Análise de Dados com Base em Processamento Massivo em Paralelo

Tutoria Aula 5

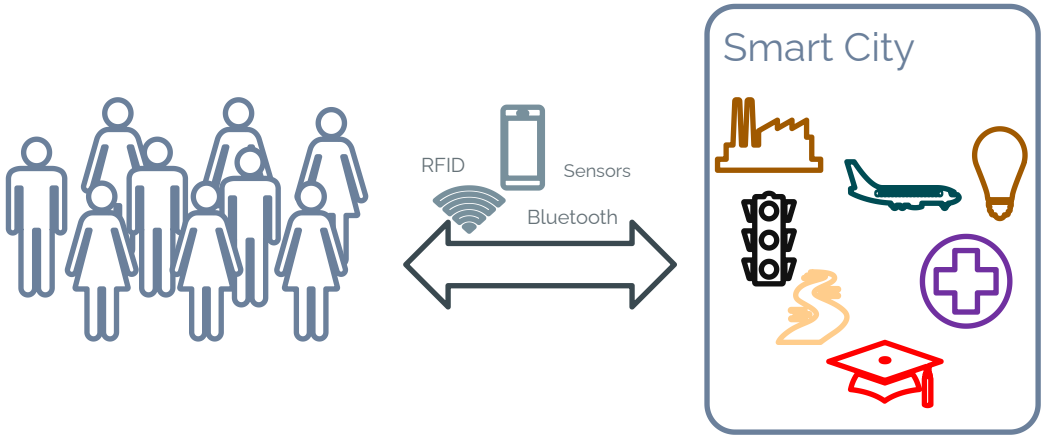
João Paulo Clarindo
ICMC/USP
jpcsantos@usp.br

Sumário

- Introdução
- Consultas espaciais e SOLAP
- Estudo de Caso
  - Esquema-estrela
  - Consultas SOLAP

Introdução

- As cidades estão crescendo!
  - Segundo a ONU, em 2025 haverá 8 bilhões de pessoas no mundo
- Aumento no uso de recursos naturais e serviços em cidades
- Com isso, é possível utilizar dispositivos de Internet das Coisas para o auxílio na tomada de decisão

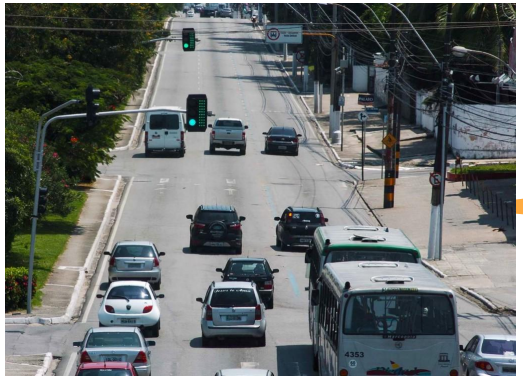


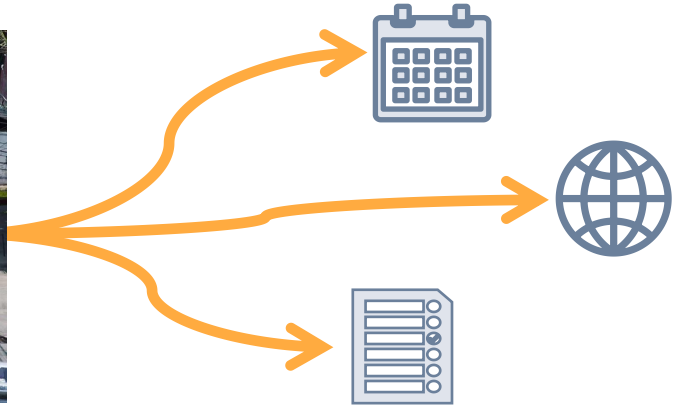
Exemplo de aplicação

- Tráfego Urbano
- Sensores que coletam dados relacionados a:
  - Quantidade de veículos
  - Ruas e avenidas
  - Velocidade média
  - Velocidade máxima

Dados gerados por uma cidade inteligente

- Convencional (nome, velocidade média...)
- Temporal (hora, dia, semana, mês...)
- Espacial (rua, bairro, distrito, cidade...)





