Databases Overview

Relational DBs and Non-Relational DBs

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg/

Have a Question?

[sli.do](sli.do)

#QA-Auto-Backend

Table of Contents

Databases Overview

Database Concepts

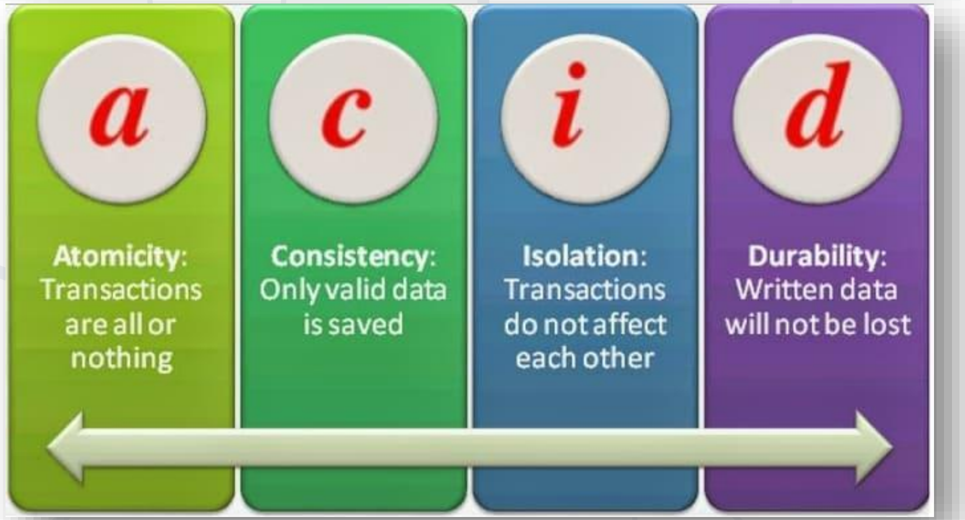
Databases - Definition and Purposes

Software University

- Database is a **structured collection** of data designed to be easily accessed, managed, and updated.
- Databases are central to modern software systems, serving as **repositories for important information**.
- **Purposes:**
  - Databases are responsible for **storing**, **organizing**, and **managing** data.
  - They provide **data persistence** and enable **data sharing** among different parts of an application.

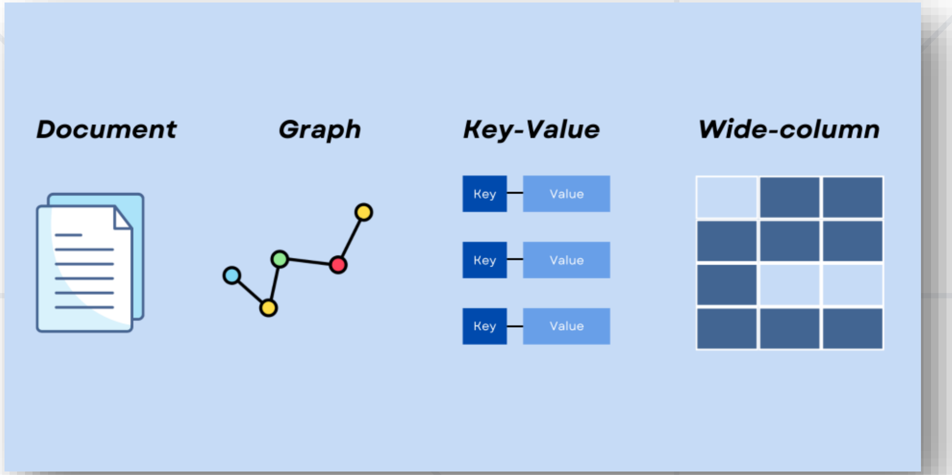
Relational Databases

- **Relational databases** (SQL Databases)
  - **Structured data** stores, where data is **organized into tables** with predefined **schemas**
  - Use the **SQL**-based query language
  - **Transactions** requiring data integrity
  - **ACID** compliance for data consistency
  - Complex relationships between data


a
c
i
d
Atomicity: Transactions are all or nothing
Consistency: Only valid data is saved
Isolation: Transactions do not affect each other
Durability: Written data will not be lost

NoSQL Databases

- **NoSQL ("not only SQL") databases**
  - A category of relational databases, designed to handle **unstructured** or **semi-structured data**.
- **Characteristics:**
  - Flexible Data Models
  - Horizontal Scalability
  - High Performance



Roles of Databases in Back-End Testing

Ensuring Data Integrity and Reliability

Data Validation

- Databases play a critical role in **data validation** during back-end testing
- **Data Validation Testing**
  - Validating the correctness of data storage and retrieval operations, ensuring that the data remains accurate and consistent throughout the testing process
- **Functional Testing**
  - Back-end functionality often relies on databases to process requests and generate responses

Data Integrity and Reliability

- **Performance Testing**
  - Assess database performance under **various workloads** to ensure that the back-end can handle user requests efficiently and maintain responsiveness
- **Security** and **Access Control**
  - Evaluate user access controls, encryption mechanisms, and data security measures
- **Scalability Testing**
  - System's ability to scale effectively as **data volumes** and **user loads** increase

Databases in Back-End Testing

- **Data Validation**
  - Verify data **accuracy** and **consistency**
- **Functional Testing**
  - Ensure **correct** back-end **functionality**
- **Performance Testing**
  - Assess database **performance**
- **Security and Access Control**
  - Protect **sensitive** data
- **Scalability Testing**
  - Evaluate system's **scalability**

SQL Server

Relational Databases

Download Clients & Servers

- Download **SQL Server Express** Edition from Microsoft

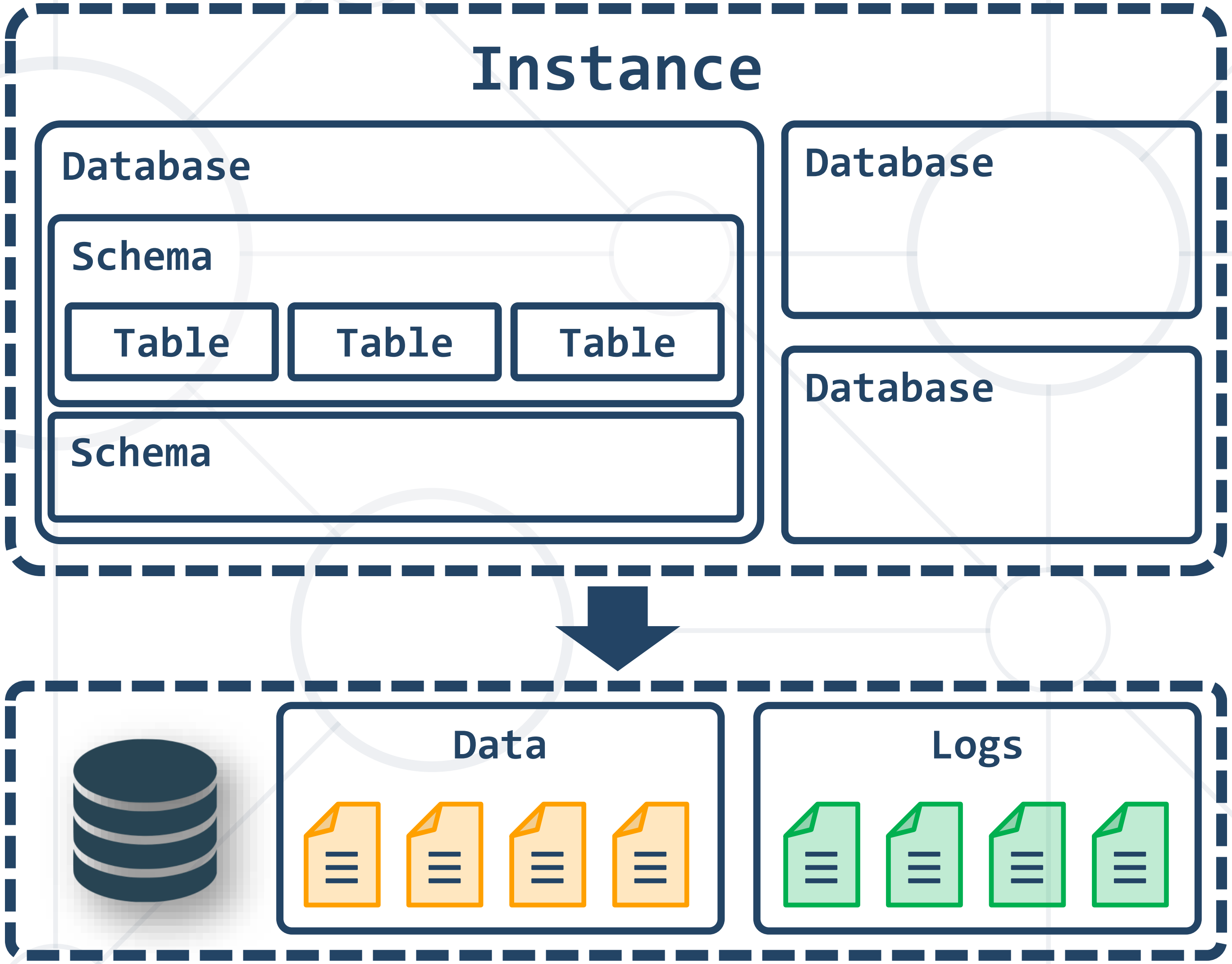
```
https://go.microsoft.com/fwlink/?linkid=866662
```

- Download SQL Server **Management Studio** separately

```
https://aka.ms/ssmsfullsetup
```

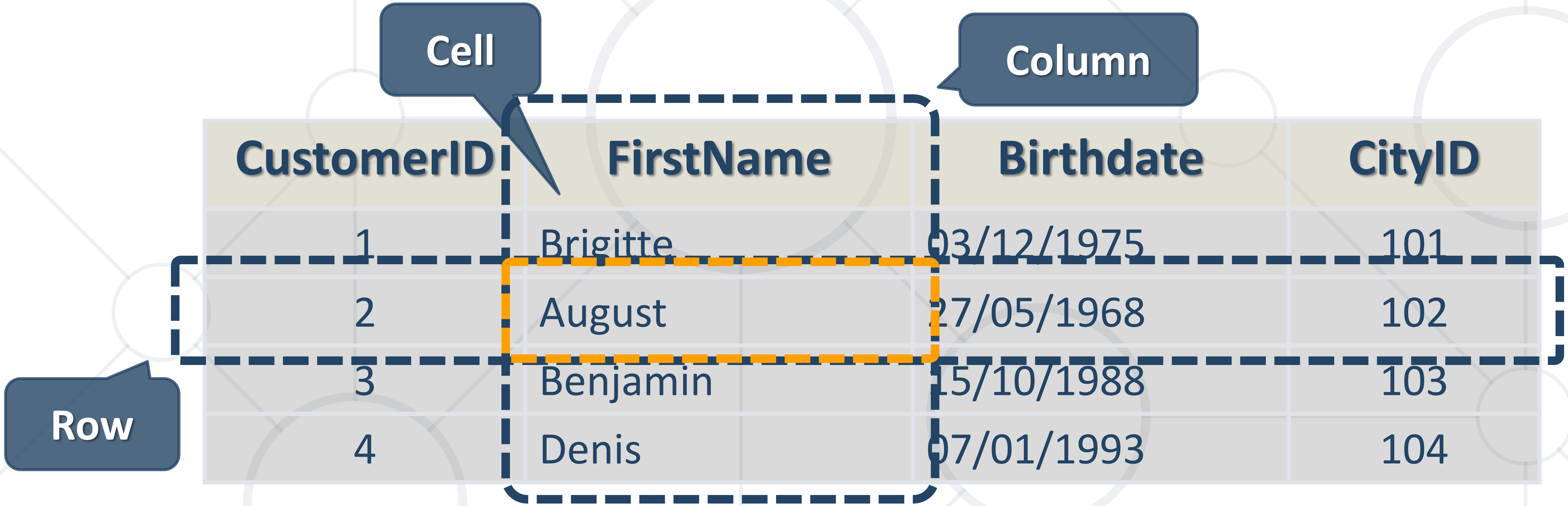
SQL Server Architecture

- Logical Storage
  - Instance
  - Database
  - Schema
  - Table
- Physical Storage
  - Data Files and Log files
  - Data Pages



Database Table Elements

- The table is the main **building block** of any database

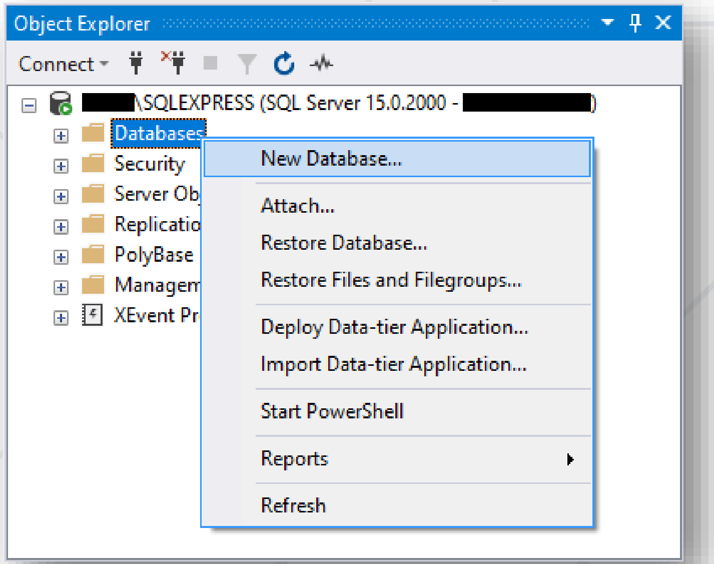


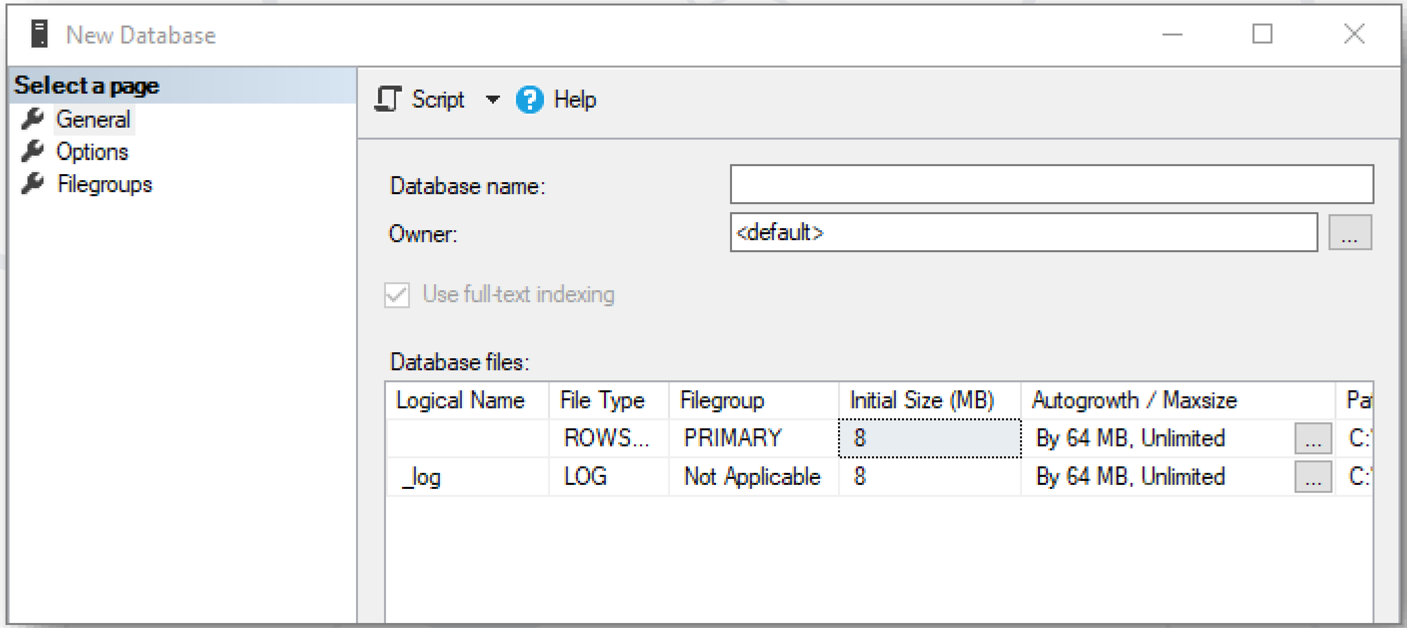
CustomerID	FirstName	Birthdate	CityID
1	Brigitte	03/12/1975	101
2	August	27/05/1968	102
3	Benjamin	15/10/1988	103
4	Denis	07/01/1993	104

