Creating a Traffic Accident Database

In a Smart City Use Case with a Wide-Column Store Database

By Andreas Kruff, Anh Huy Matthias Tran

Agenda

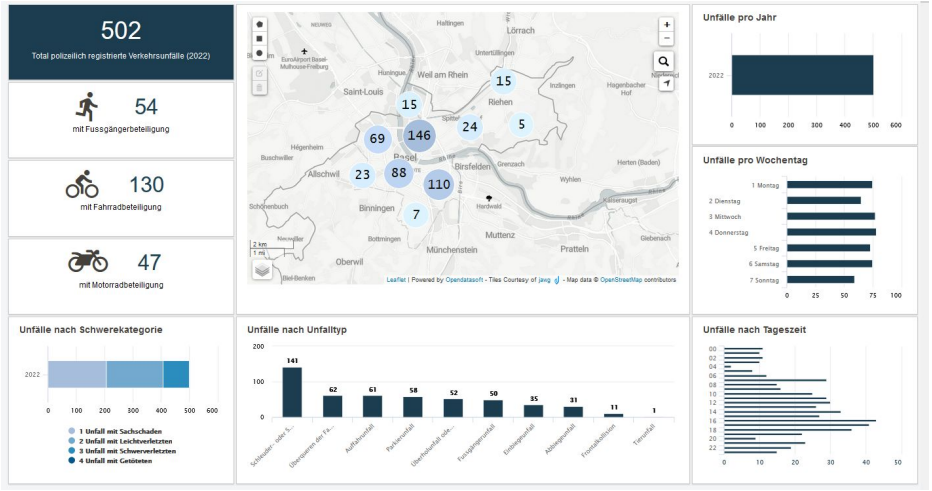
1. Introduction
2. Choosing a Database
3. Data Modeling
    a. Oracle DB
    b. Cassandra
4. Evaluation
    a. Query Performance
    b. Limitations & Considerations
5. Outlook: NoSQL Techniques
6. Conclusion

1. Introduction

Project Motivation

Open Data Project: 'Datenportal Basel Stadt'

Smart city project collecting live sensor data and feeding it towards various tables.



https://data.bs.ch/pages/verkehrsunfall_dashboard_basel/

Application Case

- Traffic Accident Analytics
- Via Monthly Traffic Reports

502

Total polizeilich registrierte Verkehrsunfälle (2022)