Exemplo de consulta

- *Verificar a quantidade de veículos que passaram por um bairro na cidade, agrupados por dia.*

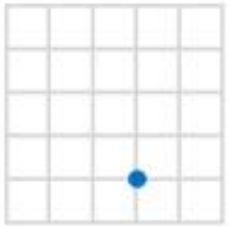
Sumário

- Introdução
- Consultas espaciais e SOLAP
- Estudo de Caso
  - Esquema-estrela
  - Consultas SOLAP

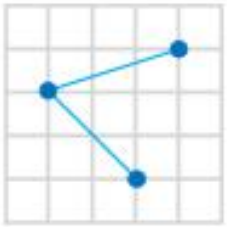
Dados Espaciais

- Dados espaciais (ou geográficos) são representações geométricas de objetos espaciais.
  - Pontos, linhas e polígonos
- Ruas, avenidas, lagoas, rios, cidades, estados

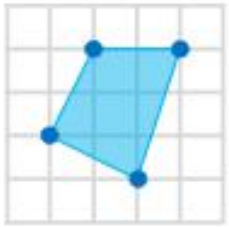
Ponto



Linhas



Polígonos

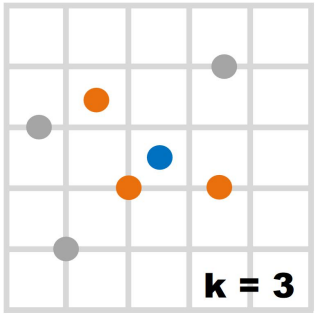


Relacionamentos espaciais

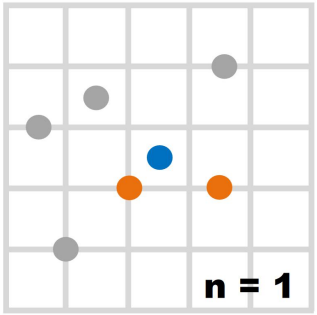
- É como objetos espaciais podem se relacionar nos contextos métricos, topológicos e direcionais
- Métrico:
  - “Qual a distância entre São Carlos e São Paulo?”
- Direcional:
  - “Qual a cidade que está à oeste de São Carlos?”
- Topológico:
  - “Quais as cidades que fazem fronteira com São Carlos?”
- Os relacionamentos espaciais são utilizados em consultas que envolvam dados espaciais.

Consultas espaciais com relacionamentos métricos

- k-NN query: dado um objeto espacial o', encontre os k objetos mais próximos de o'.
- Distance spatial join query: dado um objeto espacial o', encontre os objetos com distância maior (ou menor) ou igual a n.



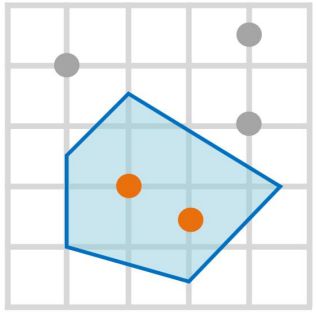
k-NN query



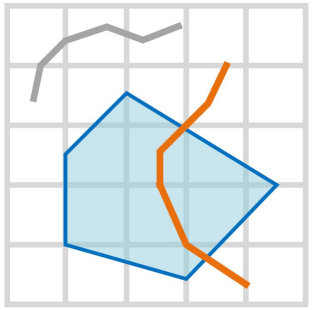
Spatial join query

Consultas espaciais com relacionamentos topológicos

- Containment query: dado um objeto espacial o', encontre todos os objetos espaciais contidos em o'
- Intersection query: dado um objeto espacial o', encontre todos os objetos que contém ao menos um ponto em comum com o'.



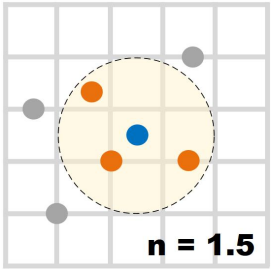
Containment query



Intersection query

Outros tipos de consultas espaciais

- Buffer query: dado um objeto espacial o', encontre todos os objetos espaciais contidos em uma região desenhada em volta de o', com uma distância n de o'



Buffer query

Consultas SOLAP

- São consultas OLAP que oferecem suporte a dados espaciais, utilizando relacionamentos espaciais.
- Os conceitos de OLAP aplicam-se também a SOLAP
    - *Drill-down*
    - *Roll-up*
    - *Slice-and-dice*
    - *Drill-across*
    - *Pivot*

Exemplo de consulta SOLAP

- *Verificar a quantidade de veículos que passaram por um bairro na cidade, agrupados por dia.*

Roll-up

Intersect query

Exemplo de consulta SOLAP

- *Verificar a quantidade de veículos que passaram em uma determinada rua em um mês*

