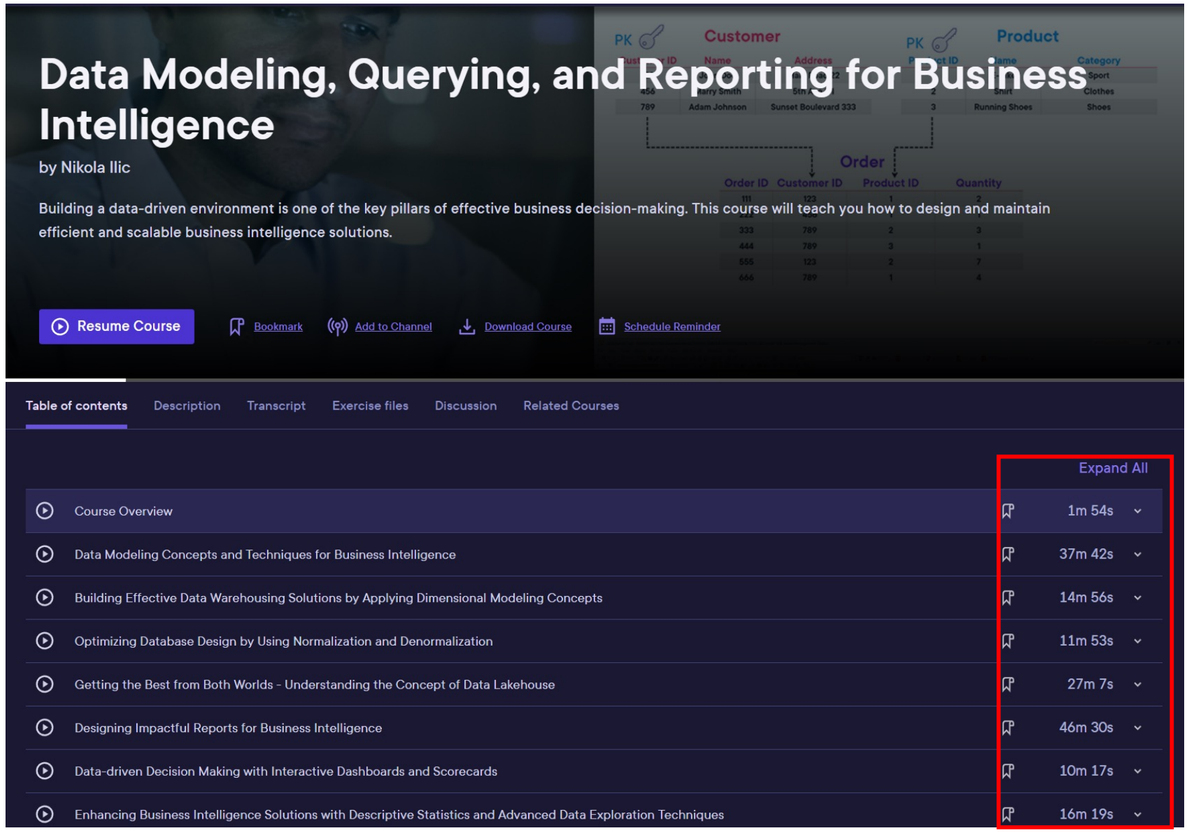
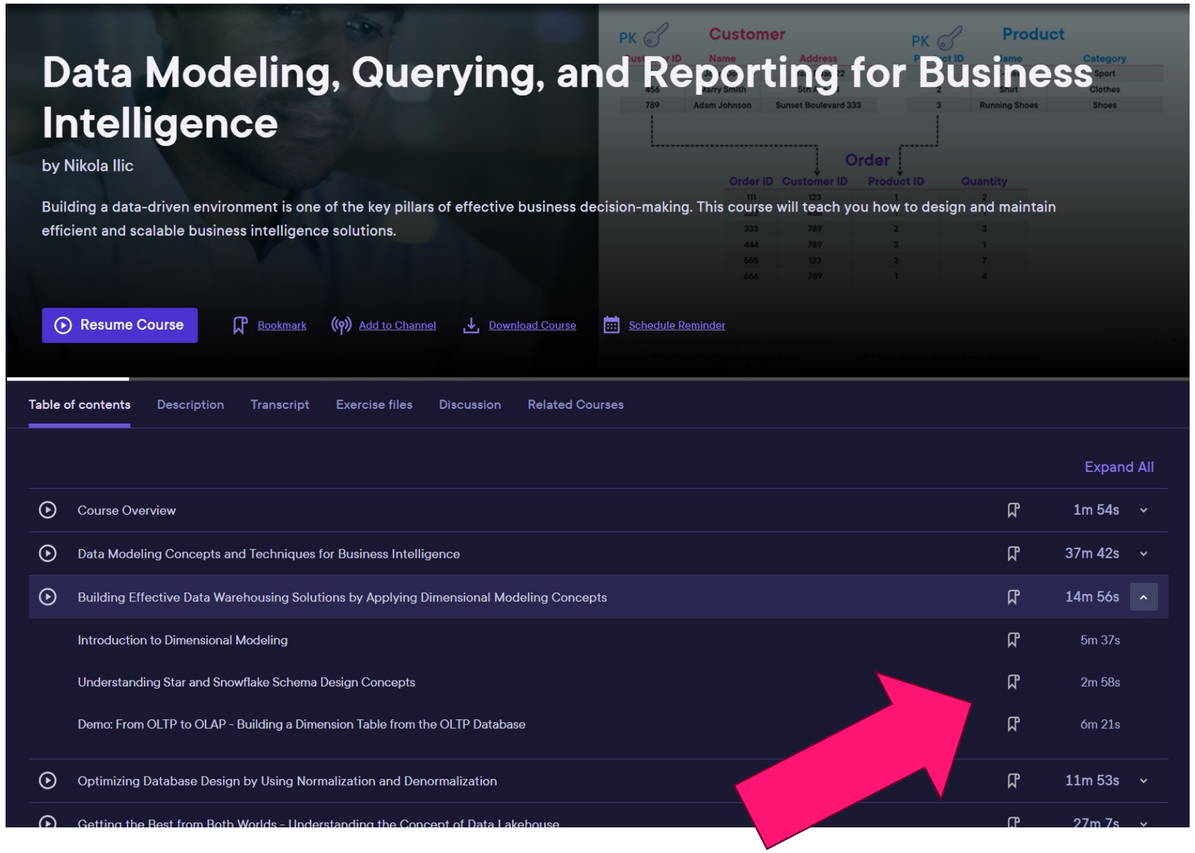
Understanding Indexing and Statistics

