

# Introduction to Databases

## How Do RDBMS Work? Managing DBs Using IDEs



**SoftUni Team**  
Technical Trainers



**SoftUni**



**Software University**

<https://softuni.bg>

# Table of Contents

1. Data Management
2. Database Engines
3. Data Types in SQL Server
4. Database Modeling
5. Basic SQL Queries



sli.do

**#csharp-db**



# **Data Management**

## When Do We Need a Database?

# Storage vs. Management (1)

- Conventional Data Storage
  - Notes
  - Receipts



# Storage vs. Management (2)

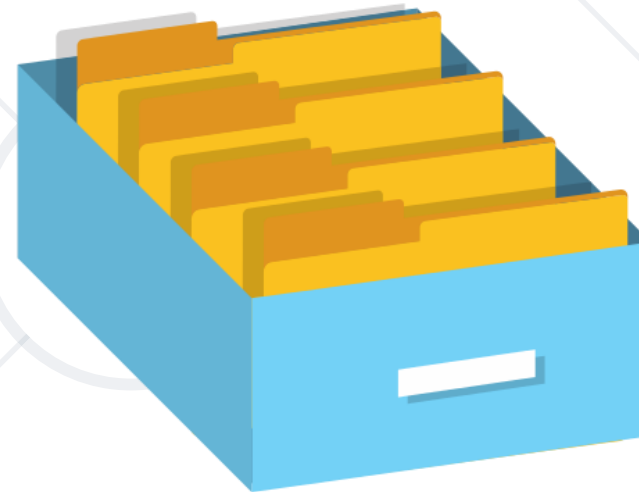
- We can group related pieces of data into separate columns



Order#	Date	Customer	Product	S/N	Unit Price	Qty	Total
315	07/16/2016	David Rivers	Oil Pump	OP147-0623	69.90	1	69.90

# Storage vs. Management (3)

- Storing data is **not** the primary reason to use a Database
- Flat storage **eventually** runs into **issues** with
  - Size
  - Ease of updating
  - Searching
  - Concurrency
  - Security
  - Consistency



# Databases and RDBMS

- A database is an **organized** collection of information
  - It imposes **rules** on the contained data
  - Relational storage first proposed by **Edgar Codd** in 1970
- A **R**elational **D**ata **B**ase **M**anagement **S**ystem provides tools to **manage** the database
  - It **parses requests** from the user and takes the **appropriate** action
  - The user doesn't have direct access to the stored data