Slice-and-dice

Containment query

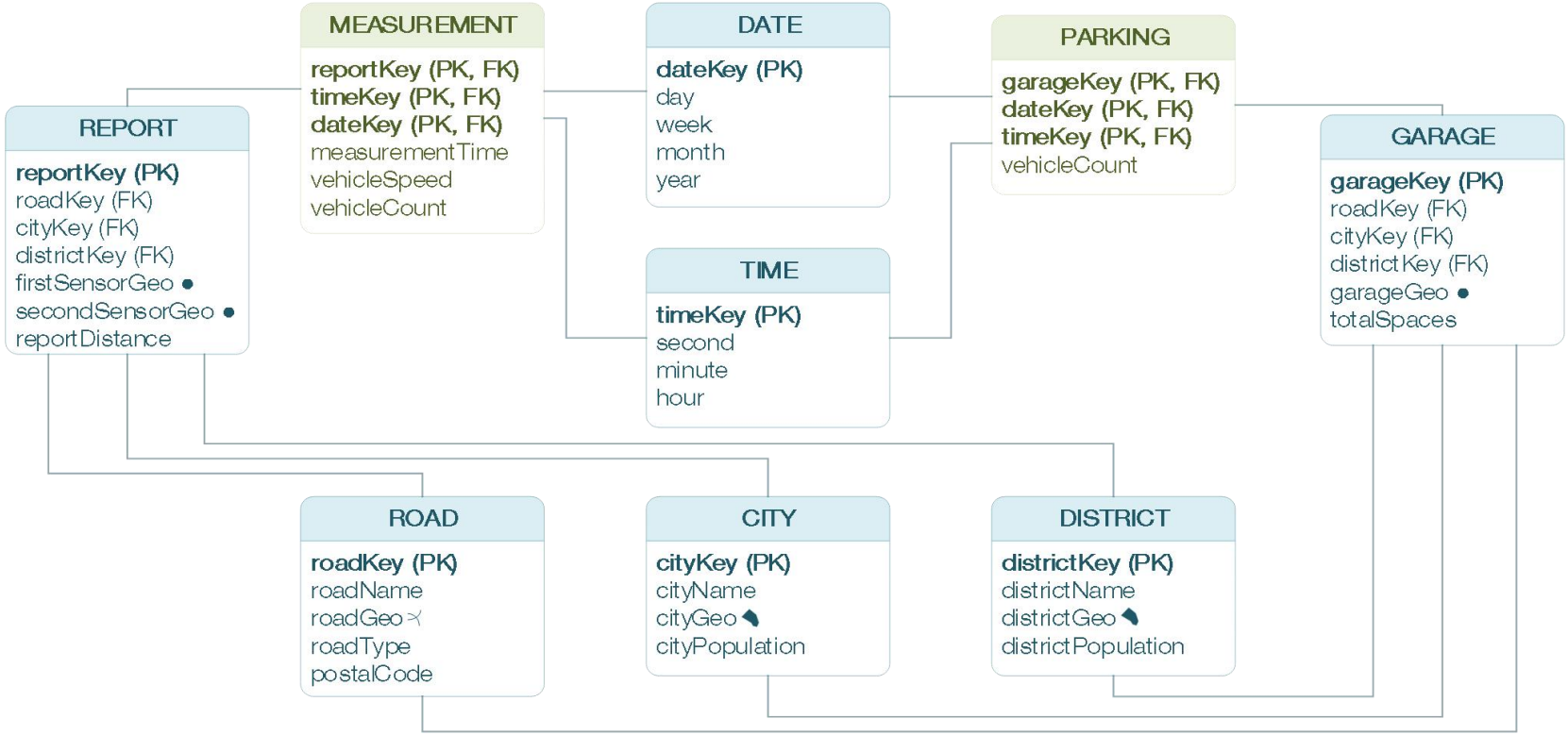
Sumário

- Introdução
- Consultas espaciais e SOLAP
- Estudo de Caso
  - Esquema-estrela
  - Consultas SOLAP

Estudo de Caso

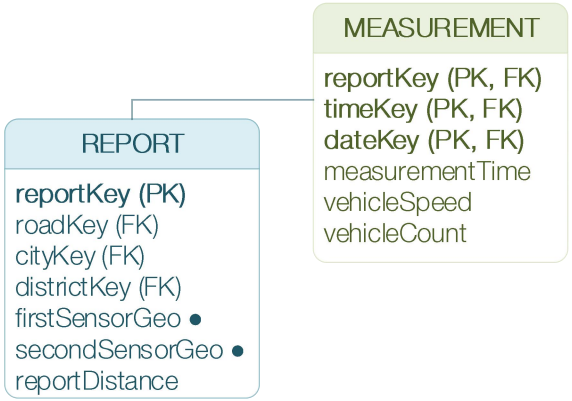
- Utilizamos o dataset CityPulse
- Município de Aarhus, Dinamarca
  - Sensores que coletaram dados do tráfego urbano entre os meses de fevereiro e junho de 2014.