Nikola Ilic
Data Mozart

@DataMozart | www.data-mozart.com


Data Modeling, Querying, and Reporting for Business Intelligence
by Nikola Ilic
Building a data-driven environment is one of the key pillars of effective business decision-making. This course will teach you how to design and maintain efficient and scalable business intelligence solutions.
Resume Course
Bookmark
Add to Channel
Download Course
Schedule Reminder
Table of contents
Description
Transcript
Exercise files
Discussion
Related Courses
Expand All
Course Overview 1m 54s
Data Modeling Concepts and Techniques for Business Intelligence 37m 42s
Building Effective Data Warehousing Solutions by Applying Dimensional Modeling Concepts 14m 56s
Optimizing Database Design by Using Normalization and Denormalization 11m 53s
Getting the Best from Both Worlds - Understanding the Concept of Data Lakehouse 27m 7s
Designing Impactful Reports for Business Intelligence 46m 30s
Data-driven Decision Making with Interactive Dashboards and Scorecards 10m 17s
Enhancing Business Intelligence Solutions with Descriptive Statistics and Advanced Data Exploration Techniques 16m 19s


Data Modeling, Querying, and Reporting for Business Intelligence
by Nikola Ilic
Building a data-driven environment is one of the key pillars of effective business decision-making. This course will teach you how to design and maintain efficient and scalable business intelligence solutions.
Resume Course
Bookmark
Add to Channel
Download Course
Schedule Reminder
Table of contents
Description
Transcript
Exercise files
Discussion
Related Courses
Expand All
Course Overview 1m 54s
Data Modeling Concepts and Techniques for Business Intelligence 37m 42s
Building Effective Data Warehousing Solutions by Applying Dimensional Modeling Concepts 14m 56s
Introduction to Dimensional Modeling 5m 37s
Understanding Star and Snowflake Schema Design Concepts 2m 58s
Demo: From OLTP to OLAP - Building a Dimension Table from the OLTP Database 6m 21s
Optimizing Database Design by Using Normalization and Denormalization 11m 53s

