```
// idle_time is in seconds
```

```

from IdleTimeTaxisStr#window.time(1 hour) select taxi, avg(idle_time)
insert into IdleTaxisStr;

```

V. CONGESTED AREAS

```

from every e1 = TaxiSecStr,
    e2 = TaxiSecStr[e2.medallion
        == e1.medallion
    and e2.ride_duration
        > e1.ride_duration],
    e3 = TaxiSecStr[e3.medallion
        == e2.medallion
    and e3.ride_duration
        > e2.ride_duration],
    e4 = TaxiSecStr[e4.medallion
        == e3.medallion
    and e4.ride_duration
        > e3.ride_duration]
select e1.medallion as medallion,
    e1.pickup_grid_x as grid_x,
    e1.pickup_grid_y as grid_y
insert into CongestedAreasStr;

```

VI. MOST PLEASANT TAXI DRIVERS

```

from TaxiSecStr
select dropoff_day, hack_license as driver,
    sum(tip_amount) as tips_total
group by dropoff_day, hack_license
order by tips_total DESC
insert into DriversTipsPerDay;

```

VII. CONCLUSÃO