54

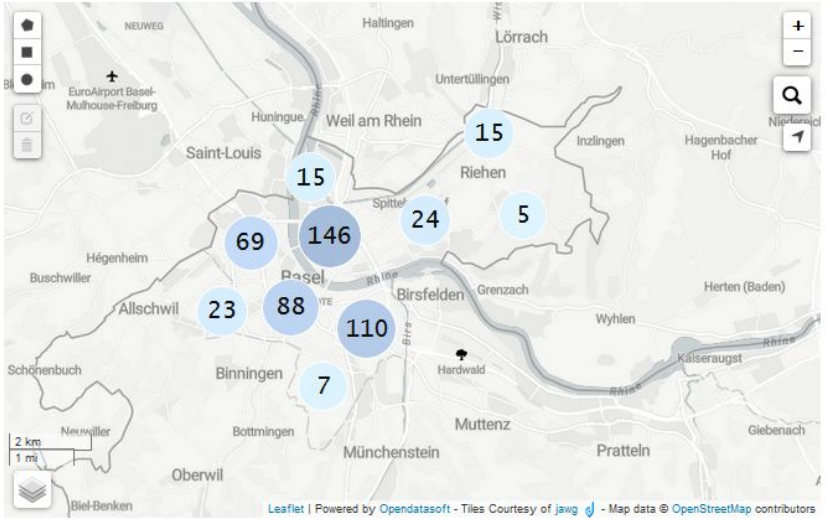
mit Fussgängerbeteiligung

130

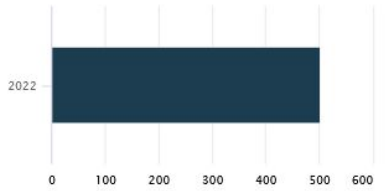
mit Fahrradbeteiligung

47

mit Motorradbeteiligung

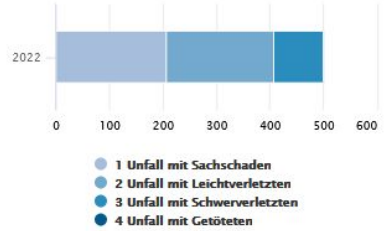


Unfälle pro Jahr

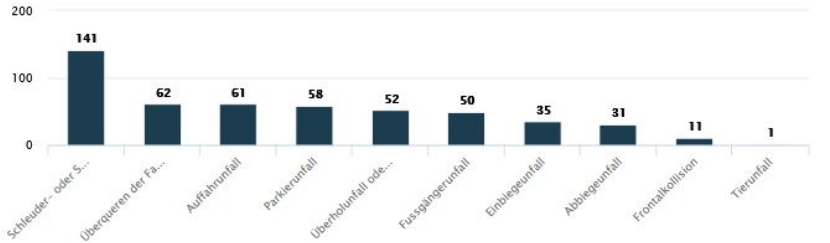


Unfälle pro Wochentag

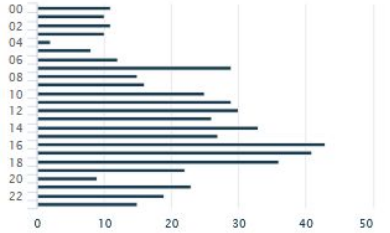
Unfälle nach Schwerekategorie

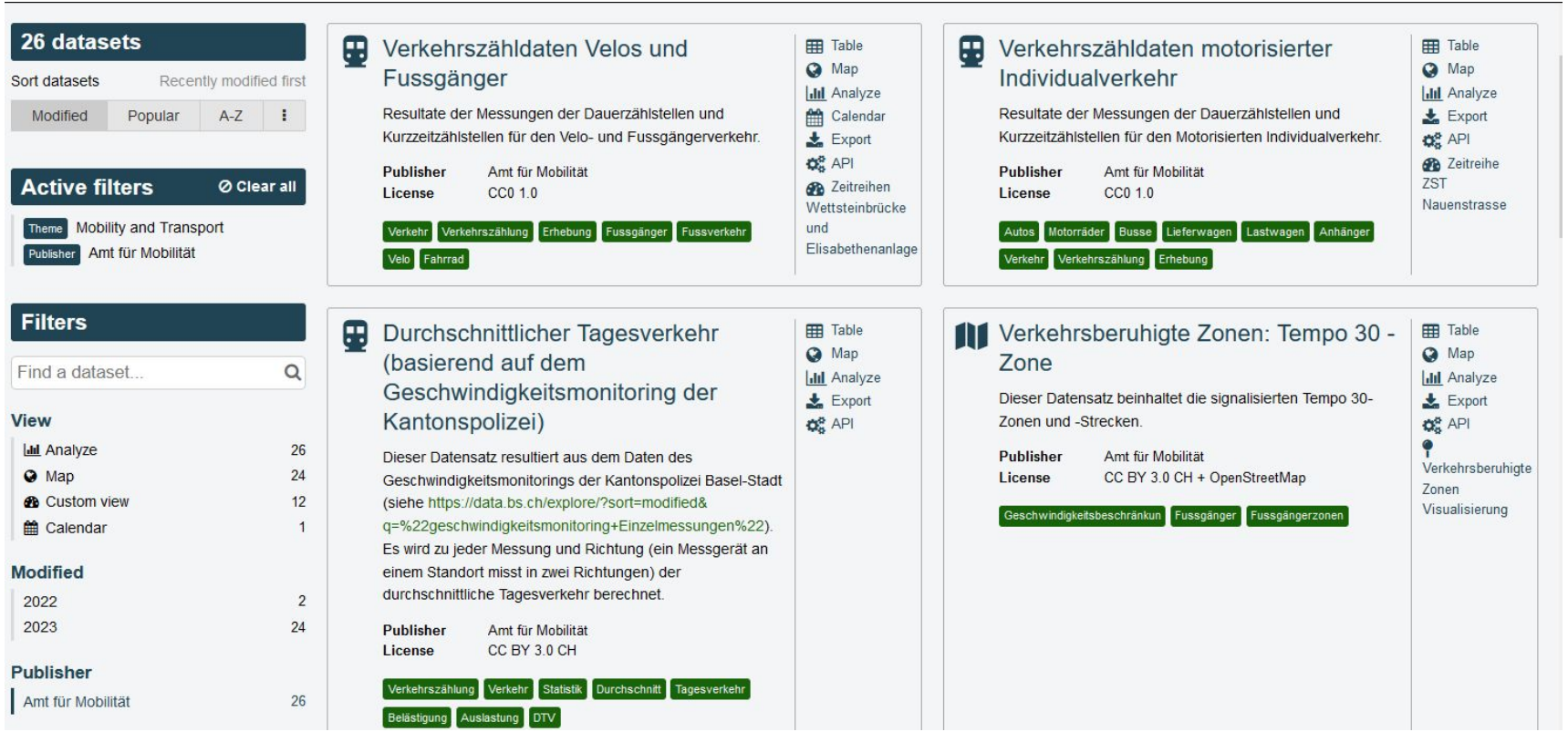


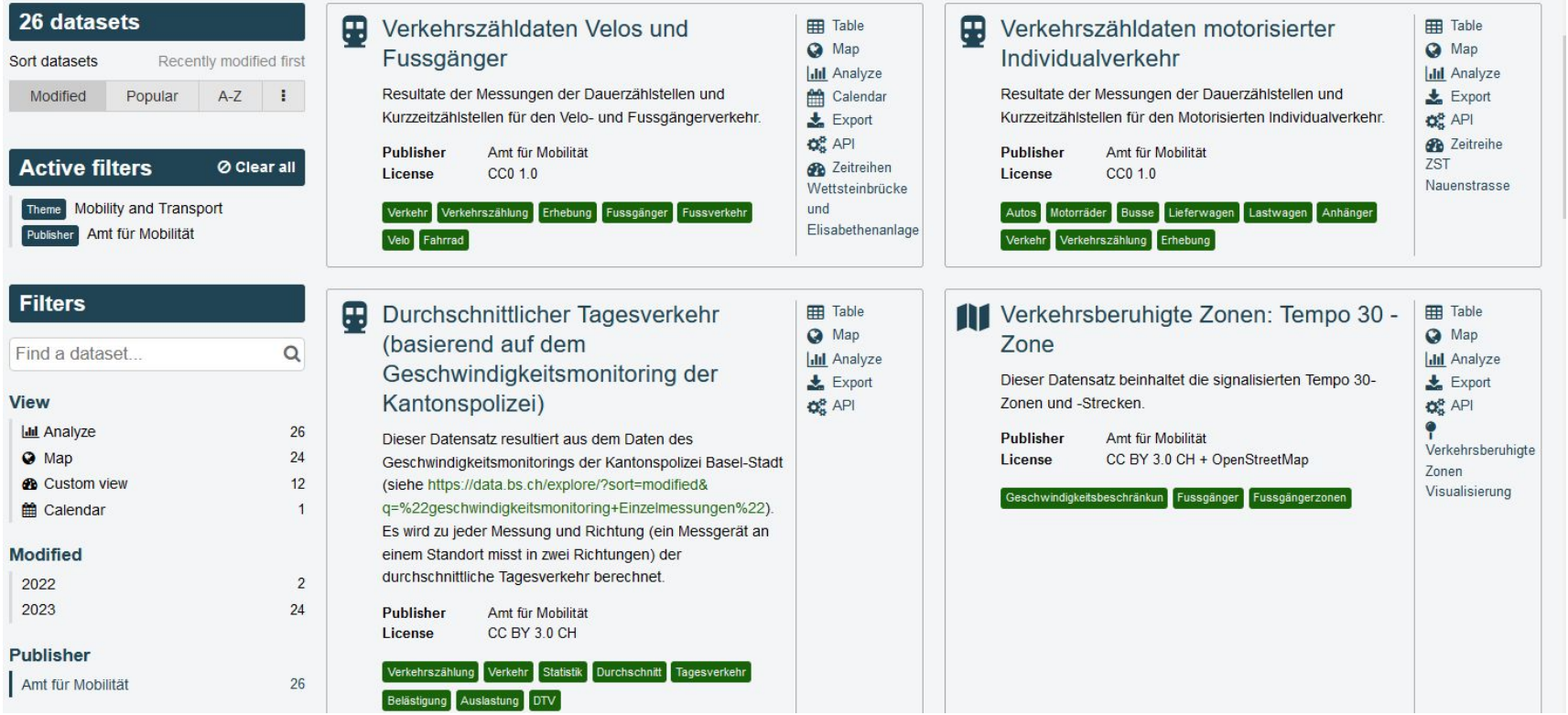
Unfälle nach Unfalltyp



Unfälle nach Tageszeit




26 datasets
Sort datasets
Recently modified first
Modified
Popular
A-Z
Active filters
Clear all
Theme
Mobility and Transport
Publisher
Amt für Mobilität
Filters
Find a dataset...
View
Analyze 26
Map 24
Custom view 12
Calendar 1
Modified
2022 2
2023 24
Publisher
Amt für Mobilität 26
Verkehrszähldaten Velos und Fussgänger
Resultate der Messungen der Dauerzählstellen und Kurzzeitzählstellen für den Velo- und Fussgängerverkehr.
Publisher Amt für Mobilität
License CC0 1.0
Verkehr
Verkehrszählung
Erhebung
Fussgänger
Fussverkehr
Velo
Fahrrad
Table
Map
Analyze
Calendar
Export
API
Zeitreihen Wettsteinbrücke und Elisabethenanlage
Verkehrszähldaten motorisierter Individualverkehr