Esquema-estrela



Consultas espaciais com relacionamentos métricos

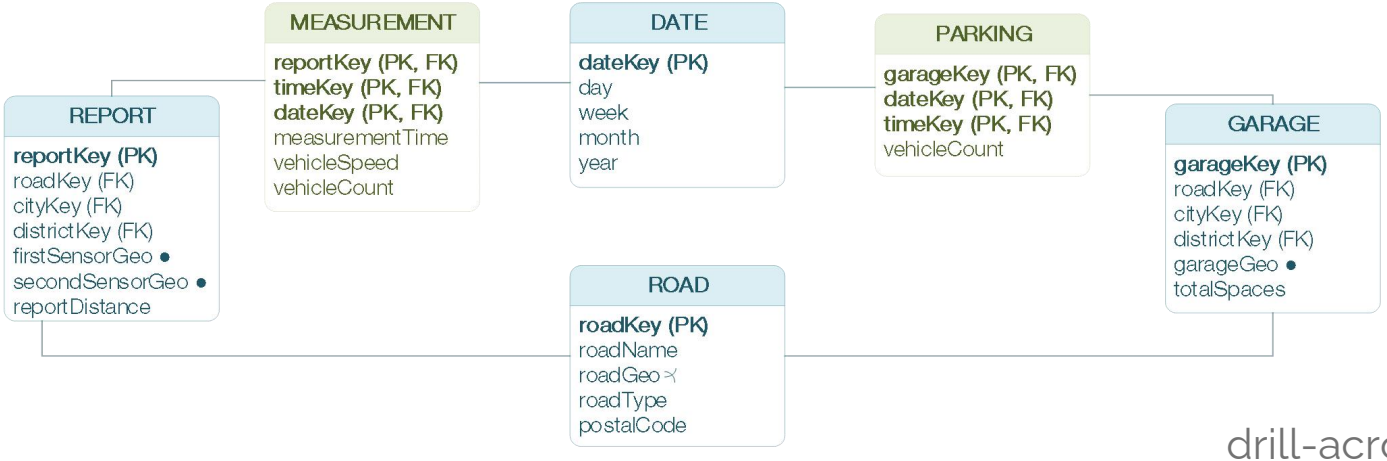
- *Selecione a velocidade média dos dez sensores mais próximos da Catedral de Aarhus no período coletado*



k-NN query

Consultas SOLAP com relacionamentos métricos

- *Selecione a quantidade média de veículos e a quantidade média de veículos etacionados, considerando estacionamentos a 100 m de distância de ruas e avenidas no município de Aarhus que contém sensores*



drill-across

distance spatial join query

Consultas SOLAP com relacionamentos métricos

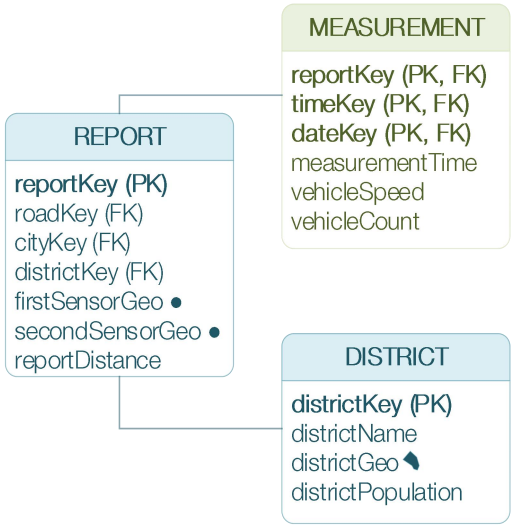
Nome do estacionamento	Nome da rua	Quantidade média de veículos que trafegaram	Quantidade média de veículos estacionados
Bruuns	Værkmestergade	5605	166
Busgadehuset	Frederiksgade	4223	84
Kakvaerksvej	Kalkværksvej	4458	49
Magasin	Åboulevarden	4223	108
Salling	Østergade	3618	191