Overview

Achieve astonishing improvements

Index types

- Non-clustered vs clustered
- Rowstore vs Columnstore

Index design recommended practices

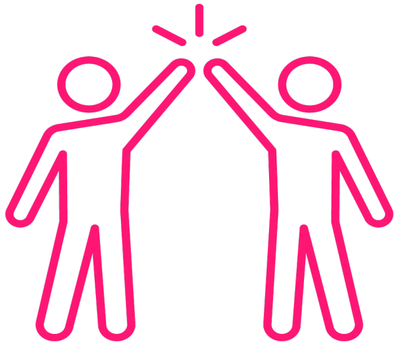
Statistics = “Secret” ingredient for the optimal execution plan

As few indexes as possible,
to be referenced by *as*
many queries as possible...

Sounds easy, right?

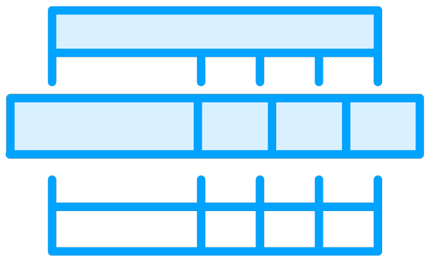
Finding the balance
Hard in real life

“One to rule them all”
There is no such approach

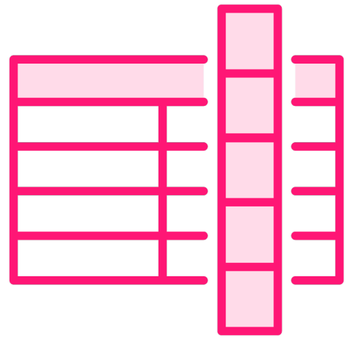


Mastery on its own
Can make a whole world of difference

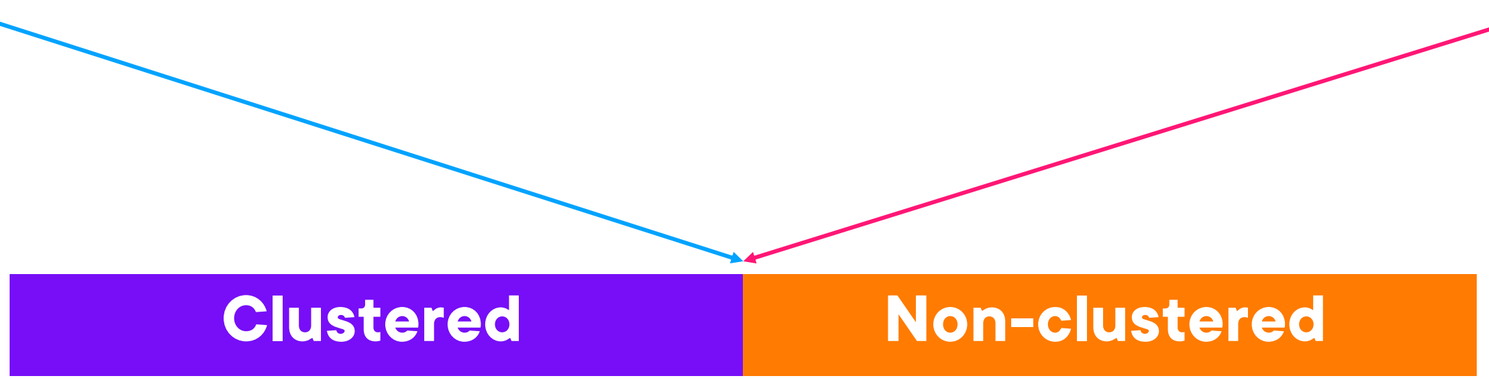
Index Types



Rowstore



Columnstore



Heap

	Product	Customer	Country	Date	Sales Amount
Row 1	Ball	John Doe	USA	2023-01-01	100
Row 2	T-Shirt	John Doe	USA	2023-01-02	200
Row 3	Socks	Maria Adams	UK	2023-01-01	300
Row 4	Socks	Antonio Grant	USA	2023-01-03	100
Row 5	T-Shirt	Maria Adams	UK	2023-01-02	500
Row 6	Socks	John Doe	USA	2023-01-05	200