Resultate der Messungen der Dauerzählstellen und Kurzzeitzählstellen für den Motorisierten Individualverkehr.
Publisher Amt für Mobilität
License CC0 1.0
Autos
Motorräder
Busse
Lieferwagen
Lastwagen
Anhänger
Verkehr
Verkehrszählung
Erhebung
Table
Map
Analyze
Export
API
Zeitreihe ZST Nauenstrasse
Durchschnittlicher Tagesverkehr (basierend auf dem Geschwindigkeitsmonitoring der Kantonspolizei)
Dieser Datensatz resultiert aus dem Daten des Geschwindigkeitsmonitorings der Kantonspolizei Basel-Stadt (siehe https://data.bs.ch/explore/?sort=modified&q=%22geschwindigkeitsmonitoring+Einzelmessungen%22). Es wird zu jeder Messung und Richtung (ein Messgerät an einem Standort misst in zwei Richtungen) der durchschnittliche Tagesverkehr berechnet.
Publisher Amt für Mobilität
License CC BY 3.0 CH
Verkehrszählung
Verkehr
Statistik
Durchschnitt
Tagesverkehr
Belästigung
Auslastung
DTV
Table
Map
Analyze
Export
API
Verkehrsberuhigte Zonen: Tempo 30 - Zone
Dieser Datensatz beinhaltet die signalisierten Tempo 30-Zonen und -Strecken.
Publisher Amt für Mobilität
License CC BY 3.0 CH + OpenStreetMap
Geschwindigkeitsbeschränkun
Fussgänger
Fussgängerzonen
Table
Map
Analyze
Export
API
Verkehrsberuhigte Zonen Visualisierung