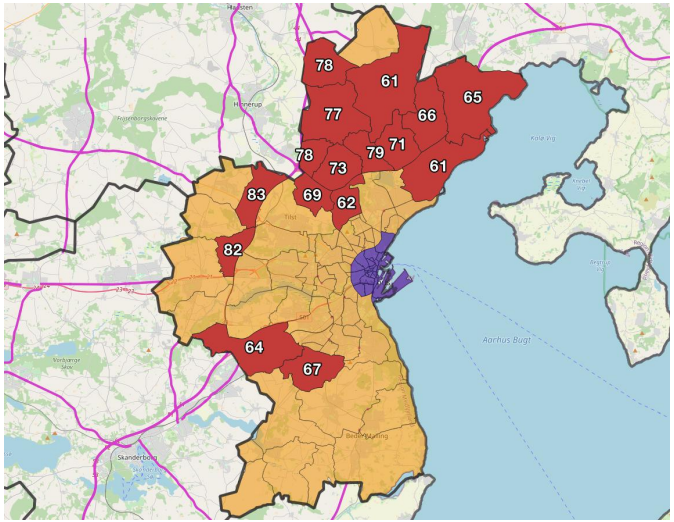
drill-across

distance spatial join query

Consultas espaciais com relacionamentos topológicos

- *Selecione os distritos cuja velocidade média reportada pelos sensores é maior que 60 km/h*

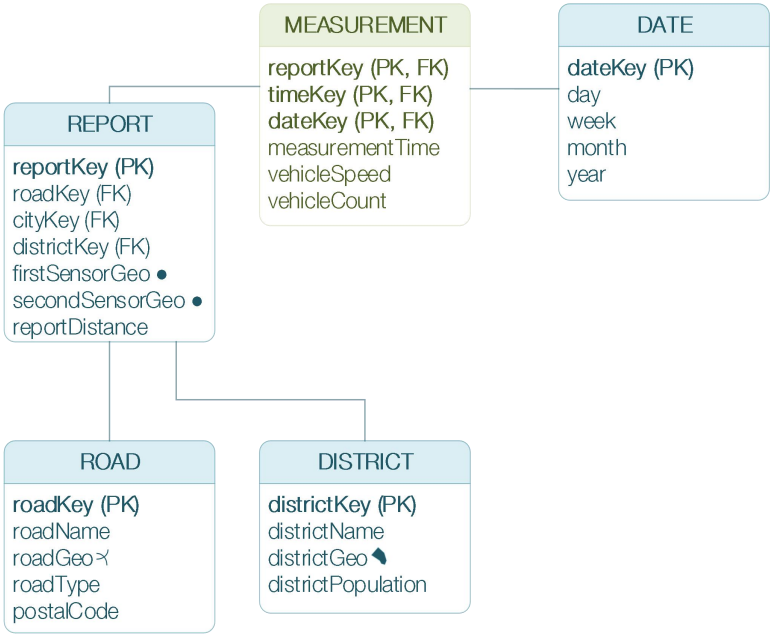


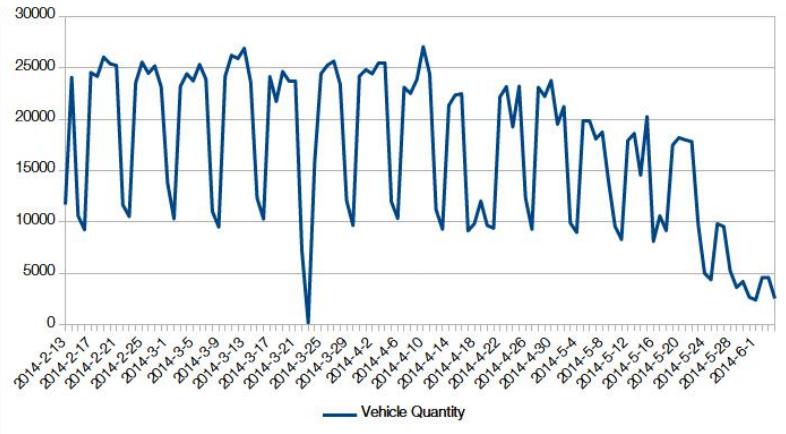


Intersect query

Consultas SOLAP com relacionamentos topológicos

- *Verifique o número de veículos que passaram pelo distrito da Universidade de Aarhus, agrupados por dia*