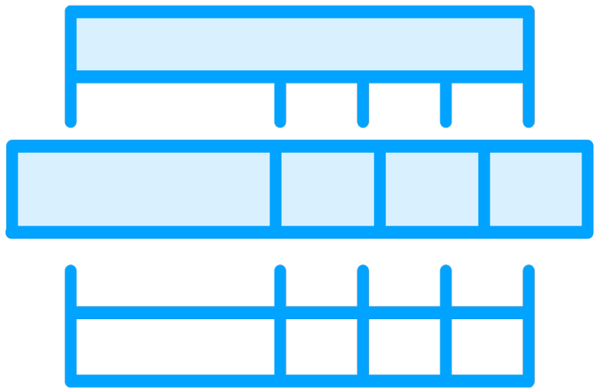
Heap

	Product	Customer	Country	Date	Sales Amount
Row 1	Ball	John Doe	USA	2023-01-01	100
Row 2	T-Shirt	John Doe	USA	2023-01-02	200
Row 3	Socks	Maria Adams	UK	2023-01-01	300
Row 4	Socks	Antonio Grant	USA	2023-01-03	100
Row 5	T-Shirt	Maria Adams	UK	2023-01-02	500
Row 6	Socks	John Doe	USA	2023-01-05	200

Clustered Rowstore Index

	Key	Product	Customer	Country	Date	Sales Amount
Row 1	1	Ball	John Doe	USA	2023-01-01	100
Row 2	2	T-Shirt	John Doe	USA	2023-01-02	200
Row 3	3	Socks	Maria Adams	UK	2023-01-01	300
Row 4	4	Socks	Antonio Grant	USA	2023-01-03	100
Row 5	5	T-Shirt	Maria Adams	UK	2023-01-02	500
Row 6	6	Socks	John Doe	USA	2023-01-05	200

Non-Clustered Index



Doesn’t sort the physical data within the table

Stored in a different location!

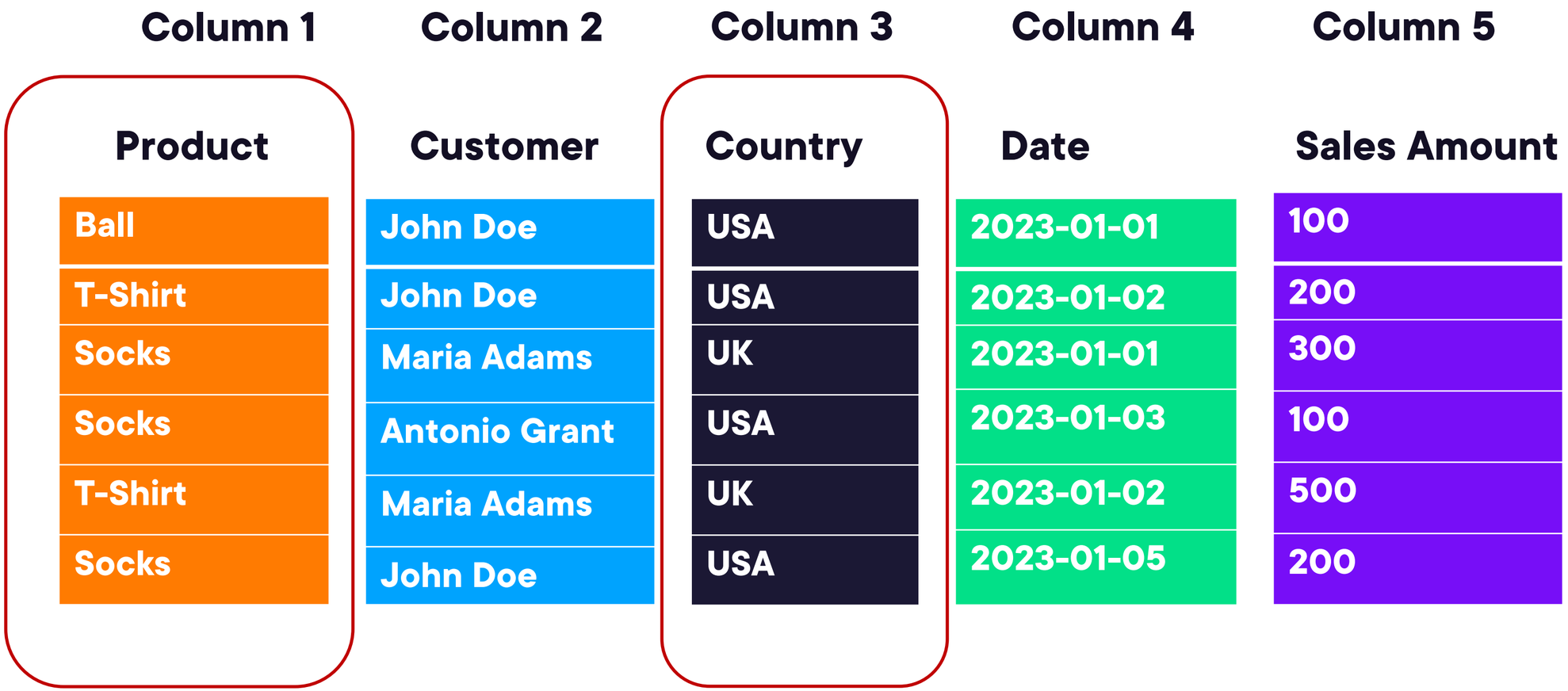
Multiple non-clustered indexes on the table