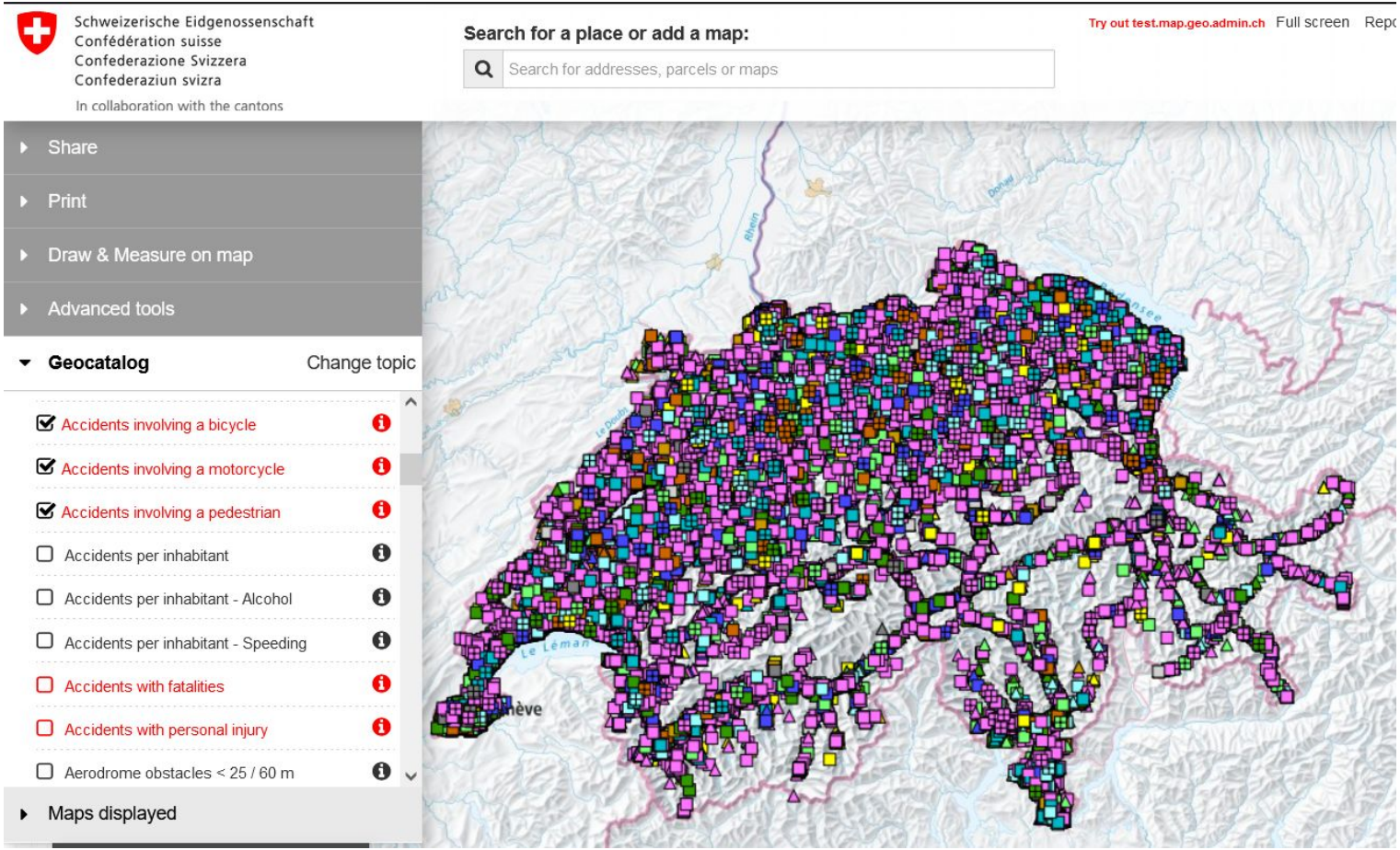
➔ However: Dataset too small

Data Set

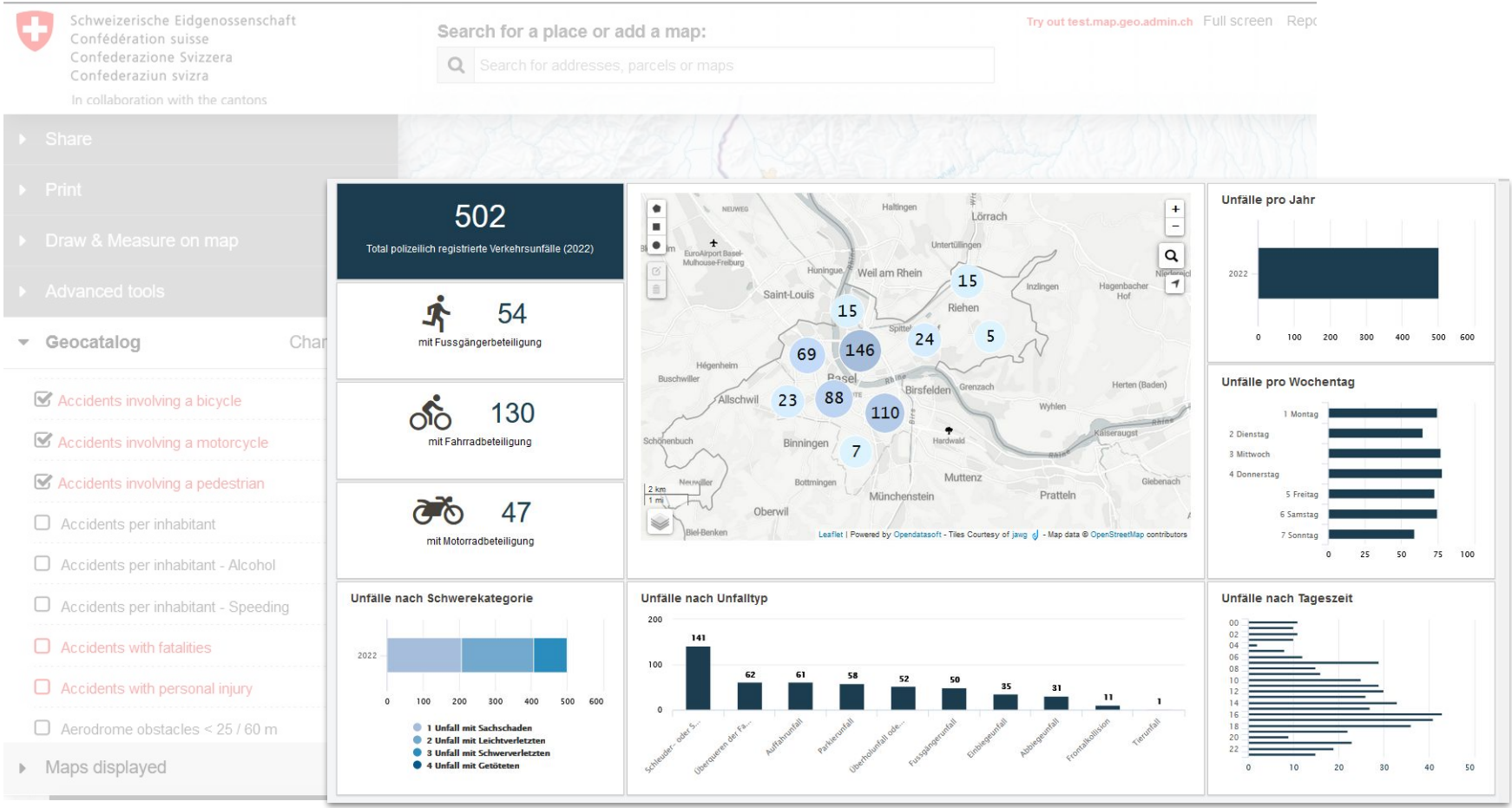
214,049 Traffic Accident Data Points by ‘Bundesamt für Strassen in Switzerland between **2011 ~ 2022**

- **Temporal Data**
  Year, Month, Day, Weekday, Hour
- **Geolocation Data**
  Canton Code, Municipality Code, Geo-Coordinates
- **Traffic Accident Data**
  Pedestrian, Bicycle, Motorcycle
  Severity, Accident Type, Road Type

https://data.geo.admin.ch/ch.astra.unfaelle-personenschaeden_alle/



- https://map.geo.admin.ch


Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
In collaboration with the cantons
Search for a place or add a map:
Search for addresses, parcels or maps
Try out test.map.geo.admin.ch
Full screen
Share
Print
Draw & Measure on map
Advanced tools
Geocatalog
Accidents involving a bicycle
Accidents involving a motorcycle
Accidents involving a pedestrian
Accidents per inhabitant
Accidents per inhabitant - Alcohol
Accidents per inhabitant - Speeding
Accidents with fatalities
Accidents with personal injury
Aerodrome obstacles < 25 / 60 m
Maps displayed
502
Total polizeilich registrierte Verkehrsunfälle (2022)
54
mit Fussgängerbeteiligung
130
mit Fahrradbeteiligung
47
mit Motorradbeteiligung
Unfälle pro Jahr
2022
0
100
200
300
400
500
600
Unfälle pro Wochentag
1 Montag
2 Dienstag
3 Mittwoch
4 Donnerstag
5 Freitag
6 Samstag
7 Sonntag
0
25
50
75
100
Unfälle nach Schwerekategorie
2022
0
100
200
300
400
500
600
1 Unfall mit Sachschaden
2 Unfall mit Leichtverletzten
3 Unfall mit Schwerverletzten
4 Unfall mit Getöteten
Unfälle nach Unfalltyp
200
100
0
141
62
61
58
52
50
35
31
11
1
Schleuder- oder S...
Überqueren der Fa...
Auffahrunfall
Parkierunfall
Überholunfall oder...
Fussgängerunfall
Einbiegeunfall
Abbiegeunfall
Frontalkollision
Tierunfall
Unfälle nach Tageszeit
00
02
04
06
08
10
12
14
16
18
20
22
0
10
20
30
40
50
Leaflet | Powered by Opendatasoft - Tiles Courtesy of jawg - Map data © OpenStreetMap contributors