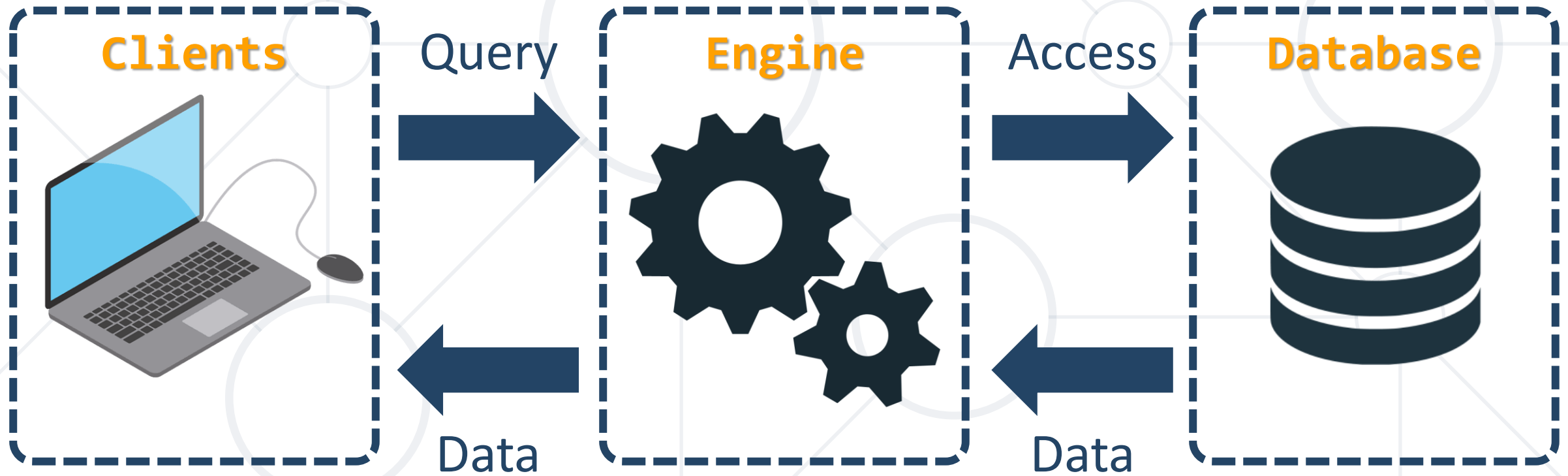




# Database Engines

# Database Engine Flow

- SQL Server uses the Client-Server Model



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

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

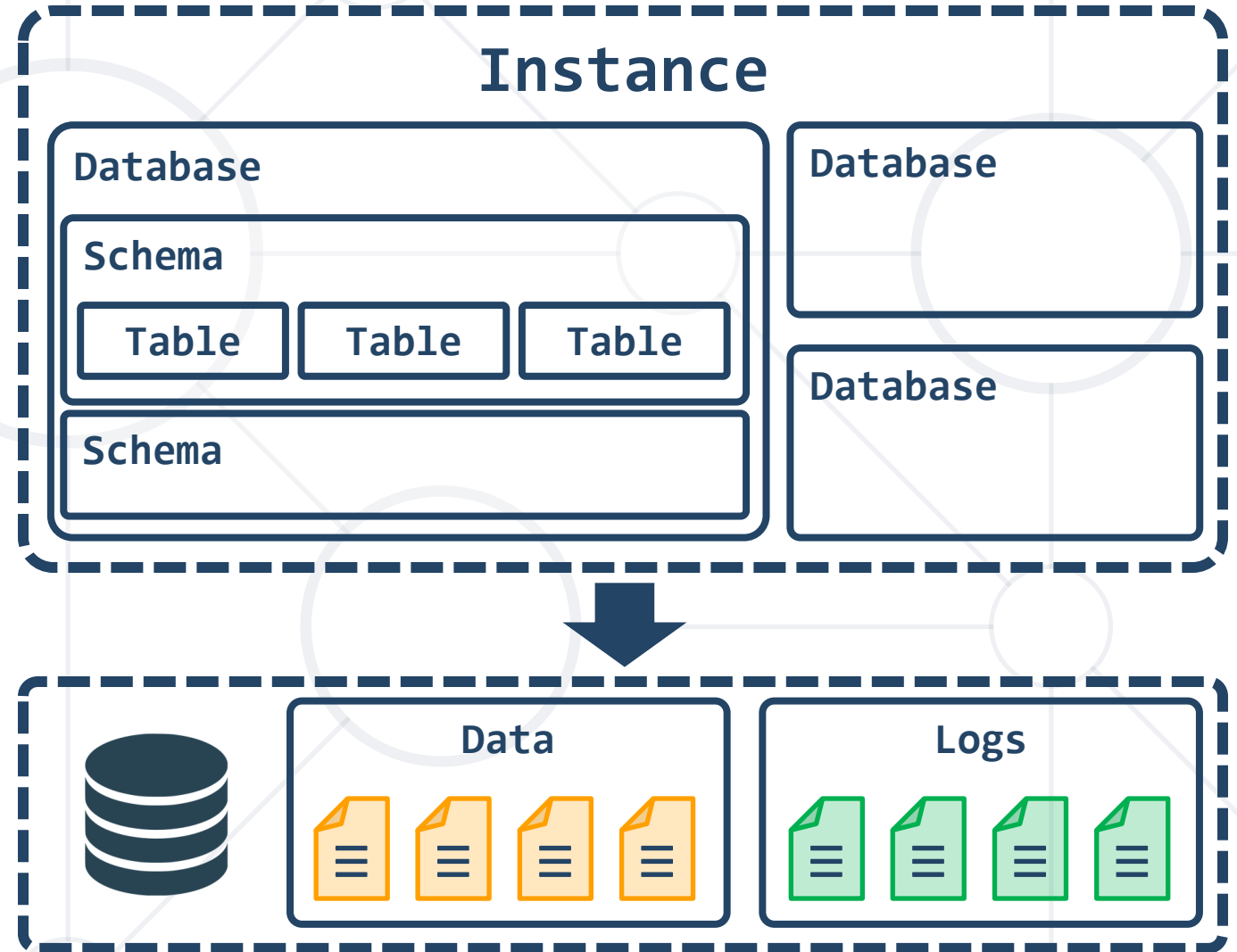
- Download SQL Server **Management Studio** separately