- Each **row** is called a **record** or **entity**
- Columns (**fields**) define the **type** of data they contain

Creating a New Database

- Select **New Database** from the **context menu** under "**Databases**"





Database name:

Owner: <default>

✓ Use full-text indexing

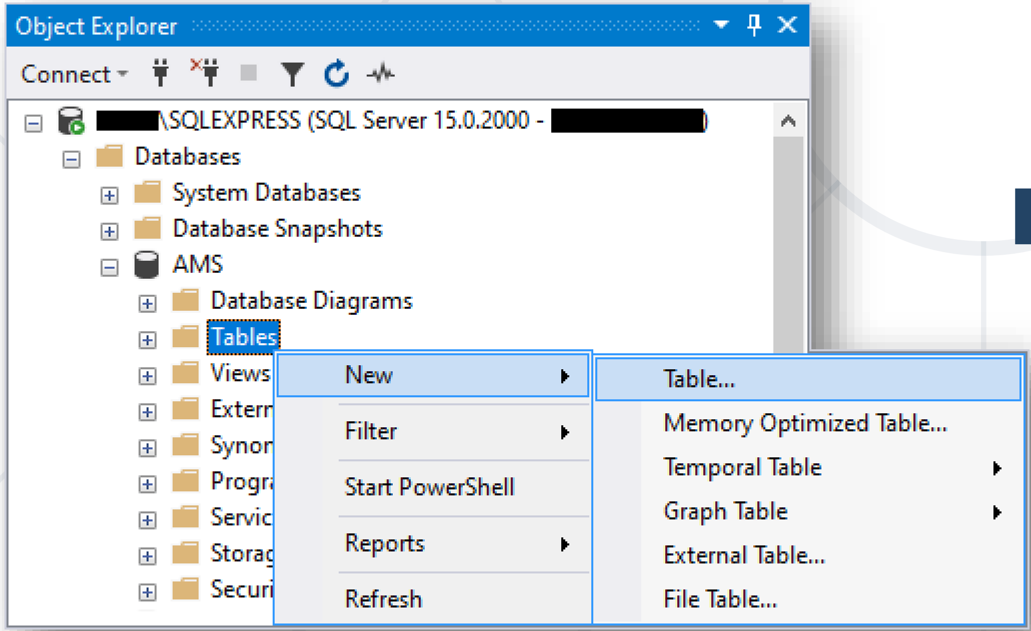
Database files:

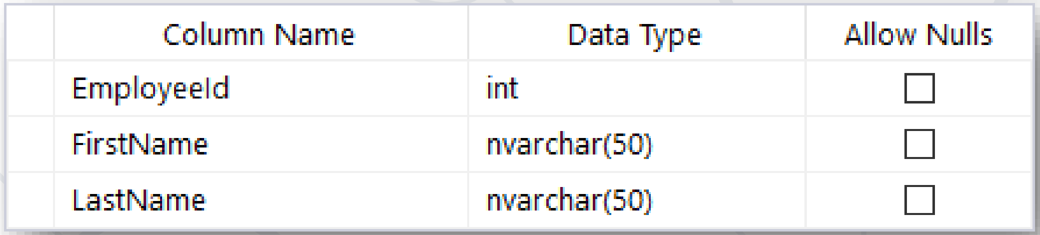
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize		Pa
	ROWS...	PRIMARY	8	By 64 MB, Unlimited	...	C:
_log	LOG	Not Applicable	8	By 64 MB, Unlimited	...	C:

- You may need to **Refresh [F5]** to see the results

Creating Tables

- Right-click from the **context menu** under "**New**" inside the desired database → "**Table**"



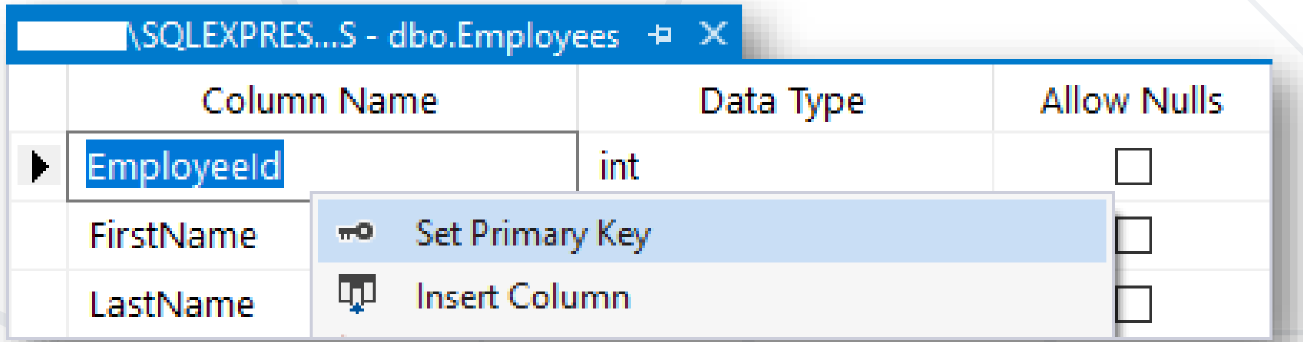


Column Name	Data Type	Allow Nulls
EmployeeId	int	☐
FirstName	nvarchar(50)	☐
LastName	nvarchar(50)	☐

- Table name can be set from its **Properties [F4]** or when it is **saved**

Creating Tables

- A **Primary Key** is used to uniquely identify and index records
- Setting **primary key** on a column:



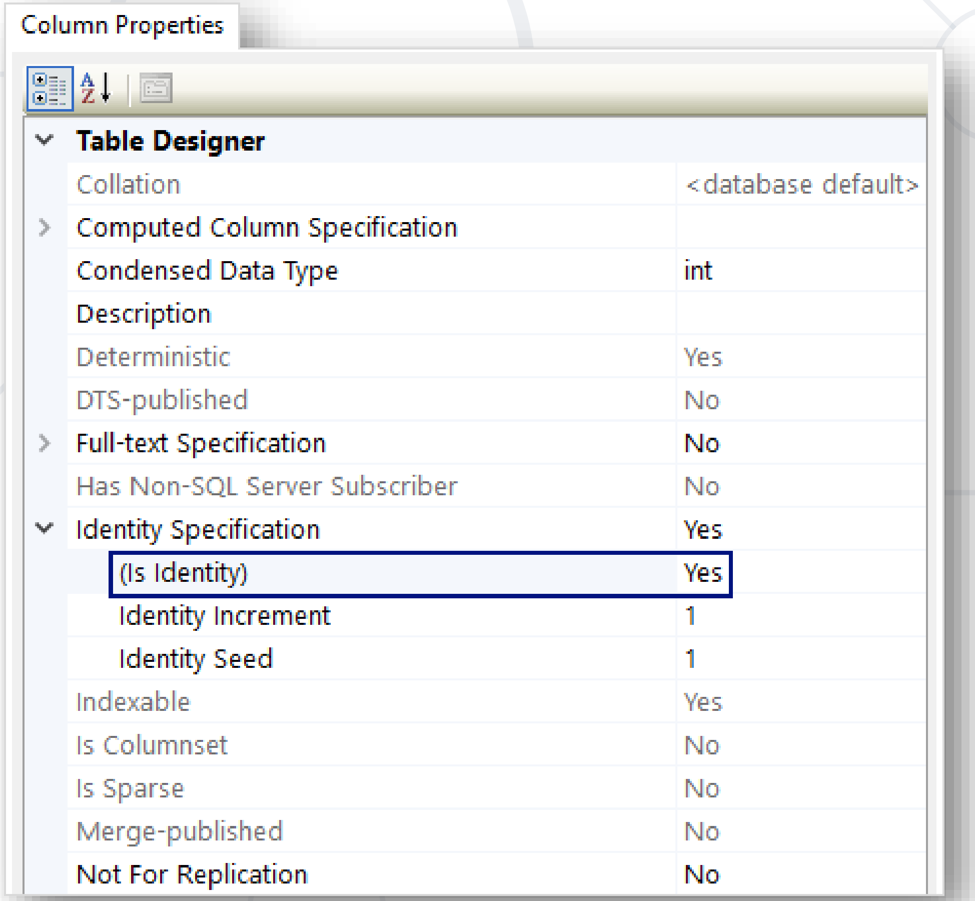
Column Name	Data Type	Allow Nulls
EmployeeId	int	☐
FirstName		☐
LastName		☐

Creating Tables

- **Identity** – The value in the column is automatically incremented when a new record is added
  - These values cannot be assigned manually
  - **Identity Seed** – the initial number (1 by default)
  - **Identity Increment** – how much each consecutive value is incremented

Creating Tables

- Setting an identity through the "`Column Properties`" window:

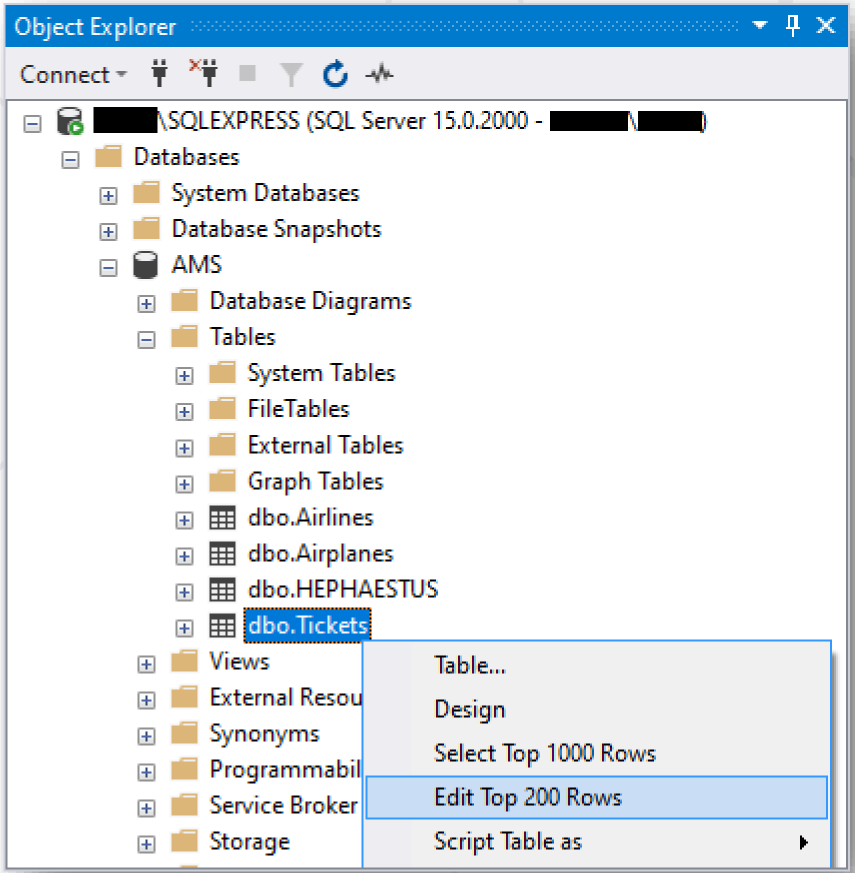


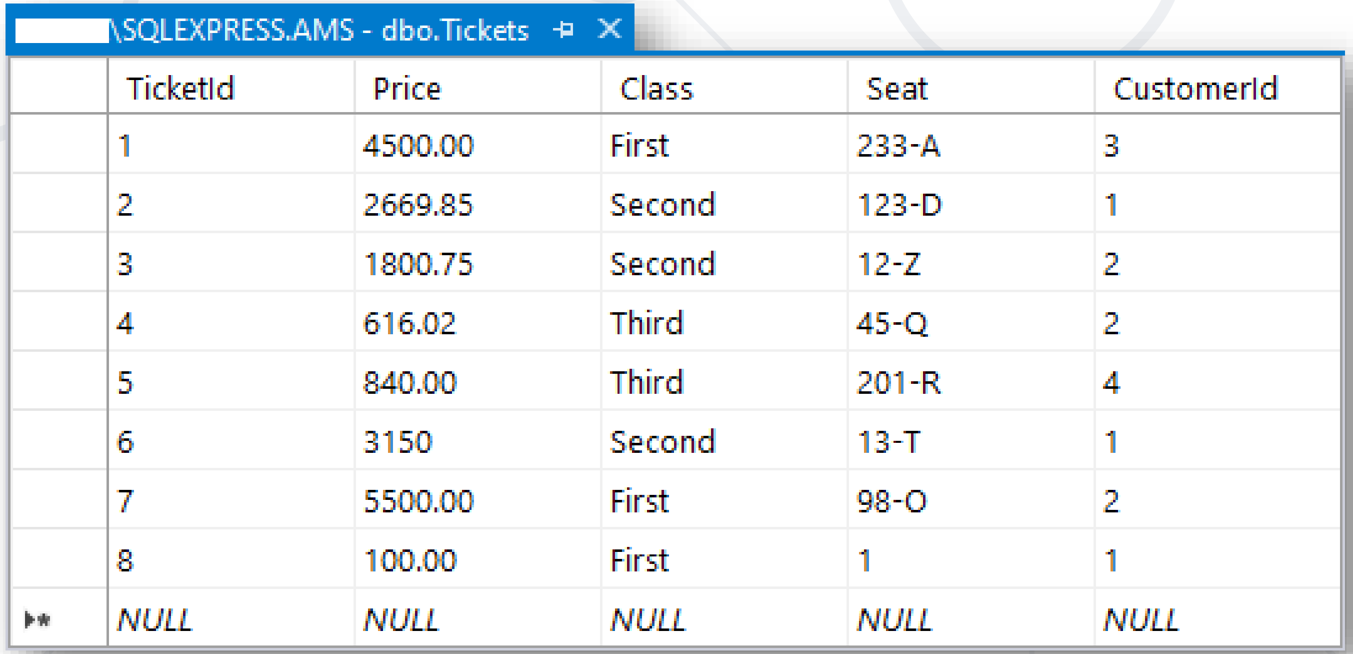
Column Properties

Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No

Storing and Retrieving Data

- We can **add**, **modify** and **read** records with Management Studio
- To insert or edit a record, click **`Edit`** from the context menu