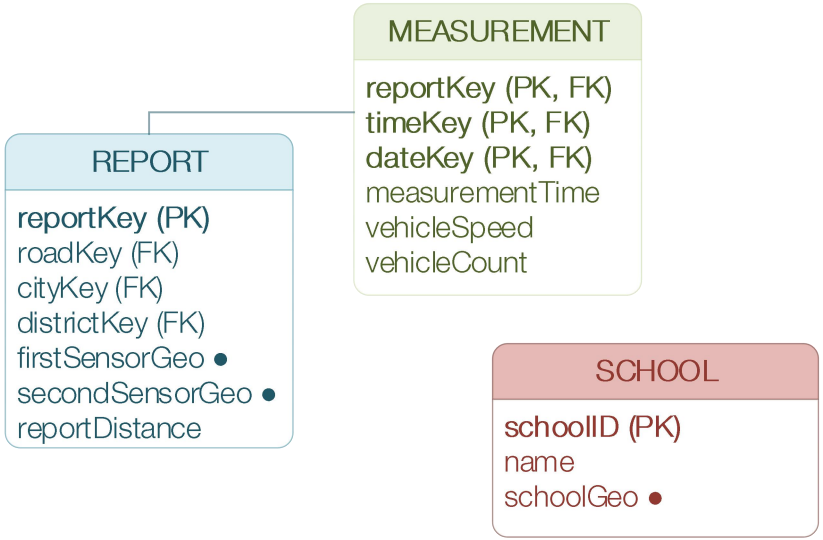


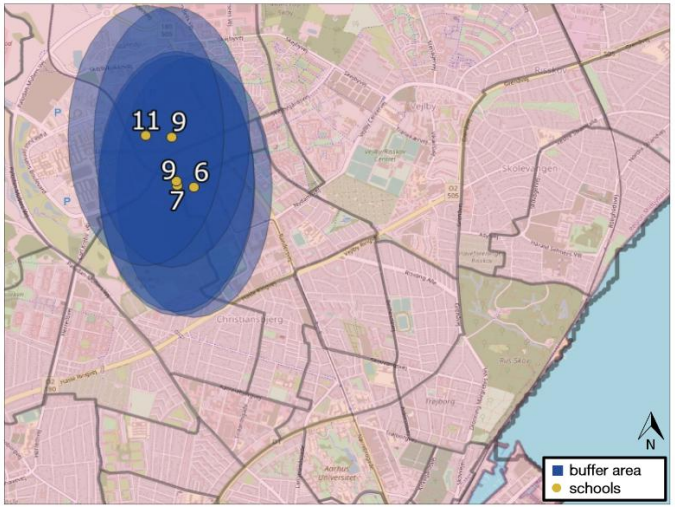


roll-up
containment query
intersect query

Consultas utilizando buffer query

- *Verifique o número de veículos que passaram em uma área de cobertura de 100 m em cada escola de Aarhus, considerando as cinco maiores quantidades.*