Use Case

Monthly Traffic Accident Report

Inspecting Key Queries involving …

1. **Temporal Data**

   Year, Month, Day, Weekday, Hour

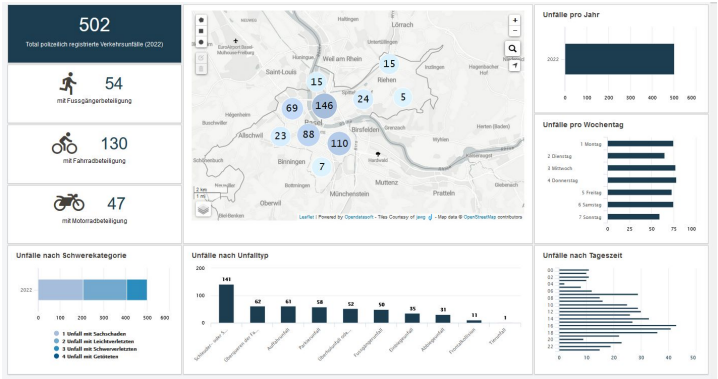
2. **Geolocation Data**

   Canton Code, Municipality Code

3. **Traffic Accident Data**

   Pedestrian, Bicycle, Motorcycle

   Severity, Accident Type, Road Type



2. Choosing a NoSQL Database

Defining Use Case Requirements: Monthly Traffic Accident Report

Traffic Accident Data from Switzerland from **2011 ~ 2022**

- **Fast Query Performance for Key Queries**
  Fast Aggregate Calculation on a monthly basis
  For Key Queries
- **Consistent Writing & Reading Operations**
  Analysis use case requires accurate data
- **Scalability**
  New sensor technology requires new tables & categories

Use Case Requirements

CAP-Theorem

- **Consistency over Availability**

  Analysis use case requires accurate data

  ➔ CP

PACELC-Theorem

- **Prioritize:**

  Consistency over Availability

  Consistency over Latency

  ➔ PC+EC

Cassandra: Key Features

- **Query-driven Design**

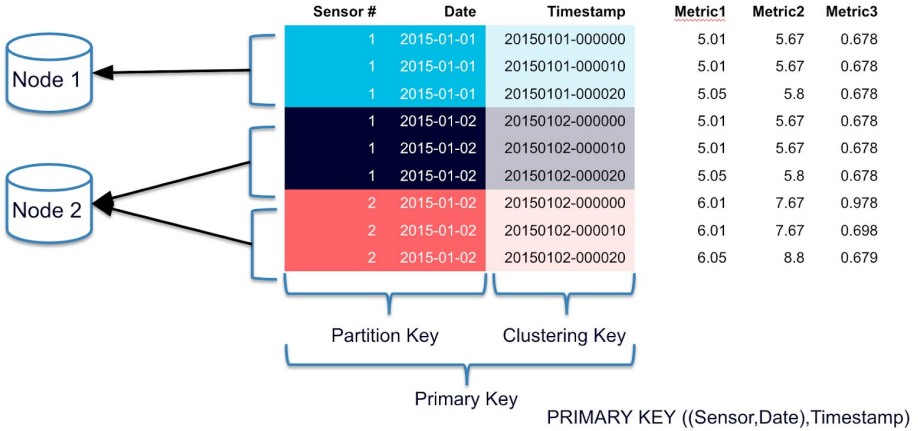
  Schema is created around **key queries**

- **Fast Performance on Aggregates**

  As a wide-column store

- **Scalability**

  Via clusters, nodes & v-nodes



➔ **Sharding via:** *partition keys & clustering keys*

https://www.instaclustr.com/blog/cassandra-data-partitioning/

Cassandra & the CAP Theorem

- **Tuneable Consistency**

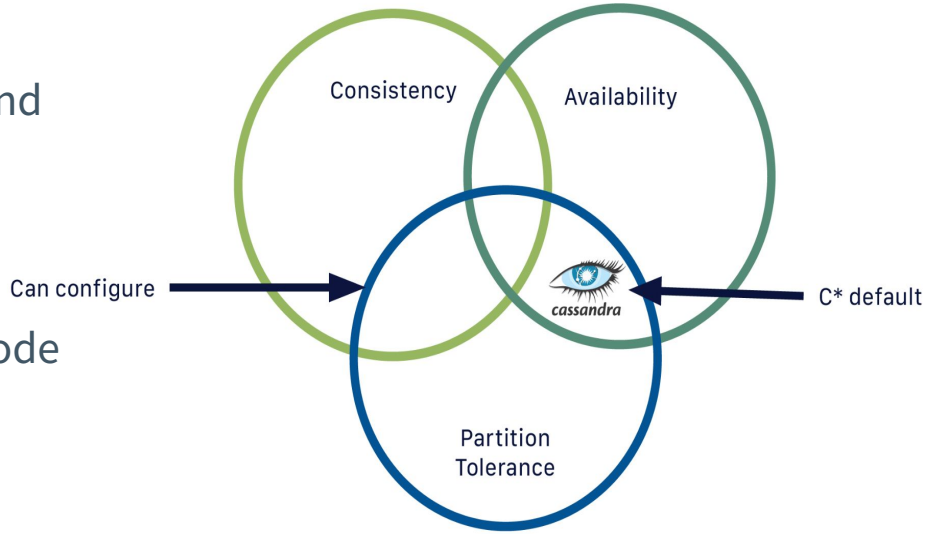
  Via quorum consistency levels and replication factors