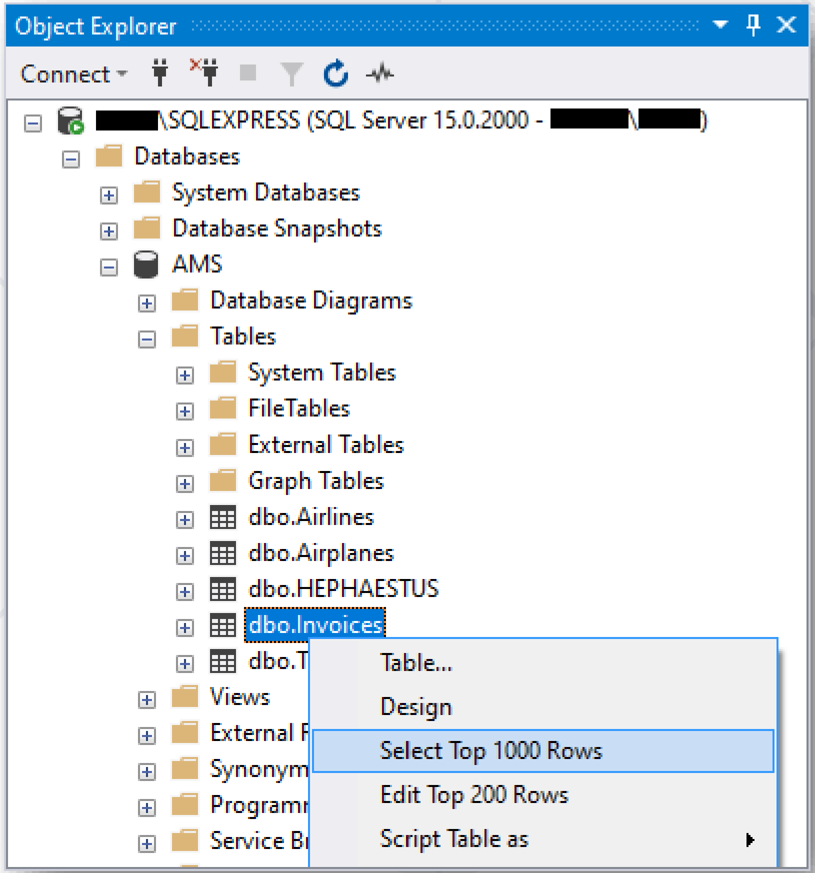
[redacted]\SQLEXPRESS.AMS - dbo.Tickets

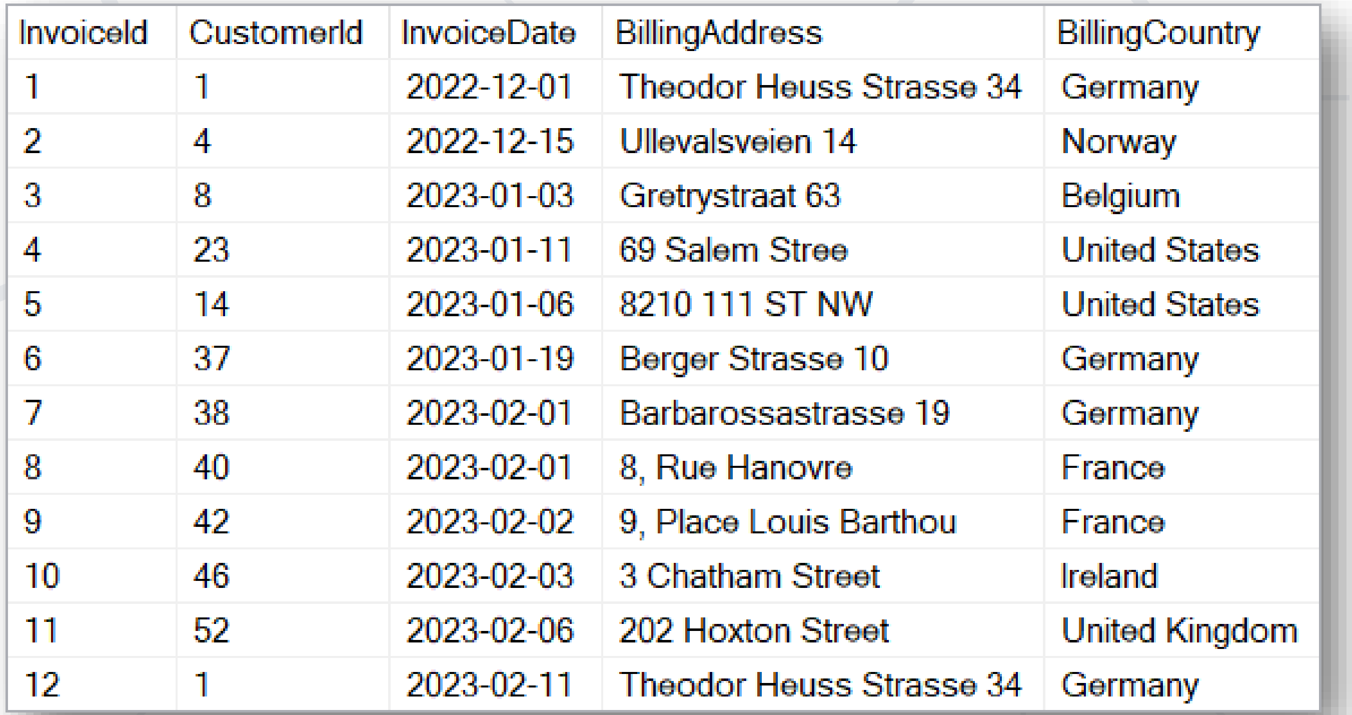
	TicketId	Price	Class	Seat	CustomerId
	1	4500.00	First	233-A	3
	2	2669.85	Second	123-D	1
	3	1800.75	Second	12-Z	2
	4	616.02	Third	45-Q	2
	5	840.00	Third	201-R	4
	6	3150	Second	13-T	1
	7	5500.00	First	98-O	2
	8	100.00	First	1	1
▸*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Enter data at the end to add a new row

Storing and Retrieving Data

- To retrieve records, click **Select** from the context menu



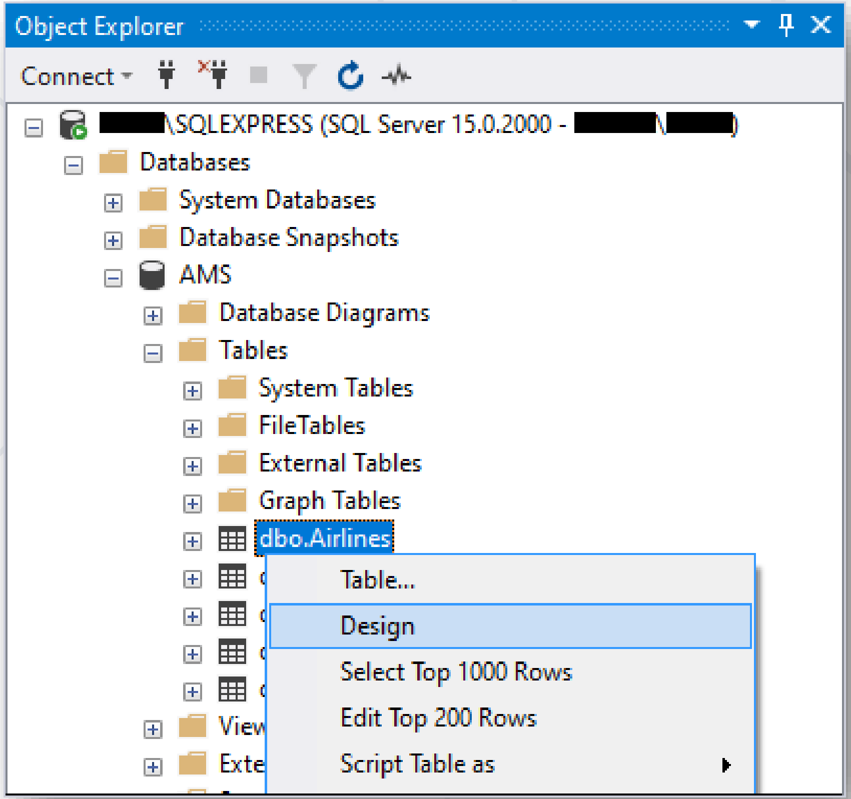


InvoiceId	CustomerId	InvoiceDate	BillingAddress	BillingCountry
1	1	2022-12-01	Theodor Heuss Strasse 34	Germany
2	4	2022-12-15	Ullevalsveien 14	Norway
3	8	2023-01-03	Gretrystraat 63	Belgium
4	23	2023-01-11	69 Salem Stree	United States
5	14	2023-01-06	8210 111 ST NW	United States
6	37	2023-01-19	Berger Strasse 10	Germany
7	38	2023-02-01	Barbarossastrasse 19	Germany
8	40	2023-02-01	8, Rue Hanovre	France
9	42	2023-02-02	9, Place Louis Barthou	France
10	46	2023-02-03	3 Chatham Street	Ireland
11	52	2023-02-06	202 Hoxton Street	United Kingdom
12	1	2023-02-11	Theodor Heuss Strasse 34	Germany

- The received information can be customized with **SQL queries**

Altering Tables

- You can change the properties of a table after its creation
- Select **Design** from the table's context menu

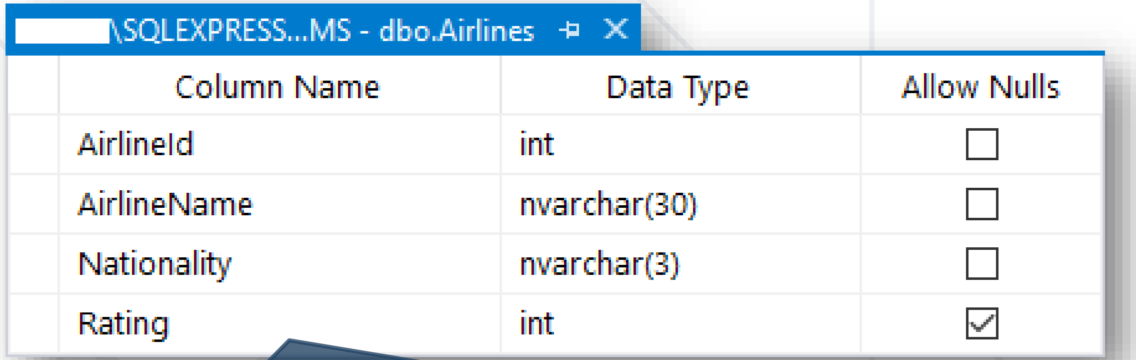


Object Explorer

Connect

[redacted]\SQLEXPRESS (SQL Server 15.0.2000 - [redacted]\[redacted])

- Databases
  - System Databases
  - Database Snapshots
  - AMS
    - Database Diagrams
    - Tables
      - System Tables
      - FileTables
      - External Tables
      - Graph Tables
      - dbo.Airlines
    - View
    - Exte



[redacted]\SQLEXPRESS...MS - dbo.Airlines

Column Name	Data Type	Allow Nulls
AirlineId	int	☐
AirlineName	nvarchar(30)	☐
Nationality	nvarchar(3)	☐
Rating	int	☑

Changes cannot conflict with existing rules!

Basic SQL Queries

Data Definition Using T-SQL

Structured Query Language

- To communicate with the Engine we use **SQL**
  - **Declarative** language
- Logically divided in four sections
  - **Data Definition** – describe the structure of our data
  - **Data Manipulation** – store and retrieve data
  - **Data Control** – define who can access the data
  - **Transaction Control** – bundle operations and allow rollback

SQL Queries

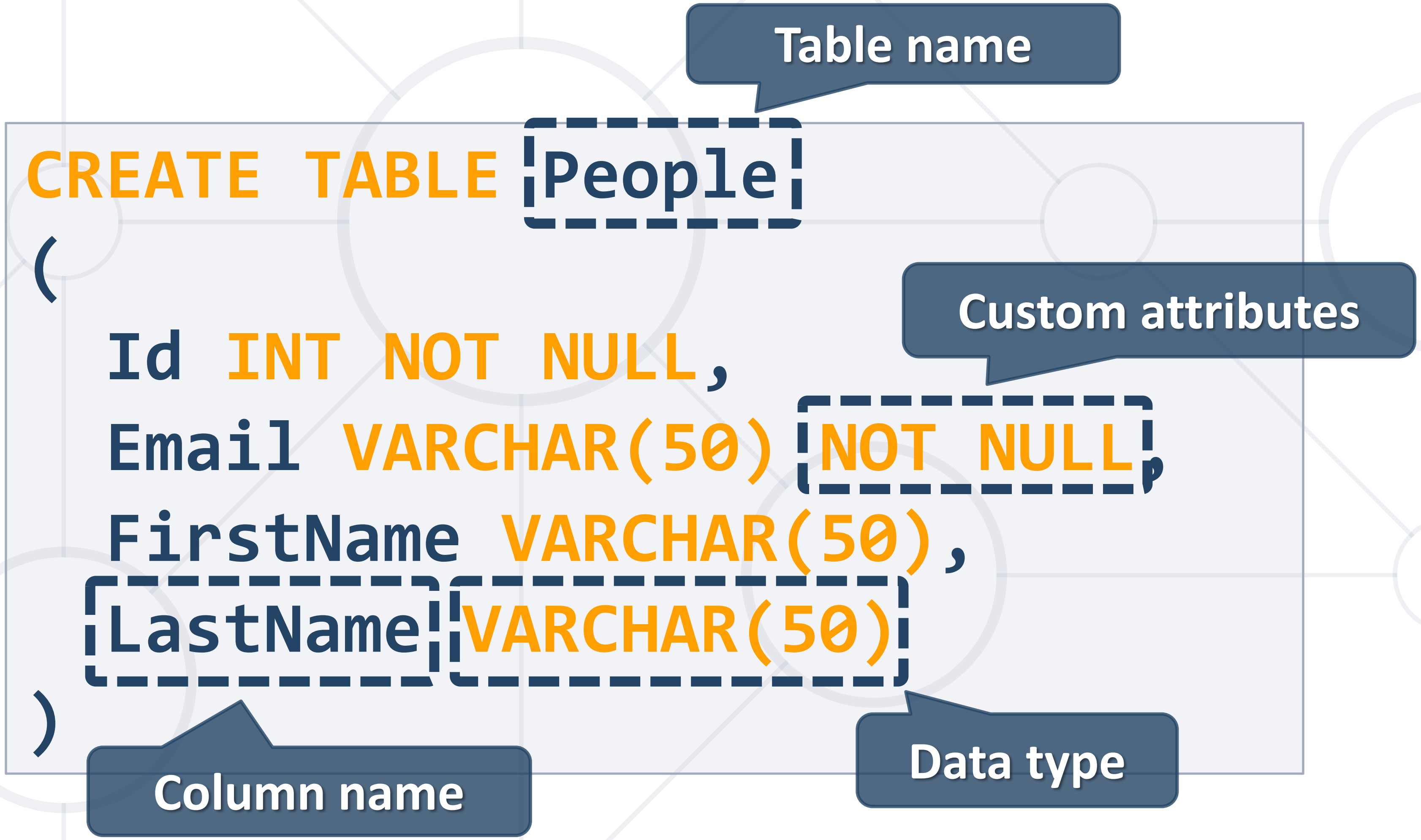
- We can communicate with the database engine using SQL
- Queries provide greater **control** and **flexibility**
- To create a database using SQL:

```sql
CREATE DATABASE Employees
```