Require additional storage

Usually slower than Clustered

- Include additional Lookup step

Columnstore Index



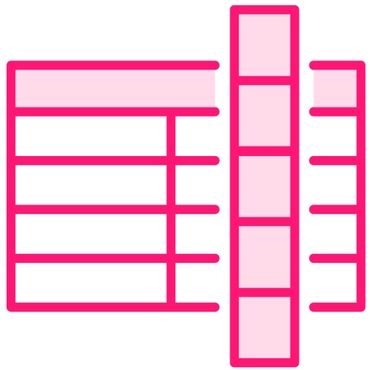
Columnstore Index

Product	Customer	Country	Date	Sales Amount
Ball	John Doe	USA	2023-01-01	100
T-Shirt	John Doe	USA	2023-01-02	200
Socks	Maria Adams	UK	2023-01-01	300
Socks	Antonio Grant	USA	2023-01-03	100
T-Shirt	Maria Adams	UK	2023-01-02	500
Socks	John Doe	USA	2023-01-05	200

✓ Suitable for analytic (OLAP) workloads

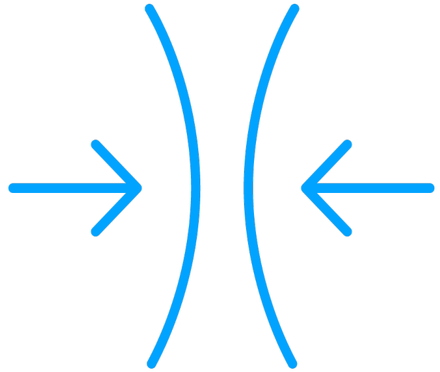
✓ Useful for large tables (100K+ rows)