- **Availability**

  Via failure protocols in case of node failure

- **Partition Tolerance**

  Via clusters, nodes and v-nodes



➔ **Evaluate: Performance** on different **Consistency Levels**

3. Data Modeling

Database Setup

OracleDB as Relational Database
Deployed in a docker environment

Cassandra as Wide-Column Store Database
Deployed in a docker environment

- 4 Nodes & Replication Factor 3

Cassandra DB

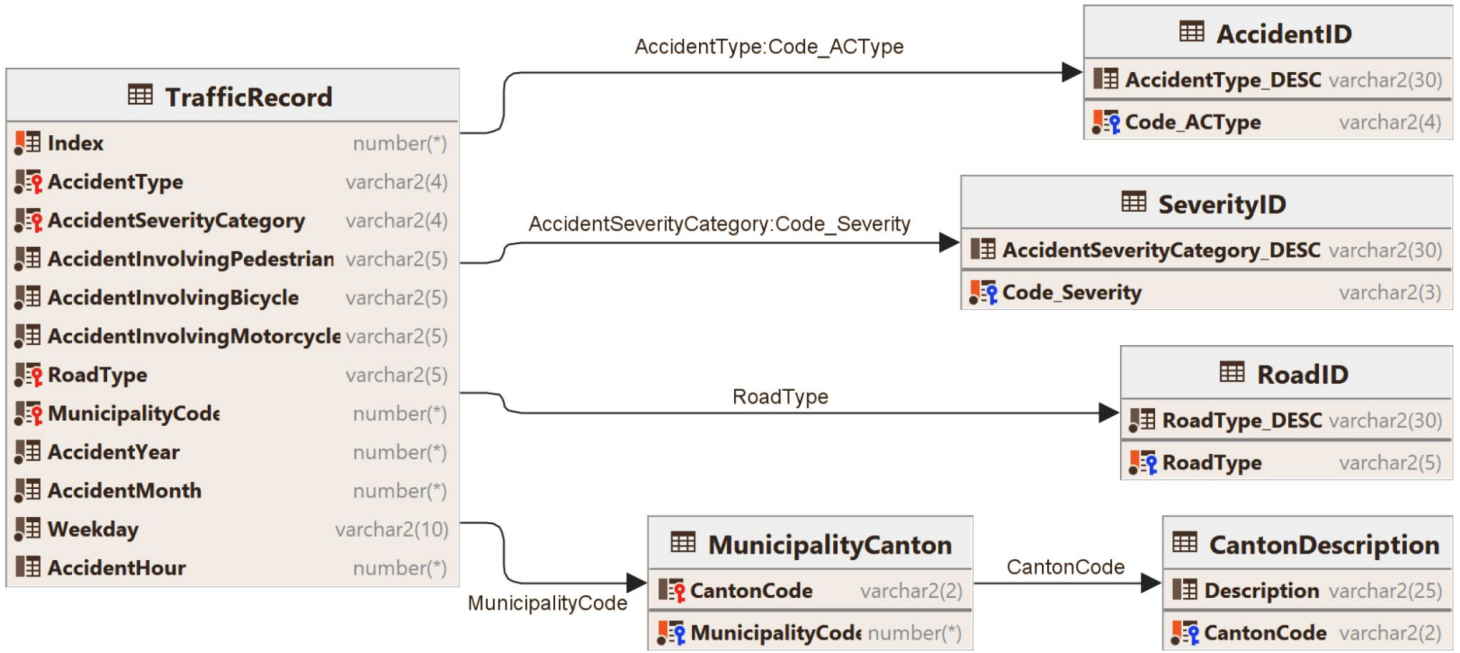
Wide-Column Store Database

Deployed in a docker environment

Setup

4 Nodes & Replication Factor 3

Oracle DB


TrafficRecord
Index number(*)
AccidentType varchar2(4)
AccidentSeverityCategory varchar2(4)
AccidentInvolvingPedestrian varchar2(5)
AccidentInvolvingBicycle varchar2(5)
AccidentInvolvingMotorcycle varchar2(5)
RoadType varchar2(5)
MunicipalityCode number(*)
AccidentYear number(*)
AccidentMonth number(*)
Weekday varchar2(10)
AccidentHour number(*)
AccidentType:Code_ACType
AccidentID
AccidentType_DESC varchar2(30)
Code_ACType varchar2(4)
AccidentSeverityCategory:Code_Severity
SeverityID
AccidentSeverityCategory_DESC varchar2(30)
Code_Severity varchar2(3)
RoadType
RoadID
RoadType_DESC varchar2(30)
RoadType varchar2(5)
MunicipalityCode
MunicipalityCanton
CantonCode varchar2(2)
MunicipalityCode number(*)
CantonCode
CantonDescription
Description varchar2(25)
CantonCode varchar2(2)

Keyqueries (in CQL)

[Q0] Query for a specific accident by ID

```
SELECT * FROM traffic_accidents.accidents WHERE accidentid = 37034;
```

[Q1] Query for all accidents in a specific canton and a specific year

```
SELECT *
FROM traffic_accidents.accidents_by_year
WHERE accidentyear = 2012
 AND cantoncode = 'ZH';
```

[Q2] Query for the involvement of bicycles, pedestrians or motorcycles

```
SELECT *
FROM traffic_accidents.accidents_by_pedestrian
WHERE cantoncode = 'ZH'
  AND accidentyear = 2012
  AND accidentinvolvingpedestrian = true;
```

Keyqueries (in CQL)