Database name

- SQL keywords are traditionally **capitalized**

Table Creation in SQL



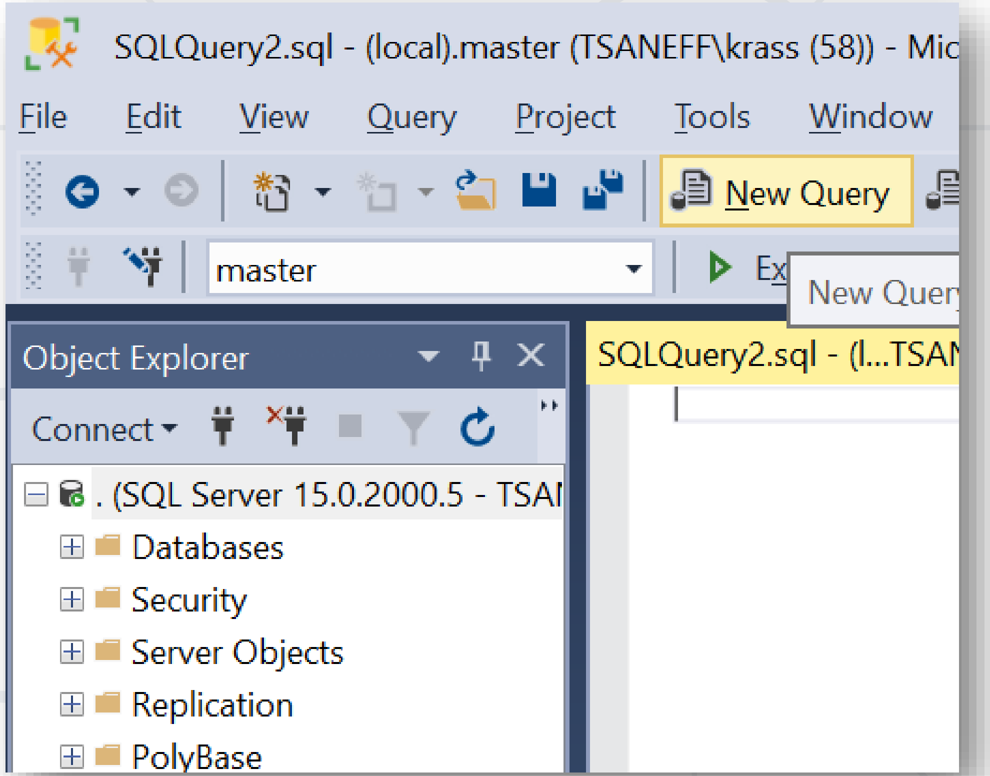
```
CREATE TABLE People
(
   Id INT NOT NULL,
   Email VARCHAR(50) NOT NULL,
   FirstName VARCHAR(50),
   LastName VARCHAR(50)
)
```

Import DB Script & Execute Queries

Working with Existing Data

Importing a DB Script in SSMS

- Open SQL Server Management Studio on your computer
- Connect to the SQL Server instance where you want to create the database
- Click on "**New Query**" to open a new query window
- Use the "**File**" menu or keyboard shortcut to open the database script file (*.sql) that you want to import



Retrieve Records in SQL

- To get all records from a table

```
SELECT * FROM Employees
```

- You can limit the number of rows and number of columns

Number of records

List of columns

```
SELECT TOP (5) FirstName, LastName
FROM Employees
```

What is T-SQL?

- **Structured Query Language**
  - Declarative language
  - Close to regular English

```
SELECT FirstName, LastName, JobTitle FROM Employees
```

  - Supports definition, manipulation and access control of records
- **Transact-SQL (T-SQL)** – SQL Server's version of SQL
  - Supports control flow (**if**-statements, **loops**)
  - Designed for writing **logic** inside the database

SQL – Examples

```
SELECT FirstName, LastName, JobTitle FROM Employees
```

```
SELECT * FROM Projects WHERE StartDate = '1/1/2006'
```

```
INSERT INTO Projects(Name, StartDate)
     VALUES ('Introduction to SQL Course', '1/1/2006')
```

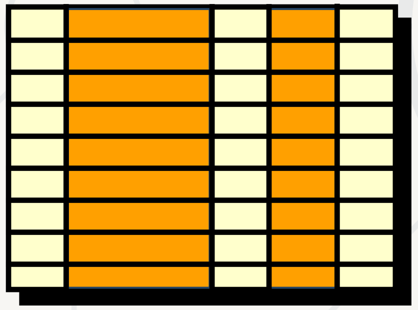
```
UPDATE Projects
   SET EndDate = '8/31/2006'
 WHERE StartDate = '1/1/2006'
```

```
DELETE FROM Projects
      WHERE StartDate = '1/1/2006'
```

Capabilities of SQL SELECT

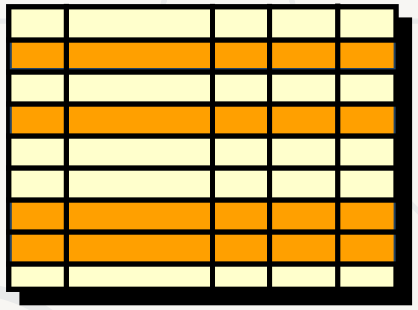
Projection

Take a subset of the columns



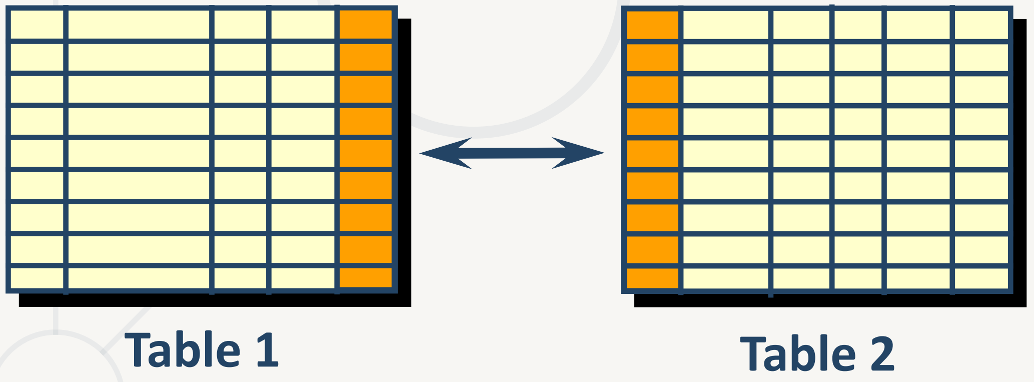
Selection

Take a subset of the rows



Join

Combine tables by some column



SELECT – Example

- Selecting **all** columns from the "Departments" table

```
SELECT * FROM Departments
```

DepartmentID	Name	ManagerID
1	Engineering	12
2	Tool design	4
3	Sales	273
…	…	…

- Selecting **specific** columns

```
SELECT DepartmentId, Name
  FROM Departments
```

DepartmentID	Name
1	Engineering
2	Tool design
3	Sales
…	…

Column Aliases

- Aliases rename a table or a column heading

Display Name

```
SELECT EmployeeID AS ID,
       FirstName,
       LastName
  FROM Employees
```

ID	FirstName	LastName
1	Guy	Gilbert
2	Kevin	Brown
…	…	…

- You can shorten fields or clarify abbreviations

```
SELECT c.Duration,
       c.ACG AS 'Access Control Gateway'
FROM Calls AS c
```

Concatenation Operator

- You can **concatenate** column names using the **+** operator
  - **String literals** are enclosed in **single quotes**
  - Column names containing **special symbols** use **brackets**

```
SELECT FirstName + ' ' + LastName AS [Full Name],
       EmployeeID AS [No.]
  FROM Employees
```

Full Name	No.
Guy Gilbert	1
Kevin Brown	2
…	…

NoSQL Databases

NoSQL (Non-Relational) Databases

- A **NoSQL** databases have dynamic schema for **unstructured** data
- Data may be stored in several ways:
  - **Document-oriented** (JSON store)
  - **Column-oriented** (table store)
  - **Graph-based**
  - **Key-value store**

NoSQL Databases

- **NoSQL databases** don't use tables
  - Instead, use **document collections** or **key-value pairs**
- More **scalable** and **high performance**
- Examples: **MongoDB**, **Cassandra**, **Redis**, etc.

```
{
  "_id": ObjectId("59d3fe7ed81452db0933a871"),
  "email": "peter@gmail.com",
  "age": 22
}
```

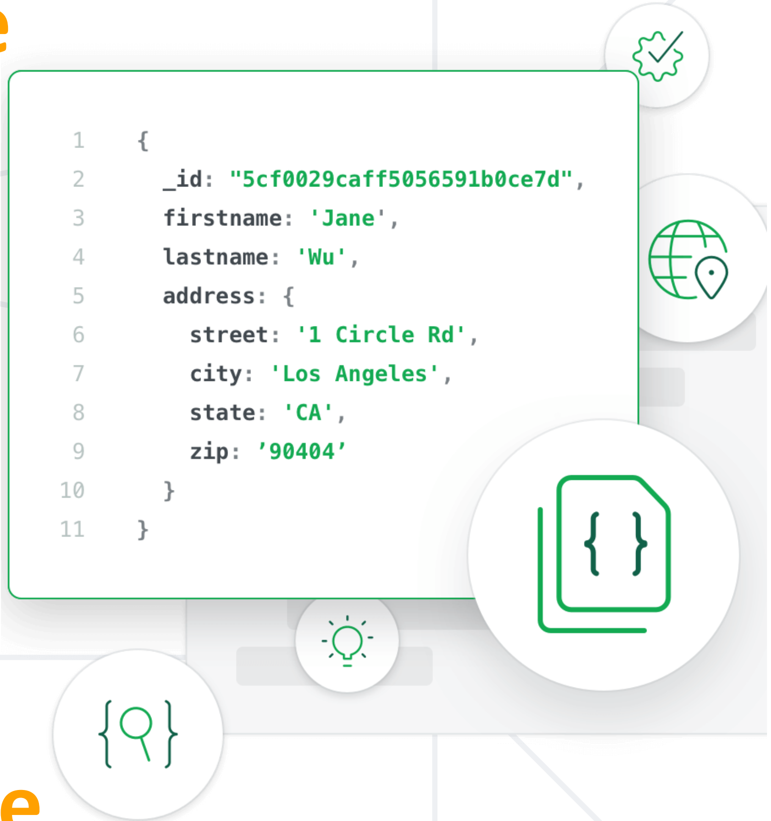
Example of JSON document in MongoDB

MongoDB Overview

Installation, Configuration, Startup

What is MongoDB?

- MongoDB is a **document database**
- It stores data in flexible, **BSON** documents
- The document model maps to the objects in the application code, making data easy to work with
- MongoDB is a **distributed database** at its core



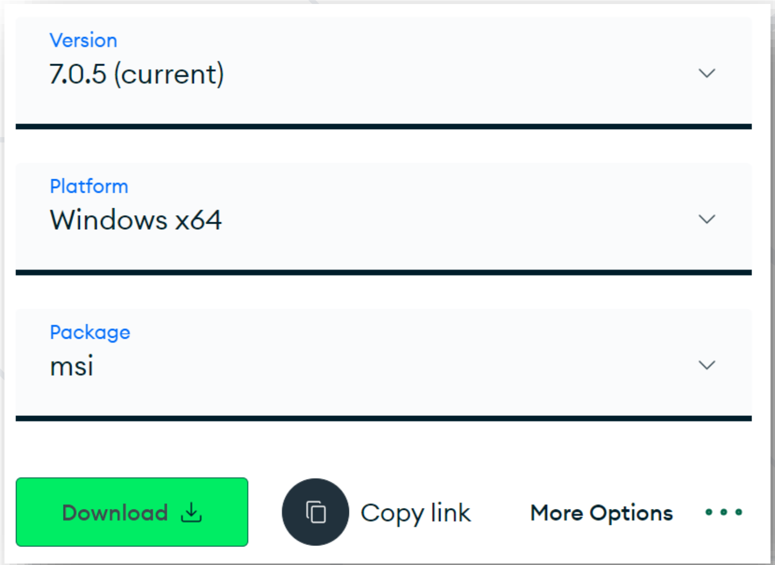
```
{
  _id: "5cf0029caff5056591b0ce7d",
  firstname: 'Jane',
  lastname: 'Wu',
  address: {
    street: '1 Circle Rd',
    city: 'Los Angeles',
    state: 'CA',
    zip: '90404'
  }
}
```

Install MongoDB

- Download from: mongodb.com/try/download/community

MONGODB COMMUNITY SERVER

MongoDB Community Server Download



Version: 7.0.5 (current)