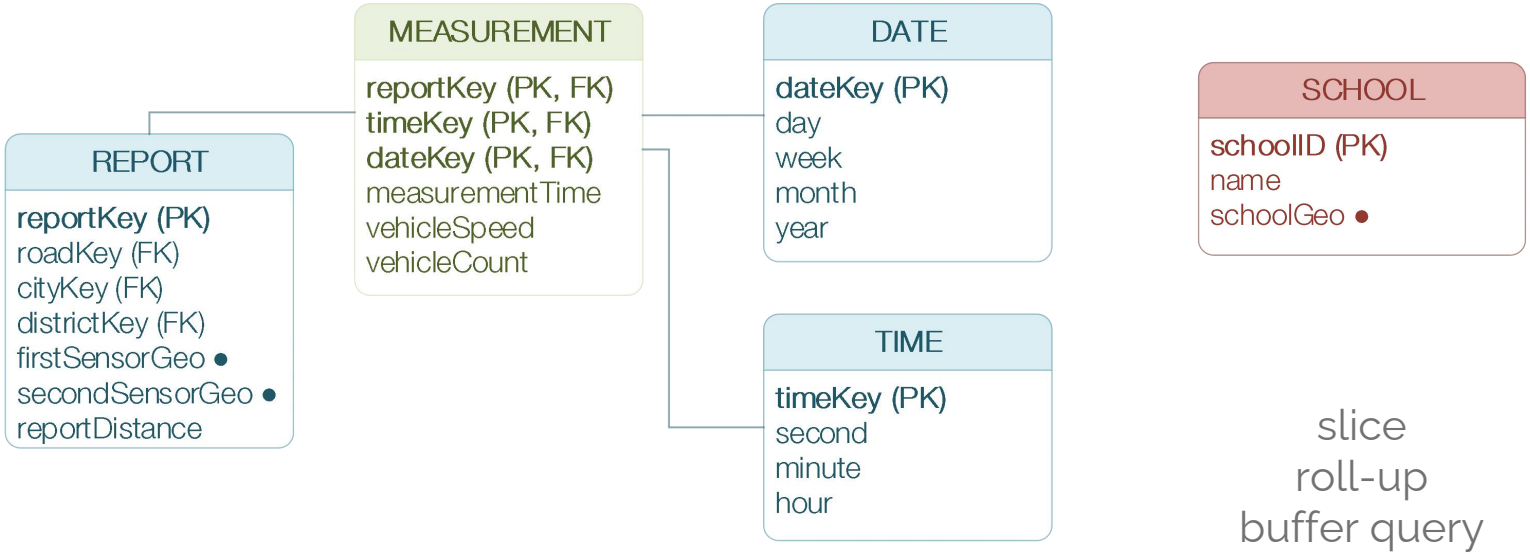




buffer query

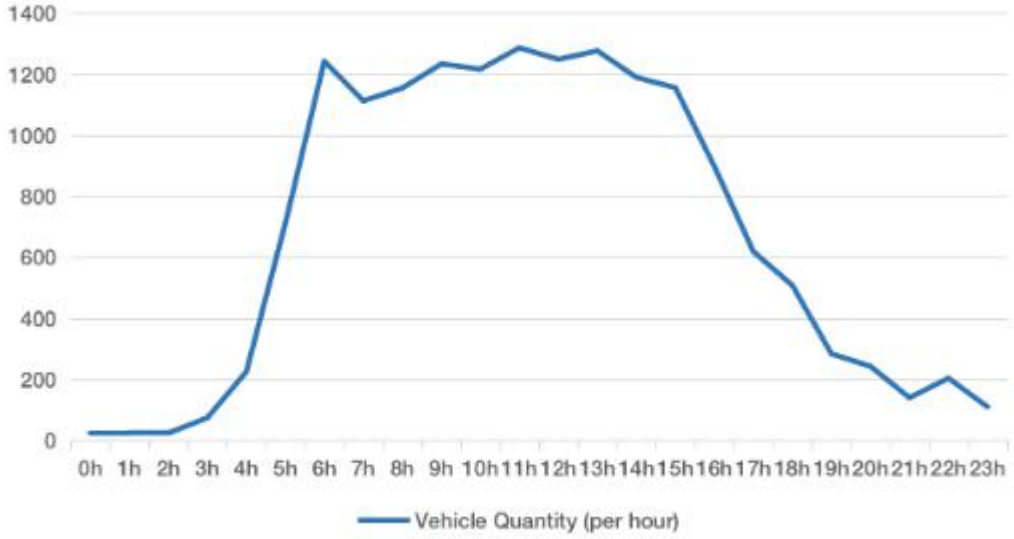
Consultas utilizando buffer query

- *Retorne a quantidade média de veículos que passaram em uma área de cobertura de 100 m em uma escola de Aarhus no mês de fevereiro de 2014, por dia, por hora.*



slice
roll-up
buffer query

Consultas utilizando buffer query



slice
roll-up
buffer query

Bibliografias indicadas para estudo de SOLAP

- **VAISMAN, A.; ZIMÁNYI, E. Data Warehouse Systems: Design and Implementation. Berlin,Heidelberg, Germany: Springer Publishing Company, Incorporated, 2014. 625 p. ISBN 978-3-642-54654-9.**
- HAN, J.; STEFANOVIC, N.; KOPERSKI, K. Selective materialization: An efficient method forspatial data cube construction. In:LNCS. Berlin, Heidelberg, Germany: Springer, 1998.
- RIVEST, S.; BÉDARD, Y.; MARCHAND, P. Toward better support for spatial decision making:defining the characteristics of Spatial On-Line Analytical Processing (SOLAP).Geomatica,v. 55, n. 4, p. 539–555, 2001.

Análise de Dados com Base em Processamento Massivo em Paralelo

Tutoria Aula 5

João Paulo Clarindo
ICMC/USP
jpcsantos@usp.br

EXTRA

Consultas SQL

Consultas espaciais em SQL

- As consultas espaciais podem ser feitas em SQL utilizando funções implementadas pelo SGBD
  - `ST_Distance (geometry g1, geometry g2);`
  - `ST_Contains (geometry geomA, geometry geomB);`
  - `ST_Intersects (geometry geomA , geometry geomB);`
  - `ST_Buffer(geometry g1, float radius_of_buffer, text buffer_style_parameters = '');`

Consultas espaciais em SQL

- Os principais SGBDs relacionais do mercado oferecem suporte a consultas espaciais nativamente ou utilizando extensões:
  - PostgreSQL e PostGIS
  - Oracle e Oracle Spatial and Graph
  - MySQL (nativo)

Consultas espaciais em SQL

- Sistemas de data warehouse oferecidas como serviço na nuvem também oferecem suporte nativo a consultas espaciai:
  - Google Cloud BigQuery
  - Amazon Redshift
  - Microsoft Azure Synapse

Consultas espaciais em Hadoop e Spark

- Hadoop e Spark não oferecem suporte nativo a consultas espaciais, sendo necessário o uso de um **sistema analítico espacial**:
  - SpatialHadoop
  - Apache Sedona
- Mais detalhes em: CASTRO, J. P.; CARNIEL, A.; CIFERRI, C. Analyzing spatial analytics systems based on Hadoop and Spark: A user perspective. Software: Practice and Experience, John Wiley and Sons Ltd, v. 50, n. 12, p. 2121–2144, 12 2020. ISSN 0038-0644.