[Q3] Query to compare road types based on the amount of accidents

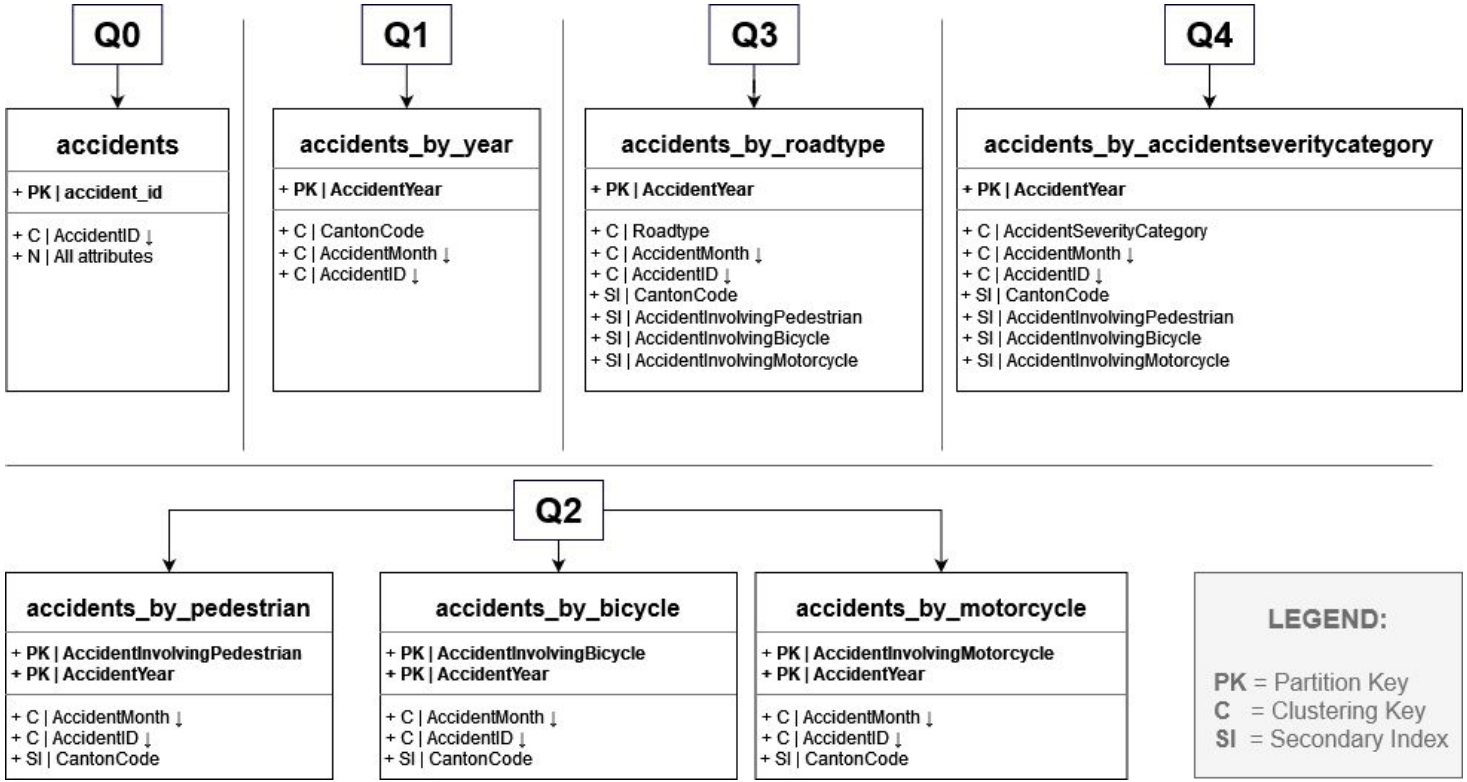
```
SELECT roadtype, COUNT(*) AS RoadTypeCount
FROM accident_by_roadtype
WHERE accidentinvolvingbicycle = true AND accidentyear = 2013
GROUP BY roadtype;
```

[Q4] Query to identify the severity of the accidents for the different parties involved

```
SELECT COUNT(*) as bicycle_count
FROM traffic_accidents.accident_by_AccidentSeverityCategory
WHERE accidentyear = 2013 AND AccidentSeverityCategory = 'as2' AND accidentinvolvingmotorcycle =true;
```

Cassandra DB

Cassandra



4. Experiments & Evaluation

General Set-Up

Consistency Levels

- **ONE:** Request one replica
- **QUORUM:** Request majority of replica (⅔)
- **ALL:** Request all replicas

Query Performance & Reading Operations

How fast for each:

key query and **consistency level**

Query Performance

Metric:

→ Latency (in ms)

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Cassandra	Q0	ALL	87	+ 97.72
Oracle	Q1	-	100	-
Cassandra	Q1	ONE	46	- 54
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Cassandra	Q3	ONE	67	- 36.19
Cassandra	Q3	QUORUM	341	+224.77
Cassandra	Q3	ALL	289	+ 175.24
Oracle	Q4	-	77	-
Cassandra	Q4	ONE	78	+ 1.29
Cassandra	Q4	QUORUM	91	+ 18.19
Cassandra	Q4	ALL	81	+ 5.19

Table 2 Query Latency in Milliseconds with the best Performance marked green (grouped by Database, Query and Consistency Level)

Query Performance

Metric:

→ Latency

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Oracle	Q1	-	100	-
Cassandra	Q1	ONE	46	- 54
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Cassandra	Q3	ONE	67	- 36.19
Cassandra	Q3	QUORUM	341	+224.77
Cassandra	Q3	ALL	289	+ 175.24
Cassandra	Q4	ONE	78	+ 1.29
Cassandra	Q4	QUORUM	91	+ 18.19
Cassandra	Q4	ALL	81	+ 5.19

Table 2 Query Latency in Milliseconds with the best Performance marked green (grouped by Database, Query and Consistency Level)

Query Performance

Metric:

➔ Latency (in ms)

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Cassandra	Q0	ALL	87	+ 97.72
Oracle	Q1	-	100	-
Cassandra	Q1	ONE	46	- 54
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Oracle	Q4	-	77	-
Cassandra	Q4	ONE	78	+ 1.29
Cassandra	Q4	QUORUM	91	+ 18.19
Cassandra	Q4	ALL	81	+ 5.19

Table 2 Query Latency in Milliseconds with the best Performance marked green (grouped by Database, Query and Consistency Level)

Query Performance

Metric:

→ Latency (in ms)

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Cassandra	Q0	ALL	87	+ 97.72
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Cassandra	Q3	ONE	67	- 36.19
Cassandra	Q3	QUORUM	341	+224.77
Cassandra	Q3	ALL	289	+ 175.24
Oracle	Q4	-	77	-
Cassandra	Q4	ONE	78	+ 1.29
Cassandra	Q4	QUORUM	91	+ 18.19
Cassandra	Q4	ALL	81	+ 5.19

Table 2 Query Latency in Milliseconds with the best Performance marked green (grouped by Database, Query and Consistency Level)

Query Performance

Metric:

→ Latency (in ms)

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Cassandra	Q0	ALL	87	+ 97.72
Oracle	Q1	-	100	-
Cassandra	Q1	ONE	46	- 54
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Cassandra	Q3	ONE	67	- 36.19
Cassandra	Q3	QUORUM	341	+224.77
Cassandra	Q3	ALL	289	+ 175.24
Oracle	Q4	-	77	-
Cassandra	Q4	ONE	78	+ 1.29
Cassandra	Q4	QUORUM	91	+ 18.19
Cassandra	Q4	ALL	81	+ 5.19

Table 2 Query Latency in Milliseconds with the best Performance marked green (grouped by Database, Query and Consistency Level)

Query Performance

Metric:

→ Latency (in ms)

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Cassandra	Q0	ALL	87	+ 97.72
Cassandra	Q1	ONE	46	- 54
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Cassandra	Q3	ONE	67	- 36.19
Cassandra	Q3	QUORUM	341	+224.77
Cassandra	Q3	ALL	289	+ 175.24
Oracle	Q4	-	77	-
Cassandra	Q4	ONE	78	+ 1.29
Cassandra	Q4	QUORUM	91	+ 18.19
Cassandra	Q4	ALL	81	+ 5.19