<https://aka.ms/ssmsfullsetup>



# SQL Server Architecture

- Logical Storage
  - Instance
    - Database
      - Schema
        - Table
        - Table
        - Table
      - Schema
  - Database
  - Database
- 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

- 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



# Data Types in SQL Server

- Numeric
  - **BIT** (1-bit), **TINYINT** (8-bit), **SMALLINT** (16-bit)
  - **INT** (32-bit), **BIGINT** (64-bit)
  - **FLOAT**, **REAL**, **DECIMAL**(**precision**, **scale**)
- Textual
  - **CHAR**(**size**) – fixed size string
  - **VARCHAR**(**size**) – variable size string
  - **NCHAR**(**size**) – Unicode fixed size string
  - **NVARCHAR**(**size**) – Unicode variable size string



# Size of Textual Characters

```
DECLARE @VarcharVar VARCHAR(5) = 'Test';  
DECLARE @NvarcharVar NVARCHAR(5) = 'Test';  
DECLARE @CharVar CHAR(5) = 'Test';  
DECLARE @NCharVar NCHAR(5) = 'Test';  
  
SELECT DATALENGTH(@VarcharVar),  
       DATALENGTH(@NvarcharVar),  
       DATALENGTH(@CharVar),  
       DATALENGTH(@NCharVar)
```

# Data Types in SQL Server (2)

- Binary data
  - **BINARY(size)** – fixed length sequence of bits
  - **VARBINARY(size)** – a sequence of bits, 1-8000 bytes or **MAX** (2GB)
- Date and time
  - **DATE** – date in range 0001-01-01 through 9999-12-31
  - **DATETIME** – date and time with precision of 1/300 sec
  - **DATETIME2** – type that has a larger date range
  - **SMALLDATETIME** – date and time (1 minute precision)
  - **TIME** – defines a time of a day (no time zone)
  - **DATETIMEOFFSET** – date and time that has time zone

# Date and Time in SQL Server

DATA TYPE	① RANGE OF VALUES	② ACCURACY	③ STORAGE SPACE
<i>SMALLDATETIME</i>	01/01/1900 to 06/06/2079	<u>1 minute</u>	4 bytes
<i>DATETIME</i>	01/01/ <u>1753</u> to 12/31/9999	0.00333 seconds	8 bytes
<i>DATETIME2</i>	01/01/ <u>0001</u> to 12/31/9999	100 <u>nanoseconds</u>	<u>6 to 8 bytes</u>
<i>DATETIMEOFFSET</i>	01/01/0001 to 12/31/9999	100 nanoseconds	8 to 10 bytes
<i>DATE</i>	01/01/0001 to 12/31/9999	1 day	<u>3 bytes</u>
<i>TIME</i>	00:00:00.00000000 to 23:59:59.99999999	100 nanoseconds	3 to 5 bytes