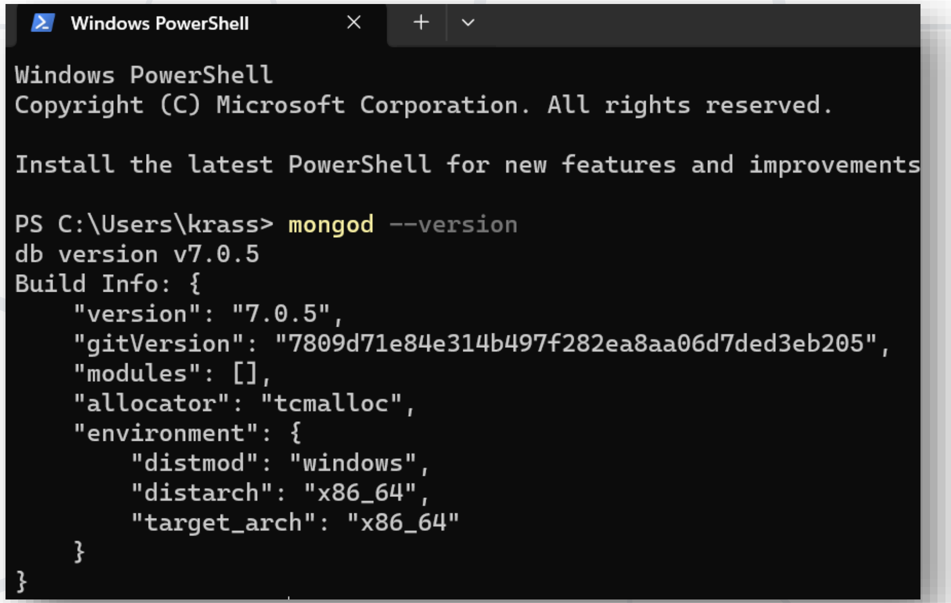
Platform: Windows x64

Package: msi

- The package includes **MongoDB Compass**

Running the MongoDB Server

- Ensure **MongoDB is installed** on your system by running `'mongod --version'` in your command line or terminal
  - If you're encountering the error "**mongod is not recognized as an internal or external command**" it typically indicates that MongoDB's executable **files are not in system's PATH environment variable**
  - The path to the MongoDB `'bin'` directory should be added to **system's PATH environment variable**



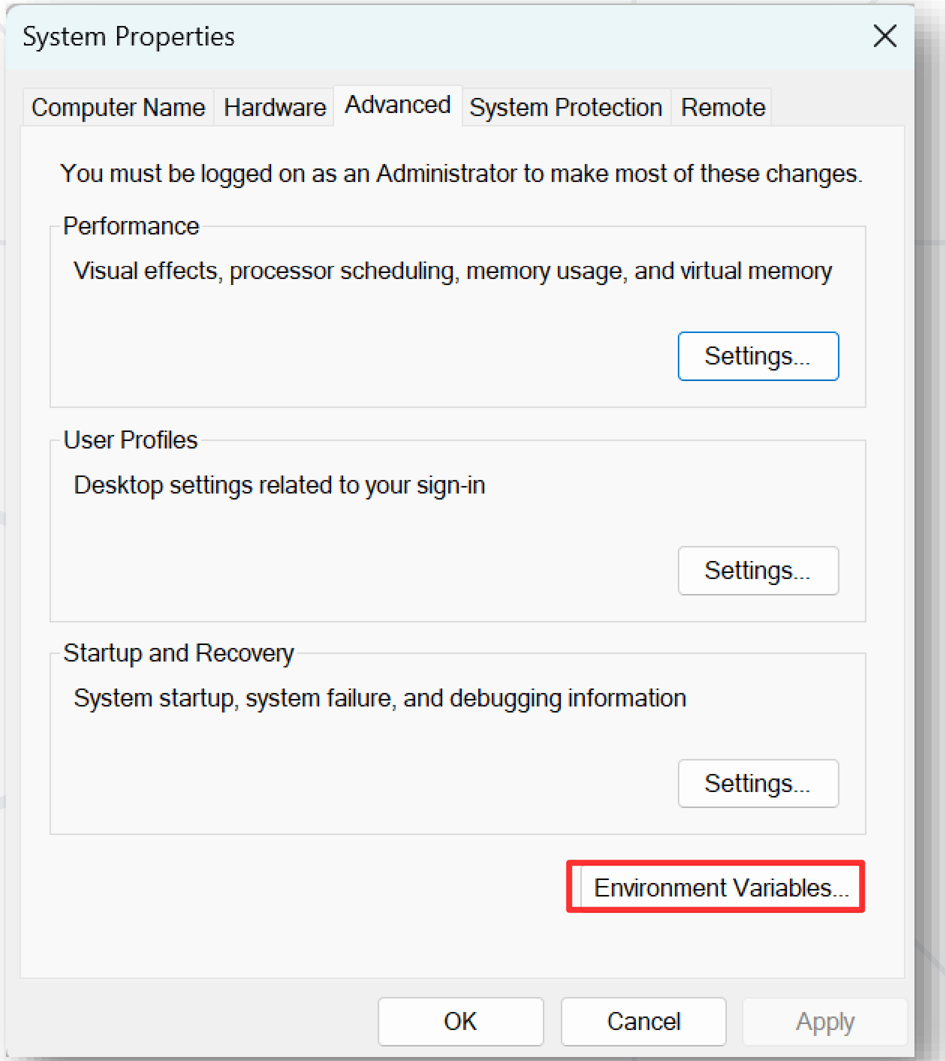
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements

PS C:\Users\krass> mongod --version
db version v7.0.5
Build Info: {
    "version": "7.0.5",
    "gitVersion": "7809d71e84e314b497f282ea8aa06d7ded3eb205",
    "modules": [],
    "allocator": "tcmalloc",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
```

Add MongoDB to the PATH EV

- Go to Control Panel
- **System** > **Advanced System Settings**
- **Environment Variables**
- **Path** > **Edit** > **New**
- Add the path to the MongoDB 'bin' directory (e.g.'`C:\Program Files\MongoDB\Server\{version}\bin`') at the end of the list



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

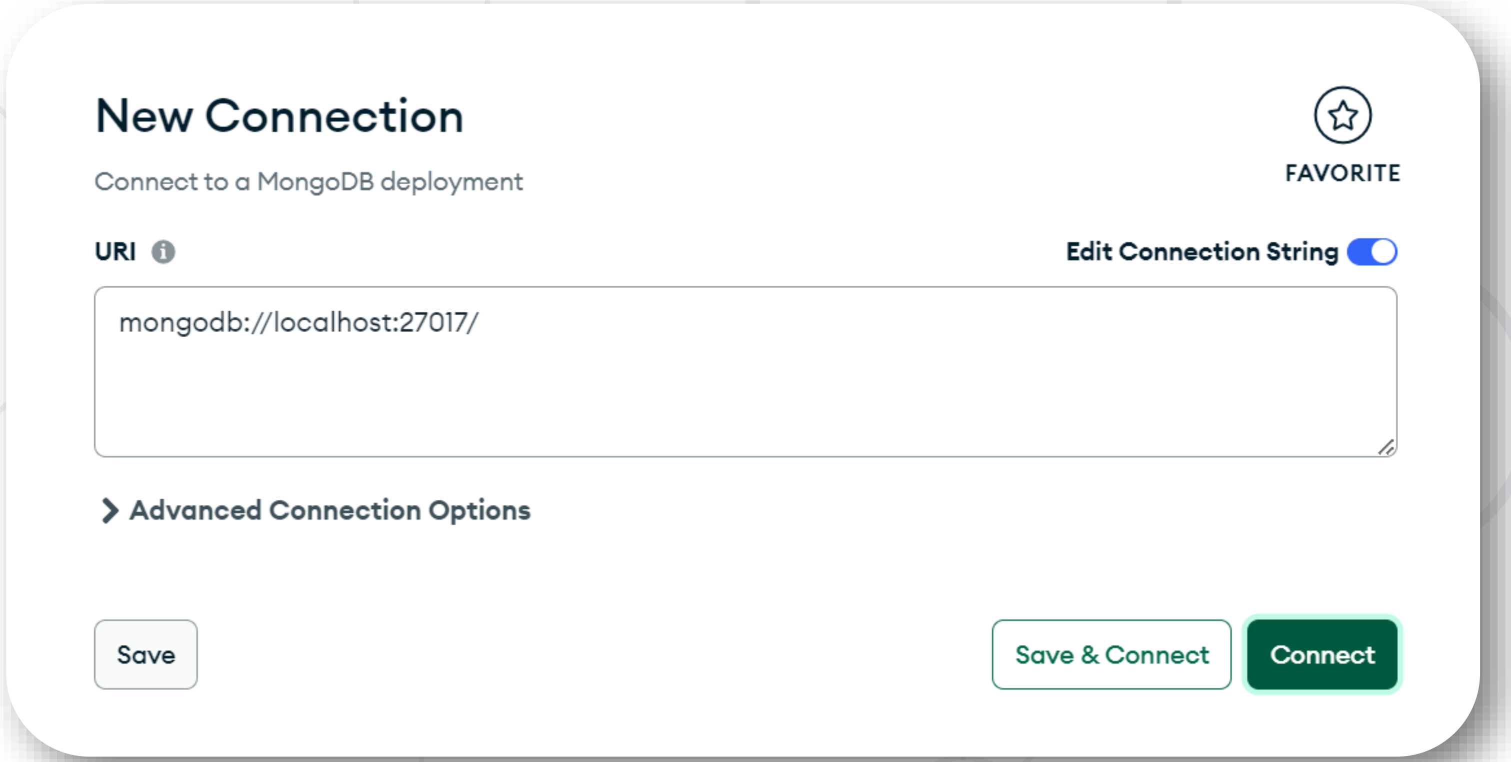
Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

Creating a Database in MongoDB Compass

- Click the "**Connect**" button to establish a connection with your **MongoDB server**



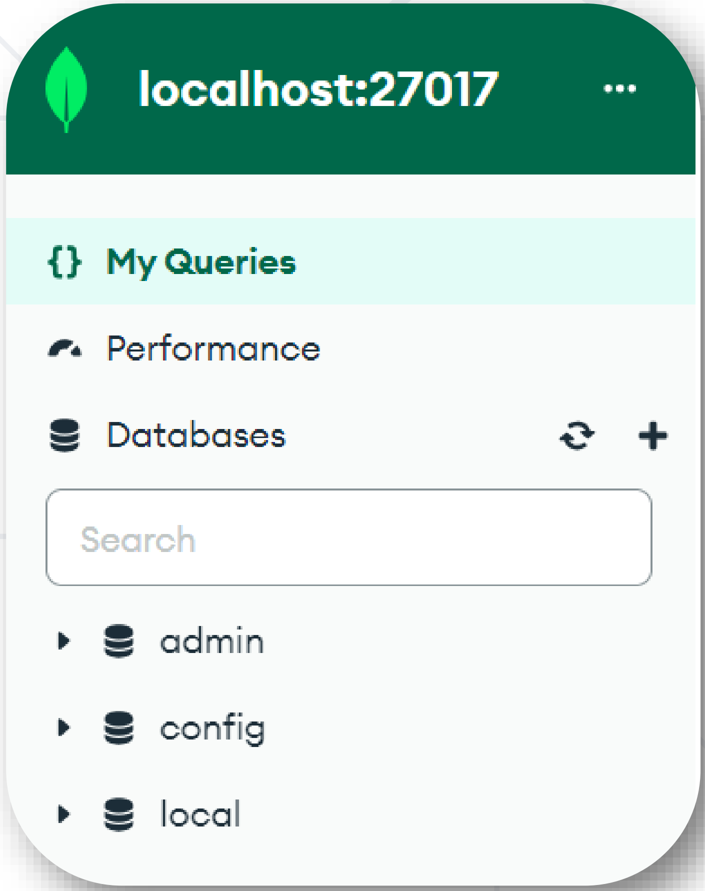
New Connection

Connect to a MongoDB deployment

URI

Edit Connection String

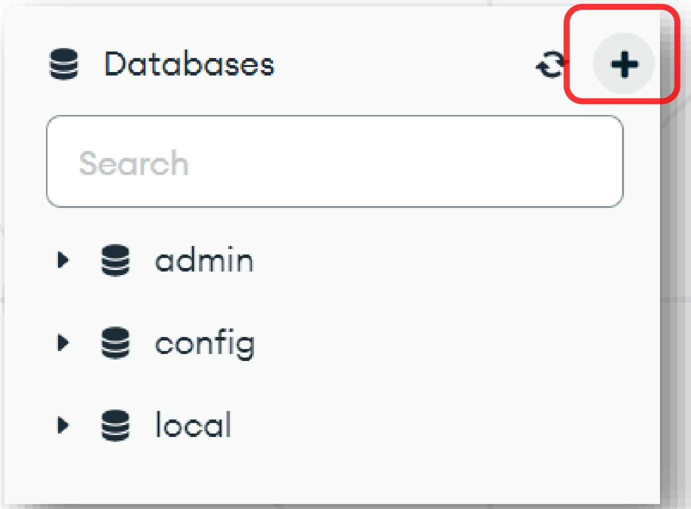
```
mongodb://localhost:27017/
```

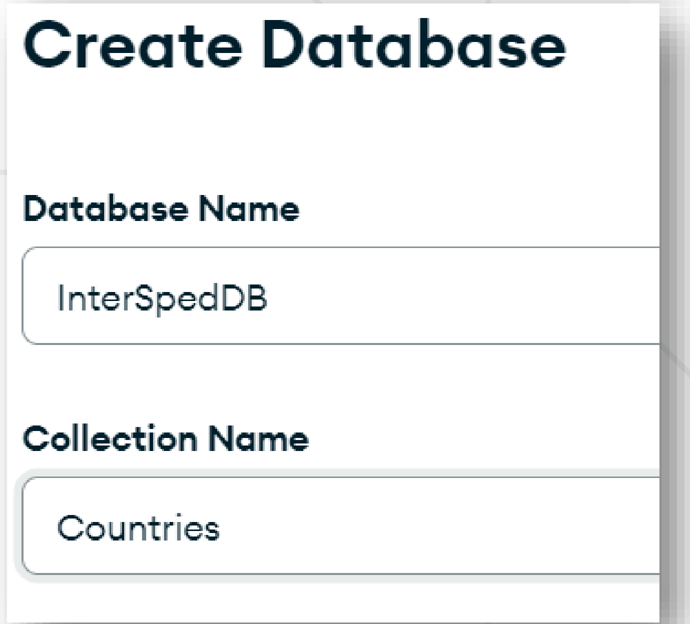


localhost:27017

Creating a Database in MongoDB Compass

- Once connected, the **MongoDB Compass** dashboard with a list of existing databases, will be shown
- "**CREATE DATABASE**" button usually found at the top right or bottom of the database list
- Entering the desired Database Name
- A **collection is similar to a table** in relational databases



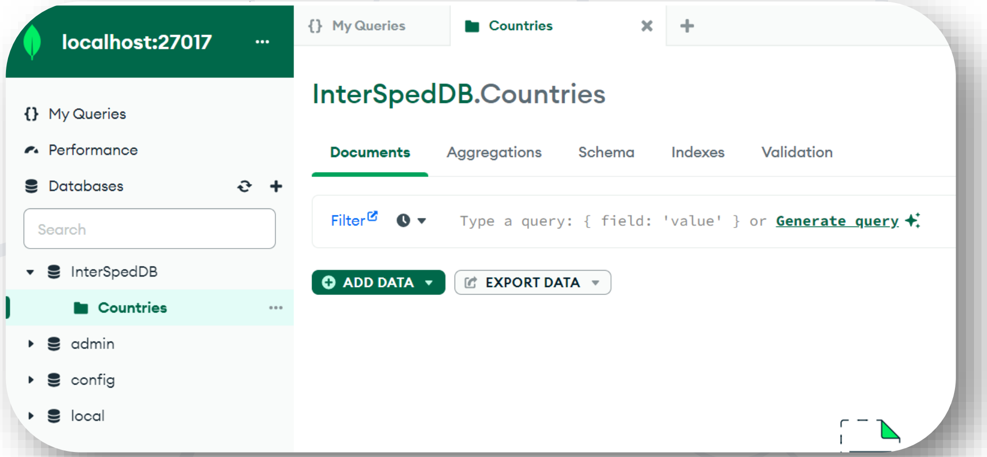

Create Database

Database Name: InterSpedDB

Collection Name: Countries

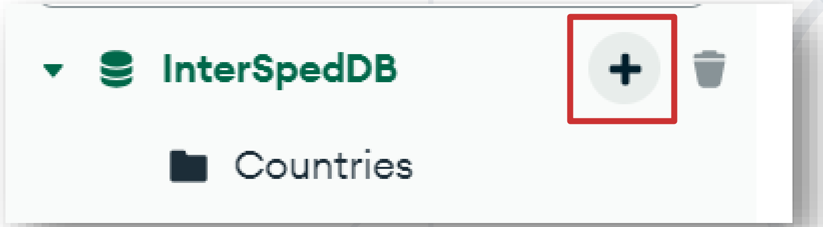
First Database Created

- After clicking "**Create Database**", Compass will create the new database and **the first collection** within it
- Databases and collections are **not physically created** until you insert data into them. The new database and collection will only be permanent after adding **at least one document** to the collection

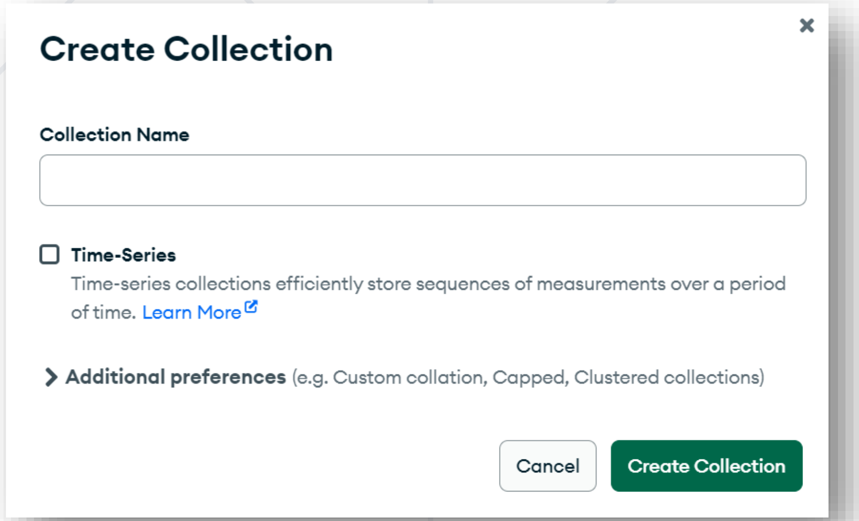


Creating a Collection in MongoDB Compass

- In the chosen database, click "**Create Collection**" button:



- A dialog box will appear, asking for details for the new collection:



Create Collection

Collection Name

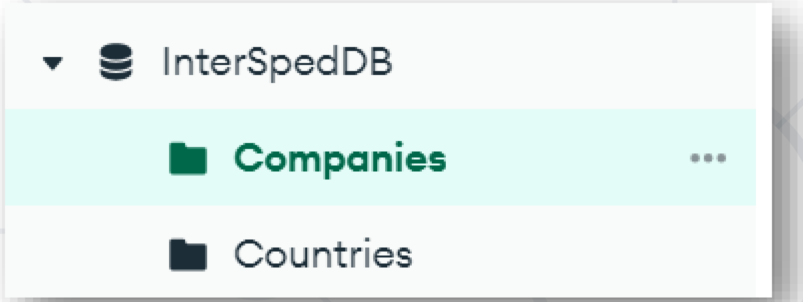
Time-Series

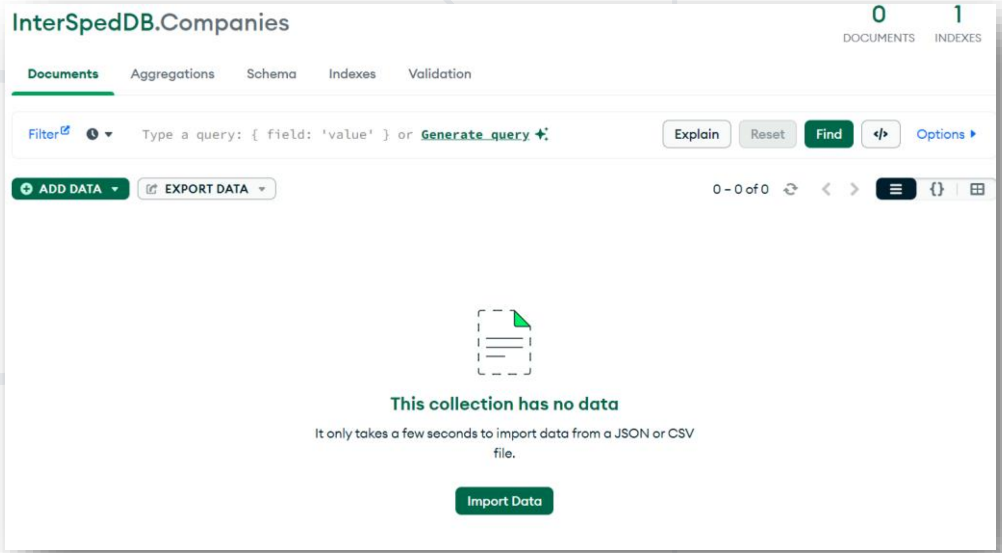
Time-series collections efficiently store sequences of measurements over a period of time. Learn More

Additional preferences (e.g. Custom collation, Capped, Clustered collections)

Using the Collection

- After creating the collection, it should appear in the **list of collections**
- With the collection created, **MongoDB Compass** can be used to insert, view and manage documents within the collection
- Setup indexes, and perform other DB management tasks





Type a query: { field: 'value' } or Generate query

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Inserting Collections in MongoDB

JSON and BSON

BSON – Binary JavaScript Object Notation

- **BSON**, or **Binary JSON**
  - The **data format** that MongoDB uses to organize and store data
- **BSON is a binary encoded JavaScript Object Notation**
  - A textual object notation widely used to transmit and store data across web based applications
- JSON is easier to understand as it is human-readable, but compared to BSON, it supports fewer data types
- **Both JSON** and **BSON** have the flexibility for presenting **complex and nested data structures**

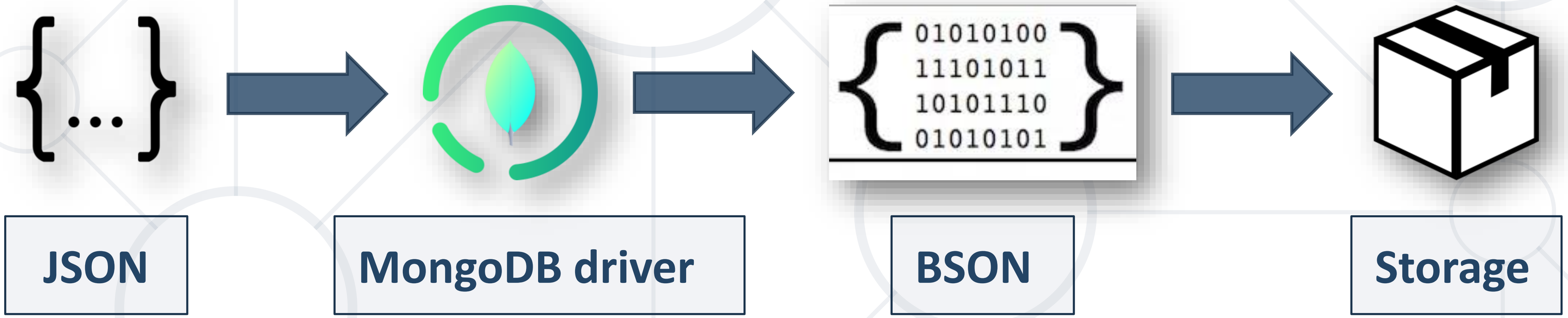
How is BSON different from JSON

- Type – BSON files are written in **binary**.
- Speed – BSON is **slow to read**, but **faster to build and scan**.
- Usage – Databases use BSON to **store data**.

- Type – JSON files are written in **text format**.
- Speed – JSON is **fast to read**, but **slower to build**.
- Usage – JSON is used to **send data** through the network (mostly through APIs).

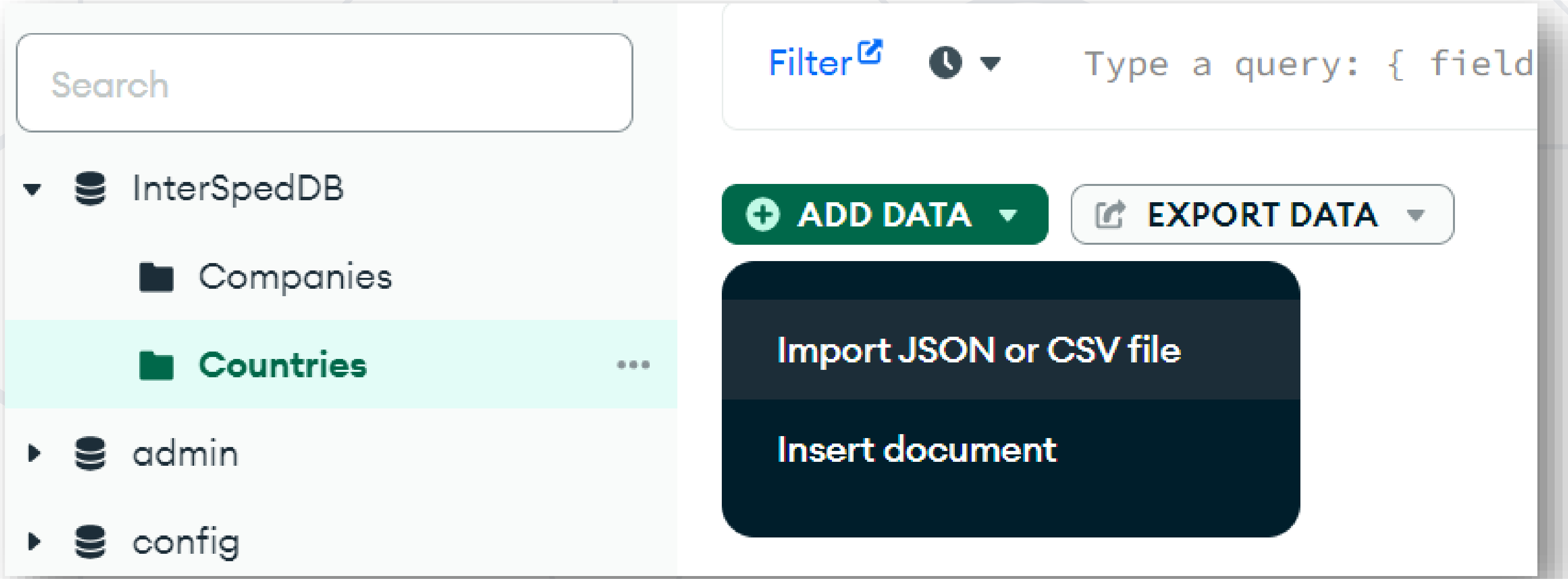
Preparing JSON Data

- **MongoDB** stores data in **BSON format**
  - That extends the JSON's capabilities with additional data types like: ObjectId, Date, and Binary.



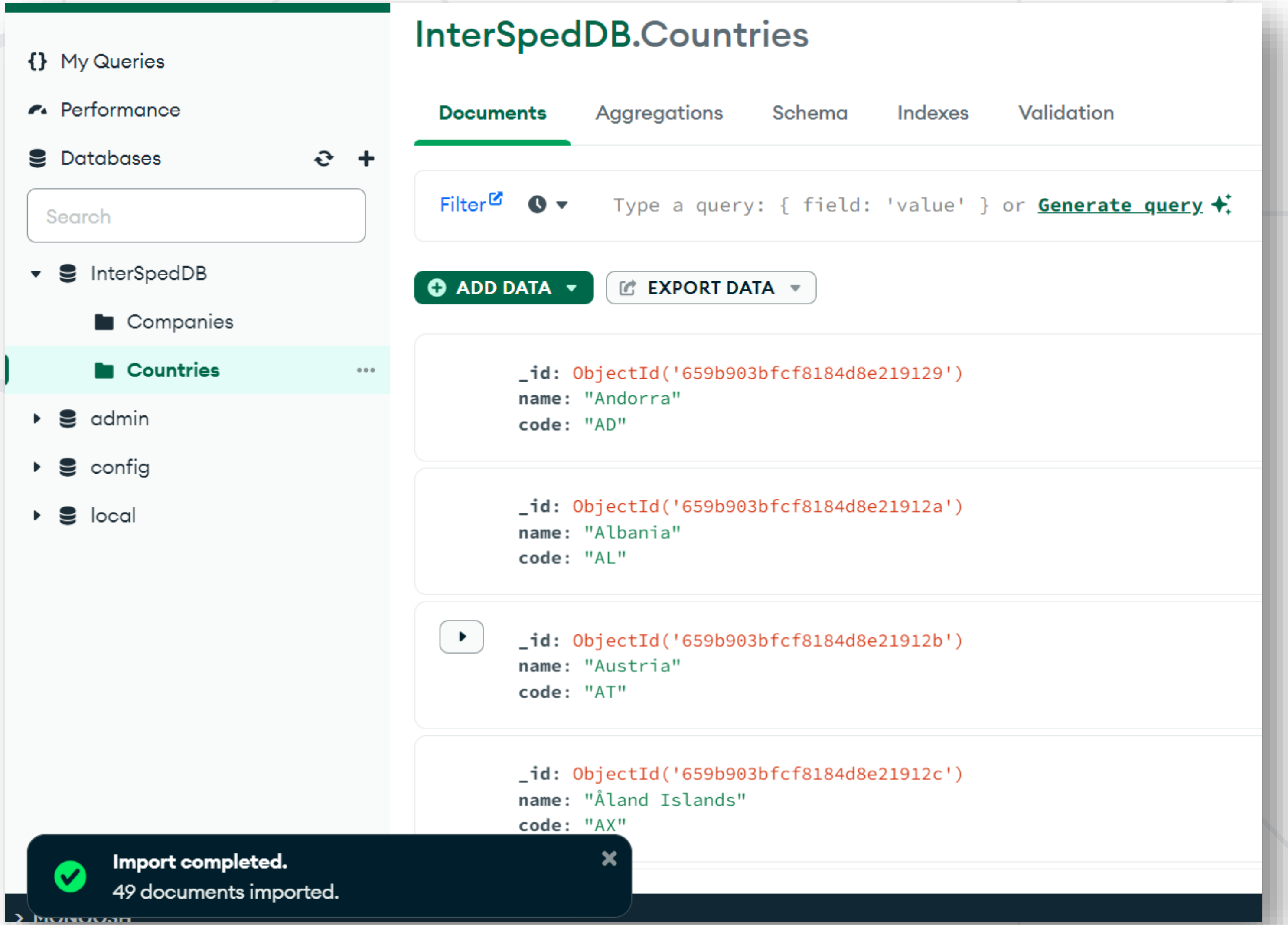
Import JSON Collection

- Inside the "**InterSpedDB**" database, locate the list of collections.
- With the "**Countries**" collection opened, you'll see any existing documents and various options.
- Click on the "**Import JSON or CSV file**":



Verify the Insertion

- Once complete, the new document should be visible in the "**Countries**" collection:



```
_id: ObjectId('659b903bfcf8184d8e219129')
name: "Andorra"
code: "AD"

_id: ObjectId('659b903bfcf8184d8e21912a')
name: "Albania"
code: "AL"

_id: ObjectId('659b903bfcf8184d8e21912b')
name: "Austria"
code: "AT"

_id: ObjectId('659b903bfcf8184d8e21912c')
name: "Åland Islands"
code: "AX"
```

Import completed.
49 documents imported.

MongoDB

Exploring Data in MongoDB

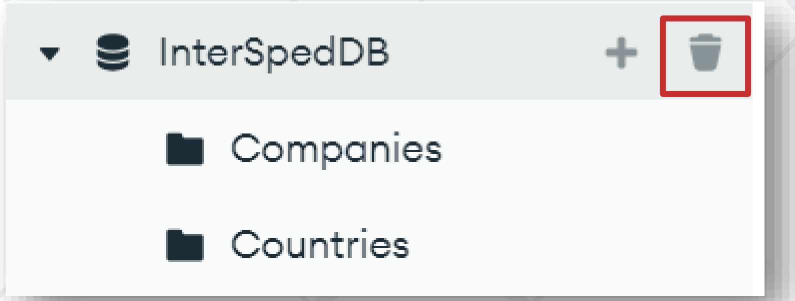
Browse / Query Collections

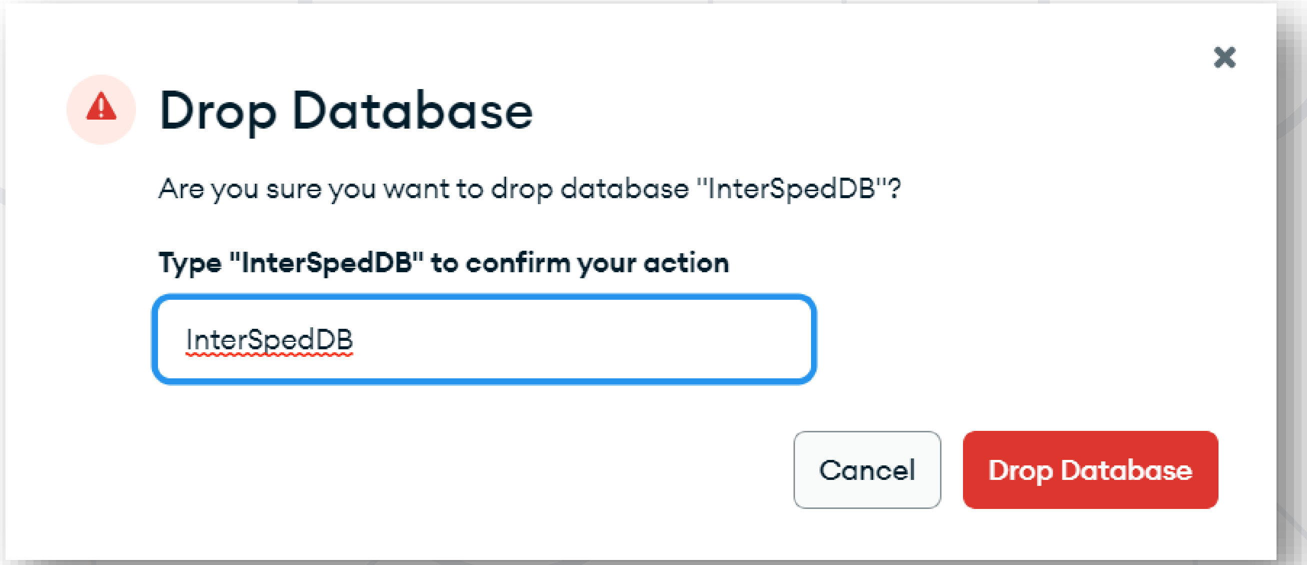
MongoDB Collections

- Collection in MongoDB is a group of documents.
- MongoDB is **schema-less**:
  - Documents in the same collection can have **different structures**.
  - In NoSQL, every **item** in the database **stands on its own**.
  - This simple modifications means that they are essentially **key-value stores**.

Delete Database

- Let's delete our "**InterSpedDB**", by clicking the delete icon:




Drop Database

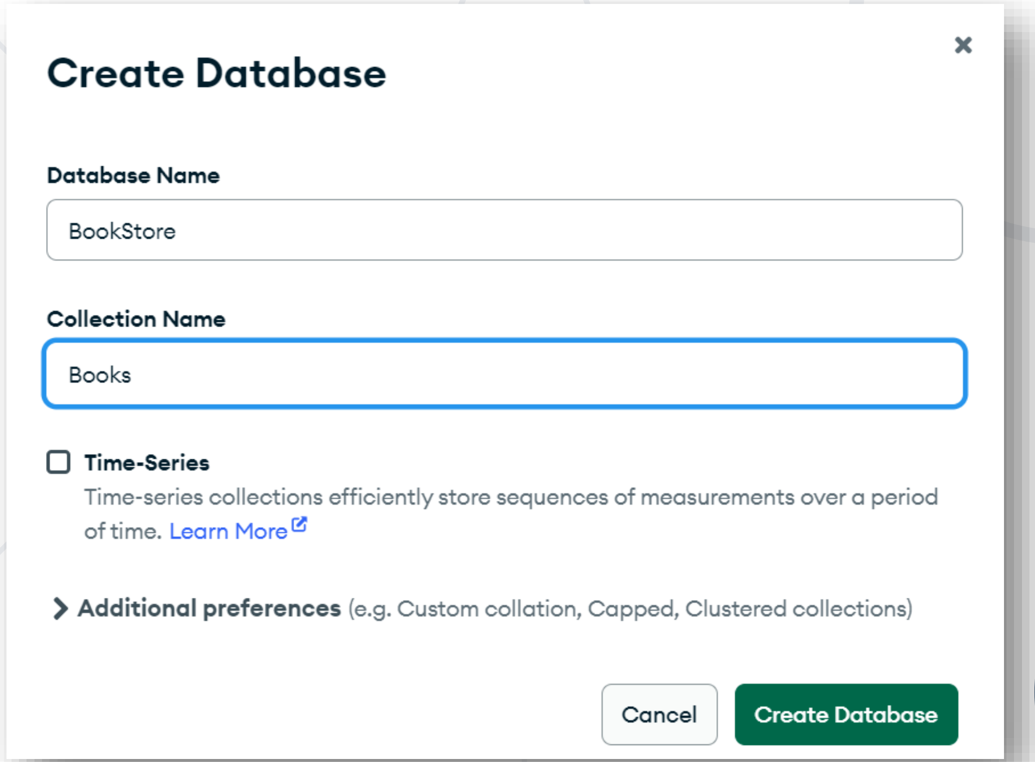
Are you sure you want to drop database "InterSpedDB"?

Type "InterSpedDB" to confirm your action

InterSpedDB

Bookstore Database

- **Create** a new database called **BookStore**



Create Database

Database Name

BookStore

Collection Name

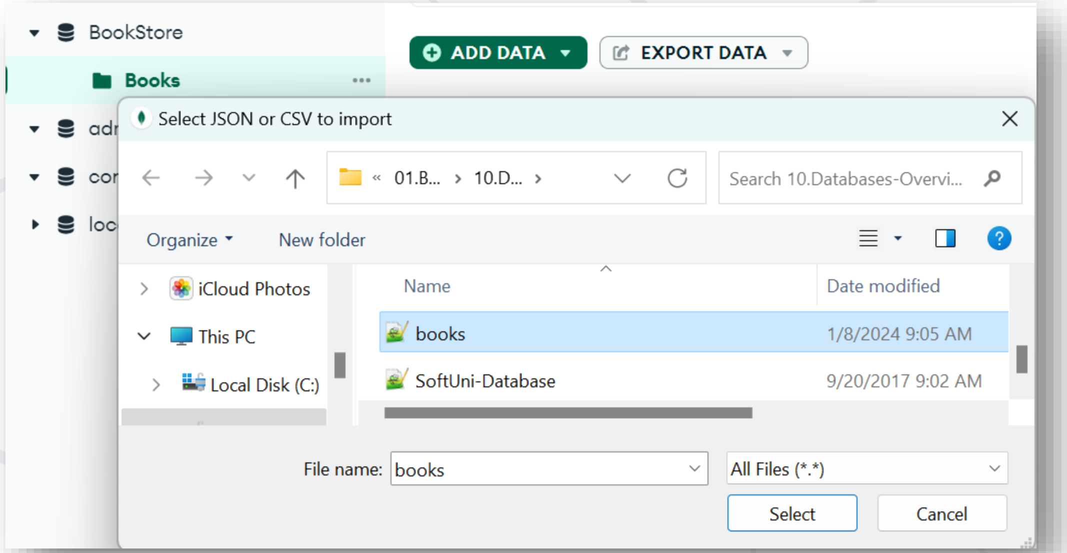
Books

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. Learn More

› Additional preferences (e.g. Custom collation, Capped, Clustered collections)

- **Insert** the provided document in the **Books collection**



Select JSON or CSV to import

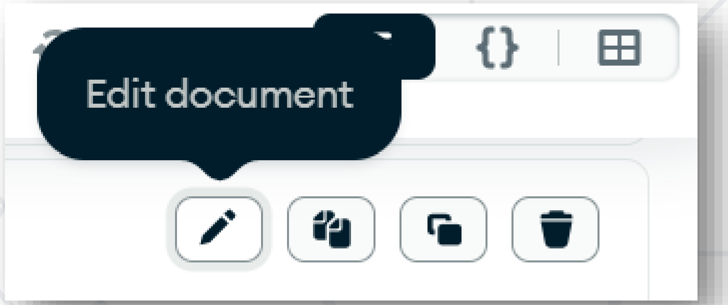
Name	Date modified
books	1/8/2024 9:05 AM
SoftUni-Database	9/20/2017 9:02 AM

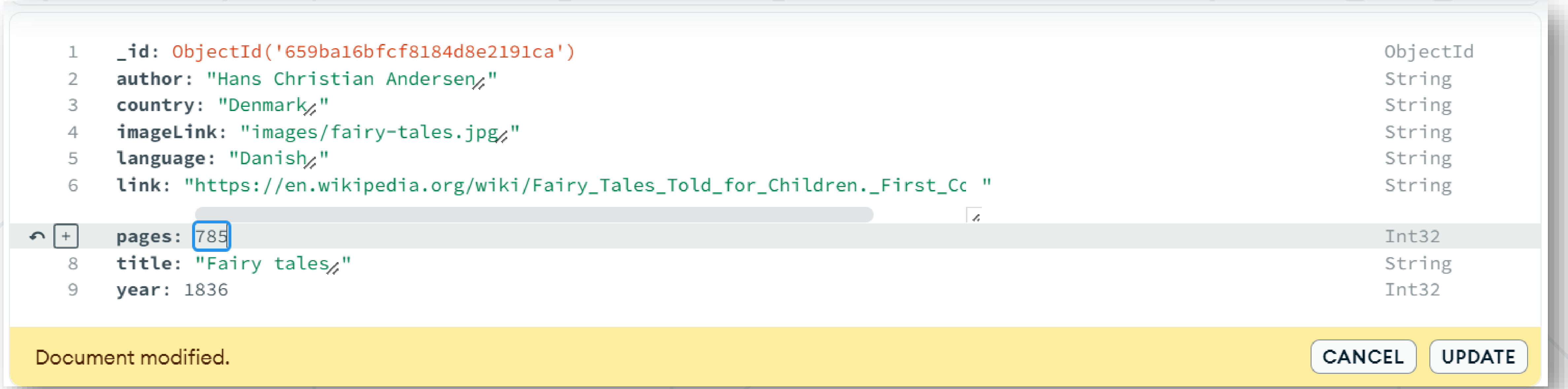
File name: books

All Files (*.*)

Update Value

- By clicking on the "**Edit**" button, a value can be **updated**



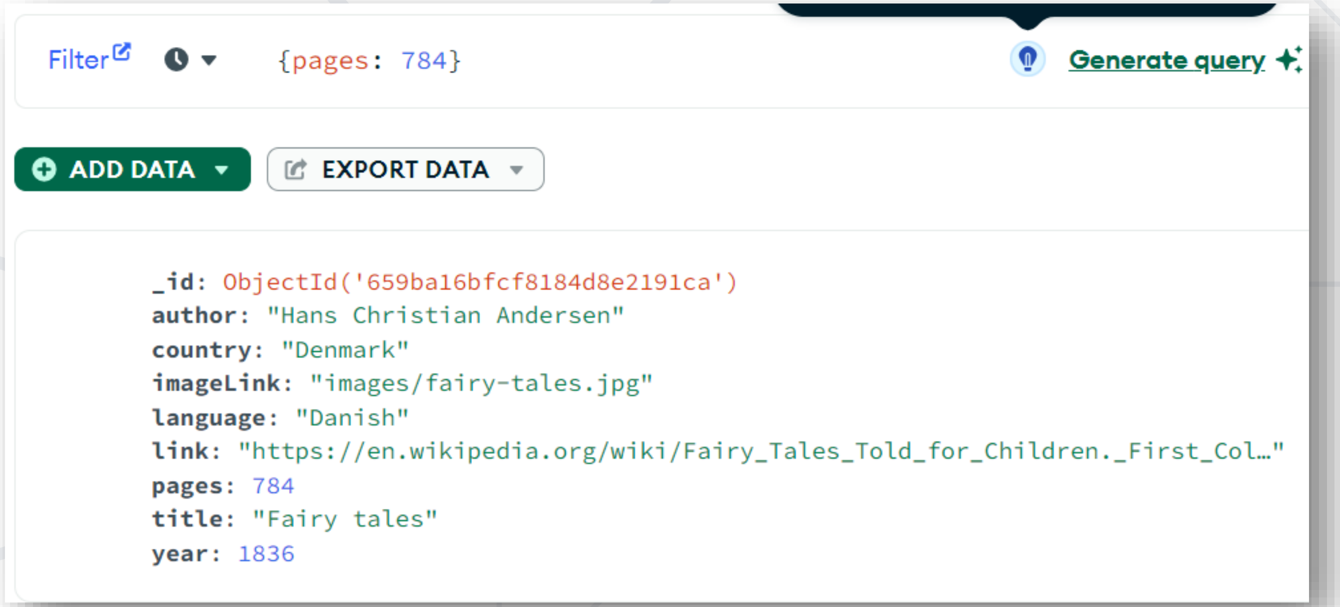