Columnstore Index



Only columns needed by query

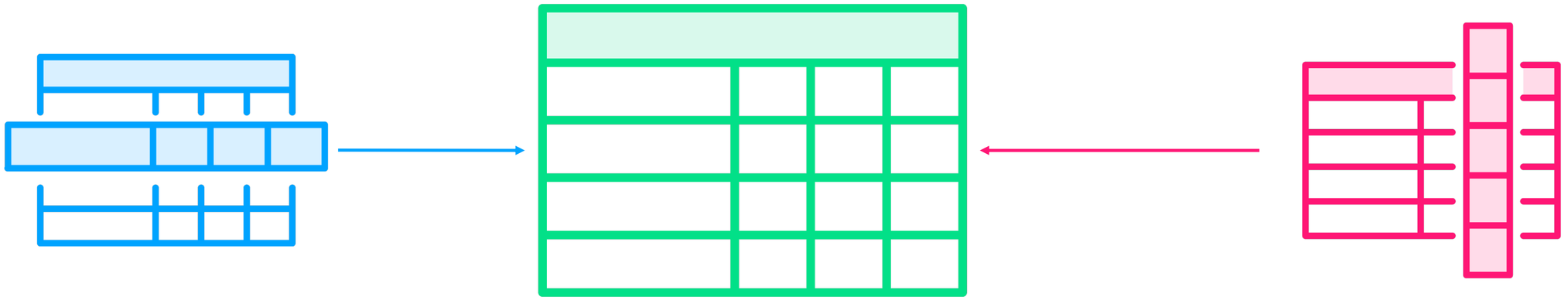
Unlike in rowstore indexes



Data compression

Data is by default more optimally compressed

Combining Index Types



Rowstore

Columnstore

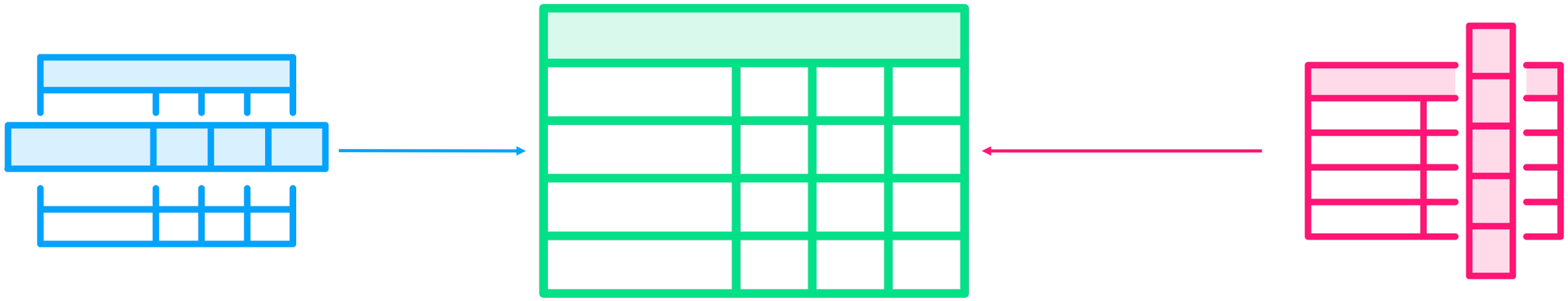
Clustered

Non-clustered 1

Non-clustered 2

Non-clustered 3

Combining Index Types



Rowstore

Columnstore

Non-clustered 1

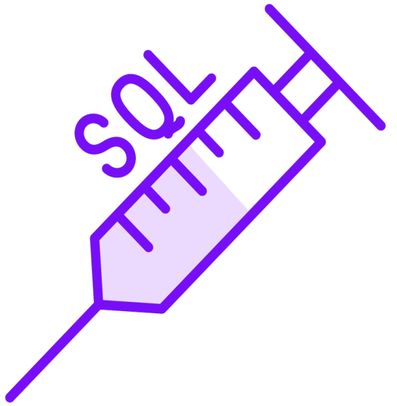
Non-clustered 2

Non-clustered 3

Clustered

Index “Price”