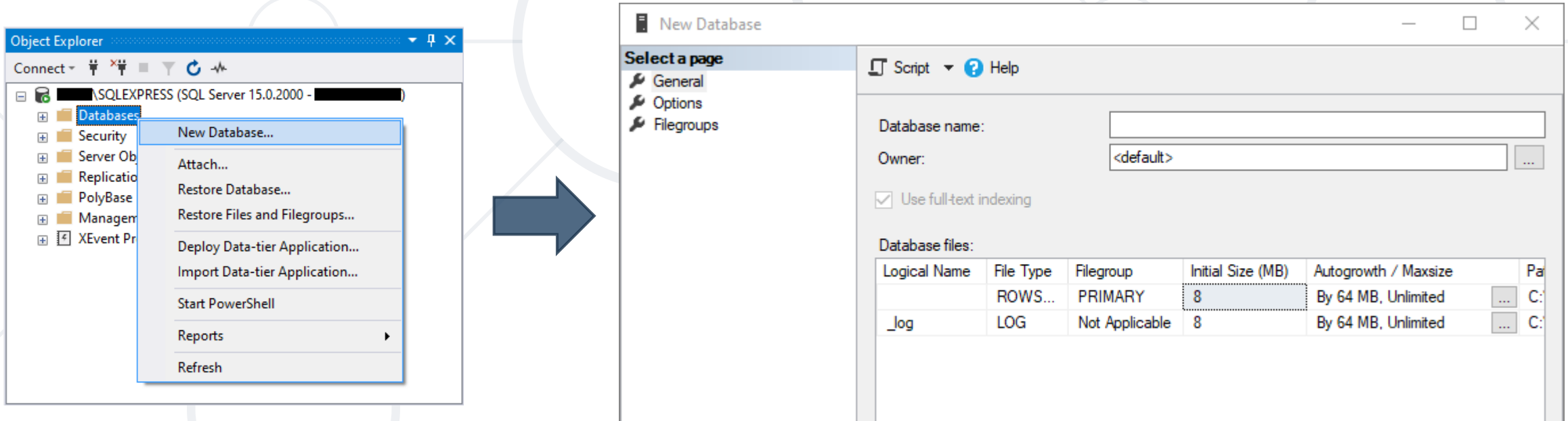


# **Database Modelling**

Data Definition Using SSMS

# Creating a New Database

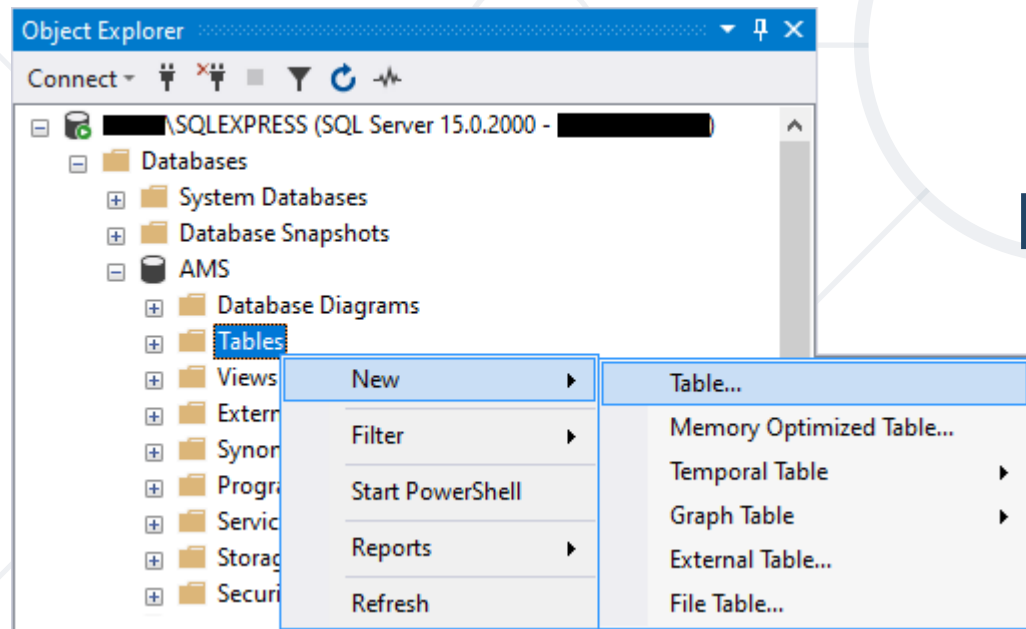
- Select **New Database** from the **context menu** under "Databases"