Table 2 Query Latency in Milliseconds with the best Performance marked green (grouped by Database, Query and Consistency Level)

Query Performance

Metric:

→ Latency (in ms)

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Cassandra	Q0	ALL	87	+ 97.72
Oracle	Q1	-	100	-
Cassandra	Q1	ONE	46	- 54
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Cassandra	Q3	ONE	67	- 36.19
Cassandra	Q3	QUORUM	341	+224.77
Cassandra	Q3	ALL	289	+ 175.24
Oracle	Q4	-	77	-
Cassandra	Q4	ONE	78	+ 1.29
Cassandra	Q4	QUORUM	91	+ 18.19
Cassandra	Q4	ALL	81	+ 5.19

Table 2 Query Latency in Milliseconds with the best Performance marked green (grouped by Database, Query and Consistency Level)

Query Performance

Metric:

→ Latency (in ms)

System	Query	Consistency Level (CL)	Latency (in ms)	Increase (in %)
Oracle	Q0	-	44	-
Cassandra	Q0	ONE	63	+ 43.18
Cassandra	Q0	QUORUM	97	+ 120.45
Cassandra	Q0	ALL	87	+ 97.72
Oracle	Q1	-	100	-
Cassandra	Q1	ONE	46	- 54
Cassandra	Q1	QUORUM	127	+ 27
Cassandra	Q1	ALL	111	+ 11
Oracle	Q2	-	123	-
Cassandra	Q2	ONE	97	- 21.14
Cassandra	Q2	QUORUM	184	+ 49.59
Cassandra	Q2	ALL	147	+ 19.51
Oracle	Q3	-	105	-
Cassandra	Q3	ONE	67	- 36.19
Cassandra	Q3	QUORUM	341	+224.77
Cassandra	Q3	ALL	289	+ 175.24
			77	-
			78	+ 1.29
			91	+ 18.19
			81	+ 5.19

he best Performance marked green (grouped by

Under the following conditions, performance using ALL is worse than QUORUM:

- The data consists of thousands of rows or more.
- One node is slower than others.
- A particularly slow node was not selected to be part of the quorum.

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useTracingPerf.html

Evaluation: Limitations & Considerations

- **Small Data Set:**
  32MB of raw data / 214,049 data points
- **Data Variety:**
  Only Text data with Geolocation
- **Limited Metrics:**
  Only Time Measurements
- **No Testing Scalability**
- **Not a real distributed Production Environment**
  Where Cassandra might excel over Oracle

6. Future Outlook: NoSQL Techniques

Future Outlook: NoSQL Techniques

- **Data Migrations:** Predictive Migration
- **Replication Strategy:** Asynchronous (lazy) migration replication
- Changing to **'NetworkTopologyStrategy'** Replication
  - Additional Considerations about **Memory Settings**
- **Multi Master Model**
- **Local** Secondary Indexing
- **Early** Materialization

6. Conclusion

Conclusion

Experiments:

- ➔ **Cassandra** outperforms **Oracle** in query performance at **Consistency Level ONE** in most key queries (Q1,Q2,Q3)
- ➔ At higher **Consistency Levels,** Oracle outperforms Cassandra at all key queries
    - ➔ Enforcing higher **Consistency** in Cassandra comes at a greater latency penalty
- ➔ **However**: Findings might change in real use cases **in production environment (with huge amount of data)**

References

[1] Abadi, D. (2012). Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story. Computer, 45(2):37–42.

[2] Apache (2023). Guarantees | Apache Cassandra Documentation.

[3] Bundesamt für Strassen (ASTRA) (2023). Verordnung über das informationssystem strassenverkehrsunfälle. Art. 9.

[4] Corbellini, A., Mateos, C., Zunino, A., Godoy, D., and Schiaffino, S. (2017). Persisting big-data: The nosql landscape. Information Systems, 63:1–23.

[5] DataStax (2020). Planning and testing apache cassandra deployments. last accessed: 28.06.2023.

[6] DataStax (2022). How consistency affects performance | CQL for Cassandra 3.0.

[7] Fachstelle für Open Government Data (2023a). Datenportal BS.

[8] Fachstelle für Open Government Data (2023b). Welcome to Smart City Basel.

[9] GEO.ADMIN.CH (2023). Verfügbare datensätze. last accessed: 28.06.2023.

[10] Hewage, T., Halgamuge, M., Syed, A., and Ekici, G. (2018). Review: Big data techniques of google, amazon, facebook and twitter. Journal of Communications, 13:94–100.

[11] Huang, X., Wang, J., Qiao, J., Zheng, L., Zhang, J., and Wong, R. K. (2017). Performance and Replica Consistency Simulation for Quorum-Based NoSQL System Cassandra. In van der Aalst, W. and Best, E., editors, Application and Theory of Petri Nets and Concurrency, Lecture Notes in Computer Science, pages 78–98, Cham. Springer International Publishing.

[12] IBM (2023). IBM Documentation Apache Cassandra.

[13] Kuhlenkamp, J., Klems, M., and Röss, O. (2014). Benchmarking scalability and elasticity of distributed database systems. Proceedings of the VLDB Endowment, 7(12):1219–1230.

[14] Lakshman, A. and Malik, P. (2010). Cassandra: a decentralized structured storage system. ACM SIGOPS Operating Systems Review, 44(2):35–40.

[15] Lynch, J. and Synder, J. (2018). Cassandra availability with virtual nodes.

[16] Meenakshi, J. (2022). Data Reprocessing Pipeline in Asset Management Platform @Netflix.

[17] Ploetz, A. (2023). Two Reasons Why Apache Cassandra Is the Database for Real-Time Applications.

References 12

[18] Roy-Hubara, N. and Sturm, A. (2020). Design methods for the new database era: a systematic literature review. Software and Systems Modeling, 19(2):297–312.

[19] solid IT gmbh (2023). Db-engines ranking. last accessed: 28.06.2023.

[20] Zafar, R., Yafi, E., Zuhairi, M. F., and Dao, H. (2016). Big data: The nosql and rdbms review. In 2016 International Conference on Information and Communication Technology (ICICTM), pages 120–126.

THANKS FOR LISTENING!

Any questions?