Memory space



Processing time
For INSERT, UPDATE, DELETE statements

Index Design Recommended Practices

Defining Impactful Indexing Strategy

Workload type

Rowstore for OLTP, columnstore for OLAP

Filtering conditions

Columns used in WHERE, HAVING, JOIN

Column data type

Use integers over text when possible

Data selectivity

Create index on a more selective column

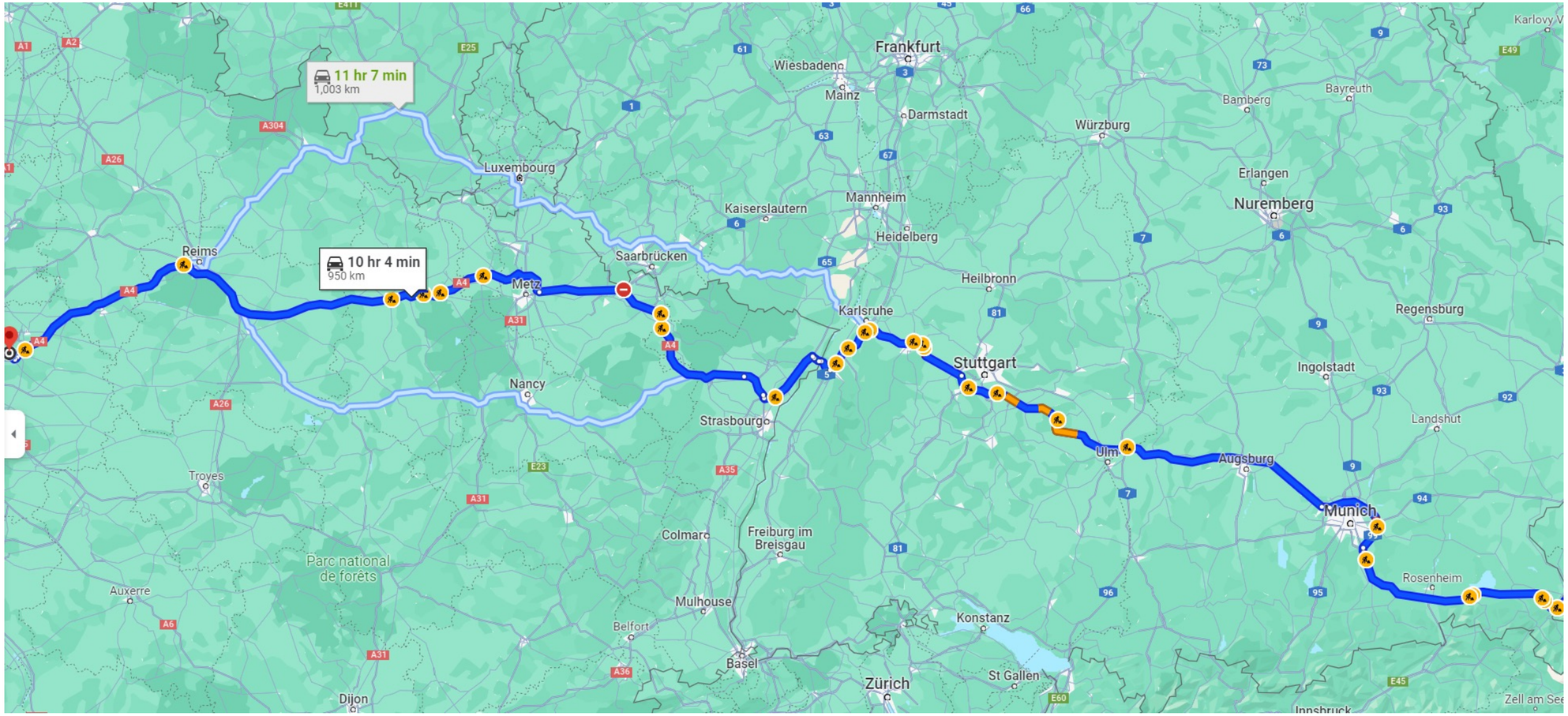
Column order matters!

Demo

Design indexes for performance

- Filtering conditions
- Data selectivity
- Column order

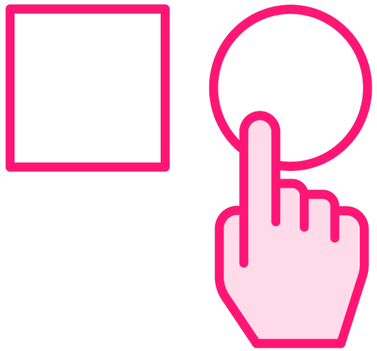
Understanding Statistics in SQL Server



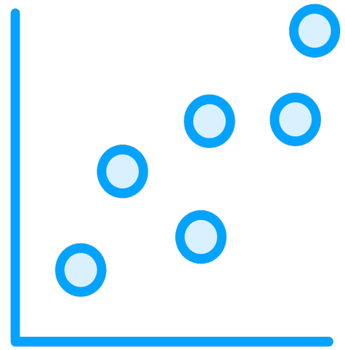
Salzburg -> Paris

Store and maintain information about the data distribution in columns

Statistics in a Nutshell



SELECT doesn't need statistics,
WHERE and JOIN must have it



Automatic statistics
Columns used in joins and filters,
and columns used in the index

Statistics on Indexed vs Non-Indexed Columns

Indexed columns

- ✓ Automatically created for every rowstore index
- ✓ Can’t be changed!
- ✓ Auto-update statistics UNLESS you disable it (not recommended)

Non-indexed columns

- ✓ Automatically created for all columns used in joins and filtering (WHERE/HAVING)
- ✓ Consider disabling only for multiple ad-hoc queries executed only once

Demo

Help SQL Server to generate the optimal execution plan

- Up-to-date vs outdated statistics
- Query behavior for indexed and non-indexed columns

Summary

Power of indexes

- Quickly find and retrieve the data

As few indexes as possible to be referenced by as many queries as possible

Rowstore vs columnstore indexes

- Columnstore for analytic workloads
- Rowstore for transactional workloads

Benefits of up-to-date statistics

- Enable query optimizer to come up with the optimal execution plan

Up Next:

Optimizing Ad-Hoc Expressions and Parameterization