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

# Creating Tables (1)

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

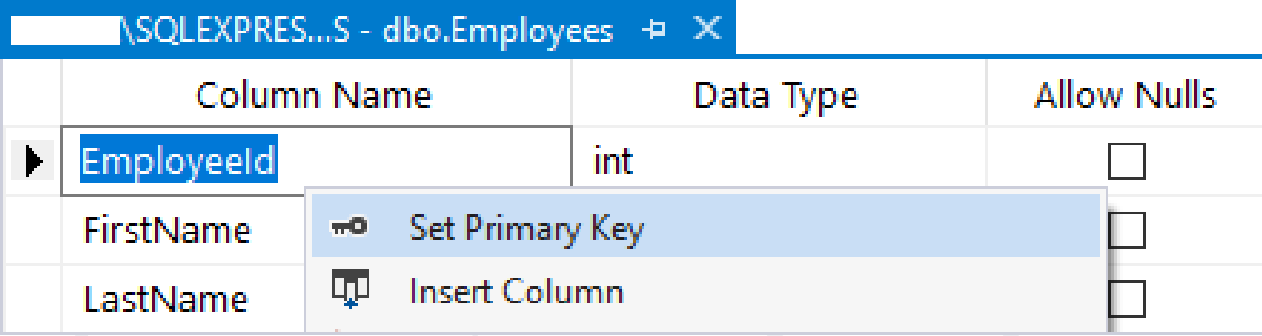


Column Name	Data Type	Allow Nulls
EmployeeId	int	<input type="checkbox"/>
FirstName	nvarchar(50)	<input type="checkbox"/>
LastName	nvarchar(50)	<input type="checkbox"/>

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

# Creating Tables (2)

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



The screenshot shows a SQL Server Enterprise Manager window titled "\SQLEXPRES...S - dbo.Employees". It displays a table with three columns: EmployeeId, FirstName, and LastName. The 'EmployeeId' column is selected, and a context menu is open over it, showing options like 'Set Primary Key' and 'Insert Column'.

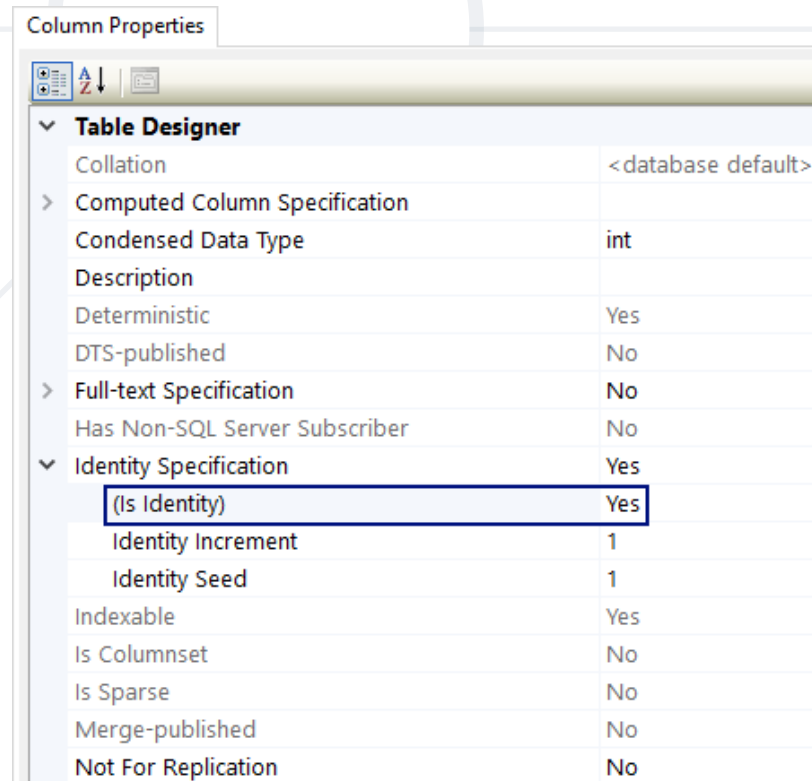
	Column Name	Data Type	Allow Nulls
▶	EmployeeId	int	<input type="checkbox"/>
	FirstName		<input type="checkbox"/>
	LastName		<input type="checkbox"/>