Consultas do estudo de caso

- As consultas foram feitas em um ambiente de SparkSQL utilizando o sistema analítico espacial Apache Sedona.
- Entretanto, as funções descritas nessas consultas funcionam na grande maioria das soluções apresentadas anteriormente, mas pode ocorrer incompatibilidades.
  - Ex.: Google Cloud BigQuery não oferece suporte a função `ST_Buffer`
- Representações espaciais em texto (latitude e longitude) foram convertidas para objeto espacial utilizando a função `ST_GeomFromWKT`, que pode ser incompatível com outras soluções, como o PostGIS.

k-NN query

```
SELECT AVG(vehicleSpeed), ST_MakeLine(firstSensorGeo,
    secondSensorGeo) AS reportGeo
FROM measurement, report
WHERE measurement.reportID = report.reportID
GROUP BY reportGeo
ORDER BY ST_Distance(reportGeo,
    ST_GeomFromWKT('POINT(10.210556 56.156944)'))
LIMIT 10
```

Drill-across com distance spatial join query

```
SELECT garage.garageID AS "Garage Name",
    road.roadName AS "Street Name",
    ROUND(AVG(measurement.vehiclecount),0)
        AS "Average number of vehicles in traffic",
    ROUND(AVG(parking.vehiclecount),0)
        AS "Average number of parked vehicles"
FROM (SELECT reportID, SUM(vehiclecount) AS vehiclecount
      FROM measurement, date
      WHERE measurement.dateID = date.dateID
        AND month = 5 AND week = 4
      GROUP BY reportID) AS measurement,
```

continua →

Drill-across com distance spatial join query

```
(SELECT parking.garageID,
        AVG(vehiclecount) AS vehiclecount
      FROM parking, date
      WHERE parking.parkingID = date.dateID
        AND month = 5 AND week = 4
      GROUP BY garageID) AS parking,
      report, garage, road
WHERE report.reportID = measurement.reportID
      AND parking.garageID = garage.garageID
      AND garage.roadID = road.roadID
      AND ST_Distance(
          ST_MakeLine(report.firstsensorgeo,
            report.secondsensorgeo), garage.geo) <= 0.01
GROUP BY garage.garageID
```

Intersect query

```
SELECT districtGeo, AVG(vehicleSpeed) AS a
FROM measurement, report, district
WHERE ST_Intersects(ST_MakeLine(firstSensorGeo,
        secondSensorGeo), districtGeo)
    AND measurement.reportID = report.reportID
    AND report.districtID = district.districtID
GROUP BY districtGeo
HAVING a >= 60
```

Roll-up com containment e intersect query

```
SELECT day, month, SUM(vehicleCount)
FROM measurement, date, report, district, road
WHERE ST_Contains(districtGeo, roadGeo)
    AND ST_Intersects(roadGeo, ST_MakeLine(firstSensorGeo,
        secondSensorGeo))
    AND measurement.reportID = report.reportID
    AND measurement.dateID = date.dateID
    AND report.districtID = district.districtID
    AND report.roadID = road.roadID
    AND district.name = 'Universitetet/Kommunehospitalet'
GROUP BY day, month
ORDER BY day, month
```

Buffer query

```
SELECT ROUND(AVG(measurement.vehicleCount),0)
    as AVGvehicleCount, schoolID
FROM schools, report, measurement,
    ST_Buffer(schools.geom, 0.01) AS schoolbuffer,
    ST_MakeLine(firstSensorGeo, secondSensorGeo)
      AS sensorset
WHERE ST_Intersects(sensorset,schoolbuffer)
    AND measurement.reportID = report.reportID
GROUP BY schoolID
ORDER BY AVGvehicleCount DESC
LIMIT 5
```

Slice e roll-up com buffer query

```
SELECT SUM(measurement.vehiclecount), hour, day
FROM schools, report, measurement, date, time
    ST_Buffer(schools.geom, 0.01) AS schoolbuffer,
    ST_MakeLine(firstSensorGeo, secondSensorGeo) AS sensorset
WHERE ST_Intersects(sensorset,schoolbuffer)
    AND measurement.reportID = report.reportID
    AND measurement.dateID = date.dateID
    AND measurement.timeID = time.timeID
    AND schools.schoolID = 4448940550
    AND month = 2
GROUP BY hour, day
ORDER BY day, hour
```

Análise de Dados com Base em Processamento Massivo em Paralelo

Tutoria Aula 5

João Paulo Clarindo
ICMC/USP
jpcsantos@usp.br