```
_id: ObjectId('659ba16bfcf8184d8e2191ca')                                    ObjectId
author: "Hans Christian Andersen"                                             String
country: "Denmark"                                                            String
imageLink: "images/fairy-tales.jpg"                                           String
language: "Danish"                                                            String
link: "https://en.wikipedia.org/wiki/Fairy_Tales_Told_for_Children._First_Cc "  String
pages: 785                                                                    Int32
title: "Fairy tales"                                                          String
year: 1836                                                                    Int32
```

Document modified.

MongoDB Query Language

- The basic structure of a MongoDB query is a **JSON-like syntax**
- **Find() operation** is used to **retrieve documents** from a collection

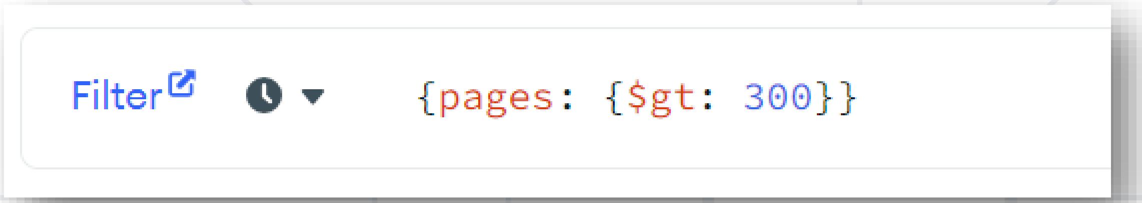


```
{pages: 784}
```

```
_id: ObjectId('659ba16bfcf8184d8e2191ca')
author: "Hans Christian Andersen"
country: "Denmark"
imageLink: "images/fairy-tales.jpg"
language: "Danish"
link: "https://en.wikipedia.org/wiki/Fairy_Tales_Told_for_Children._First_Col…"
pages: 784
title: "Fairy tales"
year: 1836
```

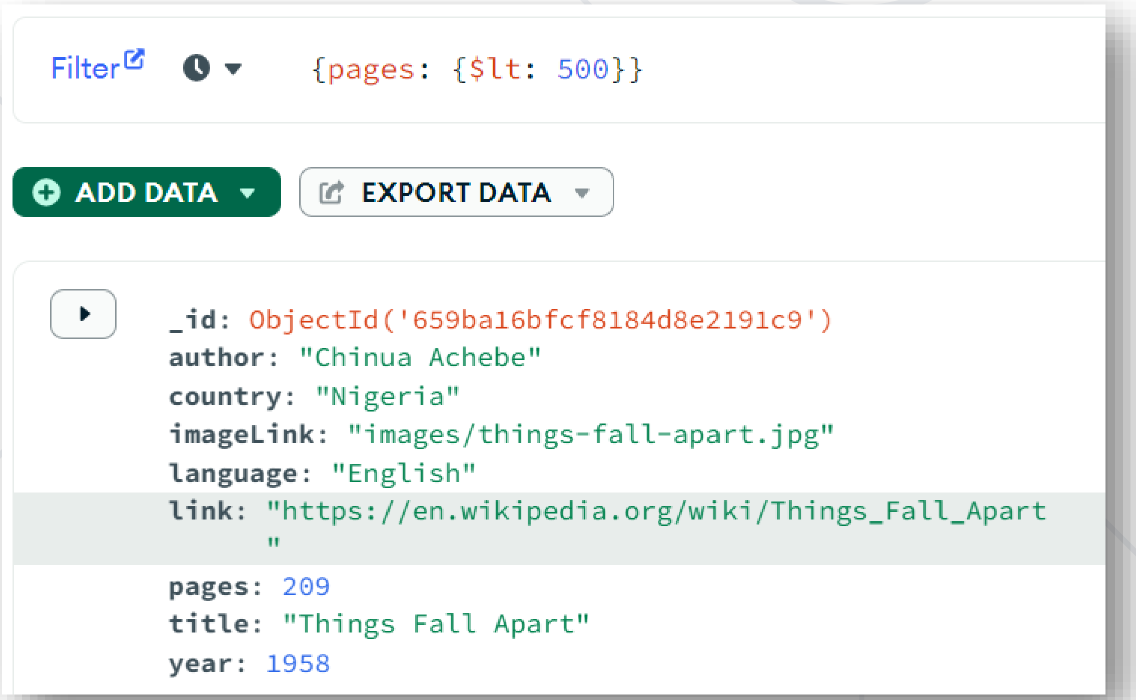
Greater Than / Lower Than

- We can filter the books where the pages field has a value **greater than** 300:



Filter `{pages: {$gt: 300}}`

- Or we can use **lower than**, to retrieve different results:

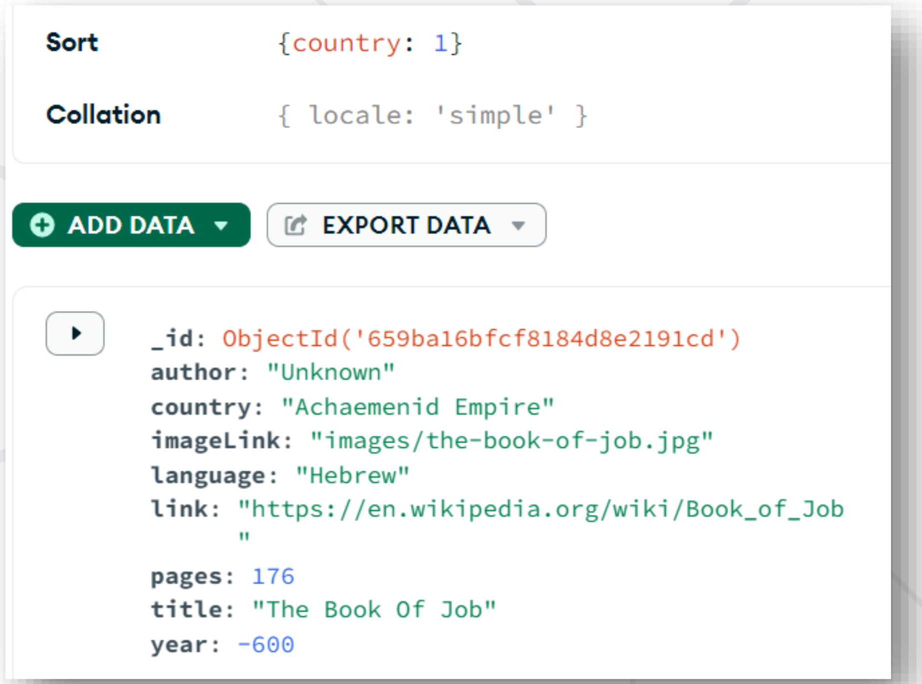


Filter `{pages: {$lt: 500}}`

```
_id: ObjectId('659ba16bfcf8184d8e2191c9')
author: "Chinua Achebe"
country: "Nigeria"
imageLink: "images/things-fall-apart.jpg"
language: "English"
link: "https://en.wikipedia.org/wiki/Things_Fall_Apart
      "
pages: 209
title: "Things Fall Apart"
year: 1958
```

Sort() Function

- The **sort()** function in MongoDB is used to **sort the result** of a query in either **ascending** or **descending** order based on one or more fields:
  - Field: The filed(s) you want to sort by
  - Order:
    - **1 for ascending order**
    - **-1 for descending order**

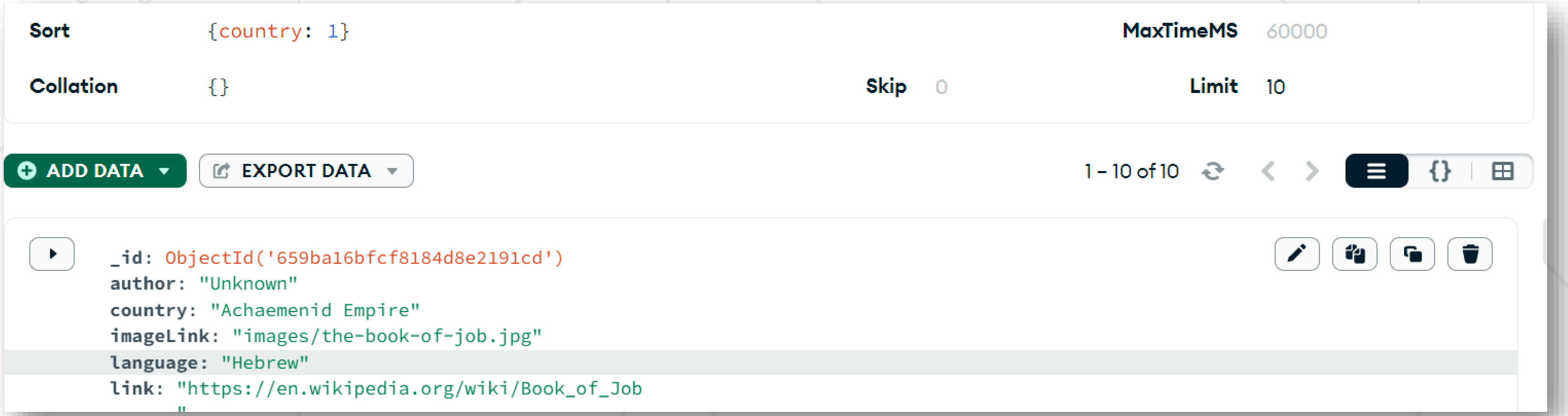


```
Sort         {country: 1}
Collation    { locale: 'simple' }

_id: ObjectId('659ba16bfcf8184d8e2191cd')
author: "Unknown"
country: "Achaemenid Empire"
imageLink: "images/the-book-of-job.jpg"
language: "Hebrew"
link: "https://en.wikipedia.org/wiki/Book_of_Job
      "
pages: 176
title: "The Book Of Job"
year: -600
```

Limit() Function

- The **limit()** function in MongoDB is used to **limit the number of documents returned** by a query
- It is useful when dealing with large collections to **avoid excessive data transfer**



Sort {country: 1} MaxTimeMS 60000
Collation {} Skip 0 Limit 10

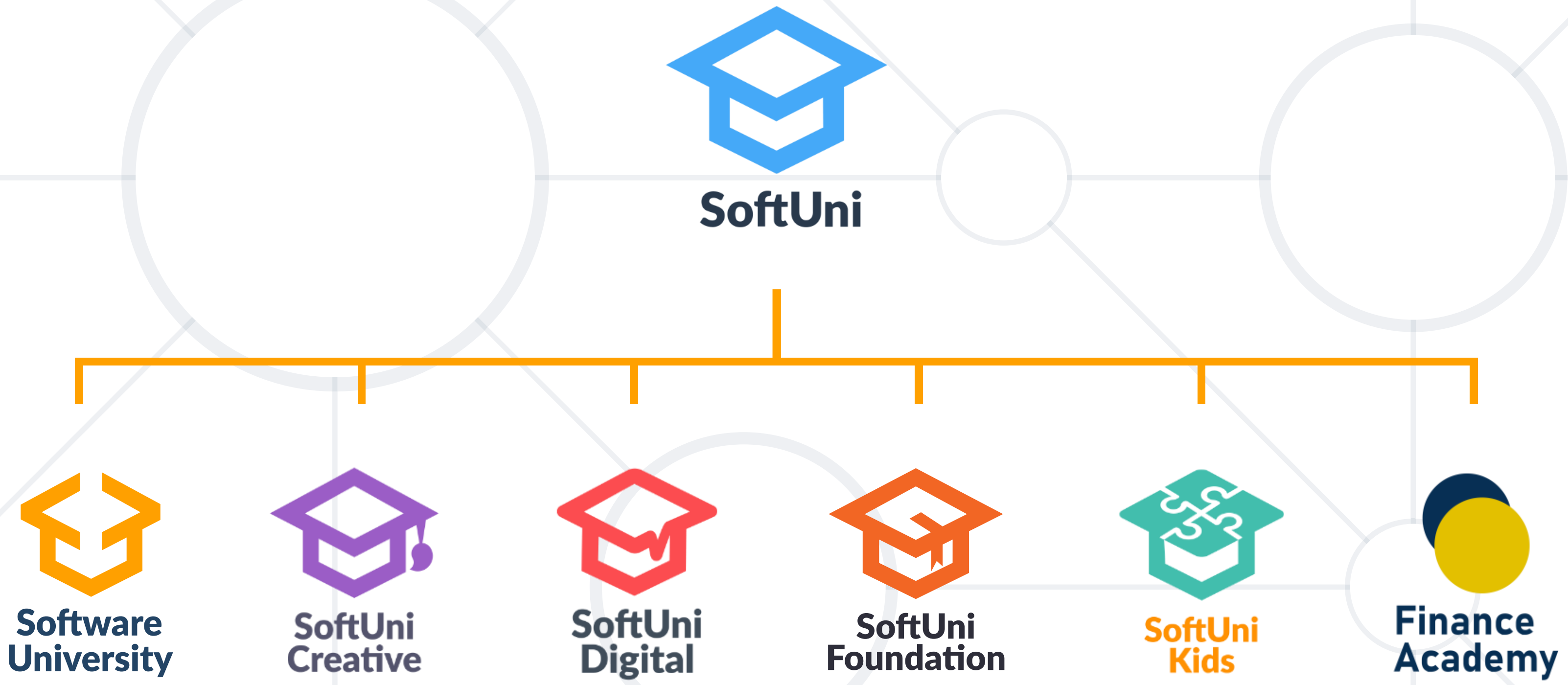
```
_id: ObjectId('659ba16bfcf8184d8e2191cd')
author: "Unknown"
country: "Achaemenid Empire"
imageLink: "images/the-book-of-job.jpg"
language: "Hebrew"
link: "https://en.wikipedia.org/wiki/Book_of_Job
```

Summary

- Databases Overview
- Roles of Databases in Back-End Testing
- Relational Databases – SQL Server Essentials
- NoSql Databases – MongoDB Essentials
- Performing Basic Queries in MongoDB

Questions?

Software University



SoftUni Diamond Partners

SmartIT

Coca-Cola HBC Bulgaria

SOFTWARE GROUP

DXC TECHNOLOGY

createX

Postbank

Решения за твоето утре

SUPER HOSTING .BG

DRAFT KINGS

VIVACOM

INDEAVR

Serving the high achievers

AMBITIONED

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License