- **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



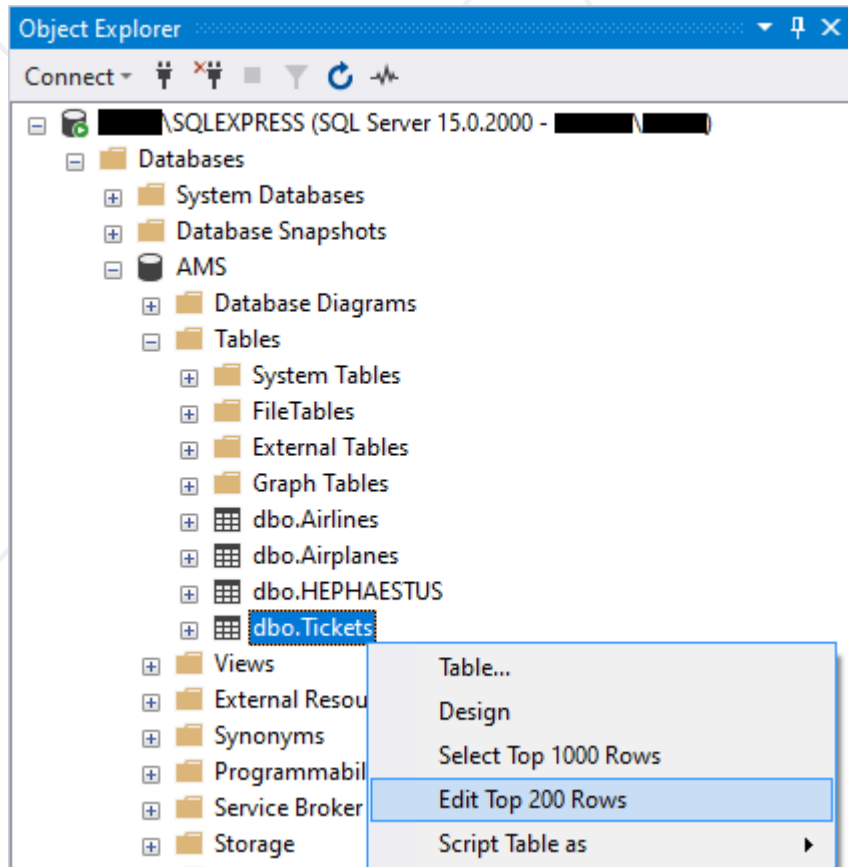
# Creating Tables (4)

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



# Storing and Retrieving Data (1)

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

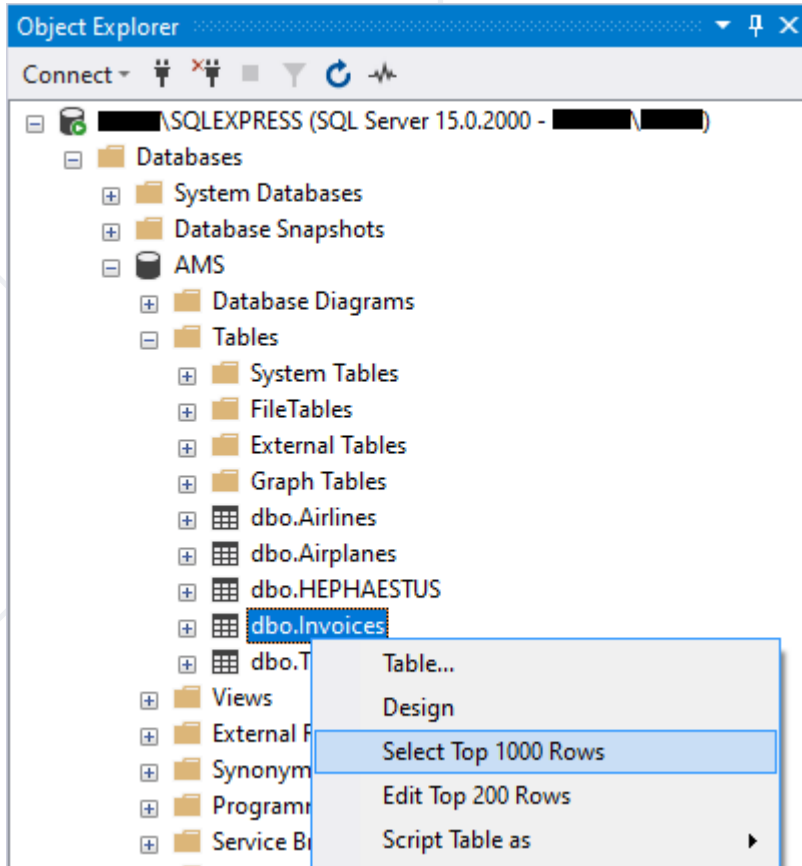


	TicketId	Price	Class	Seat	CustomerId
	1	4500.00	First	233-A	3
	2	2669.85	Second	123-D	1
	3	1800.75	Second	12-Z	2
	4	616.02	Third	45-Q	2
	5	840.00	Third	201-R	4
	6	3150	Second	13-T	1
	7	5500.00	First	98-O	2
	8	100.00	First	1	1
▶▶	NULL	NULL	NULL	NULL	NULL

Enter data at the end to add a new row

# Storing and Retrieving Data (2)

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

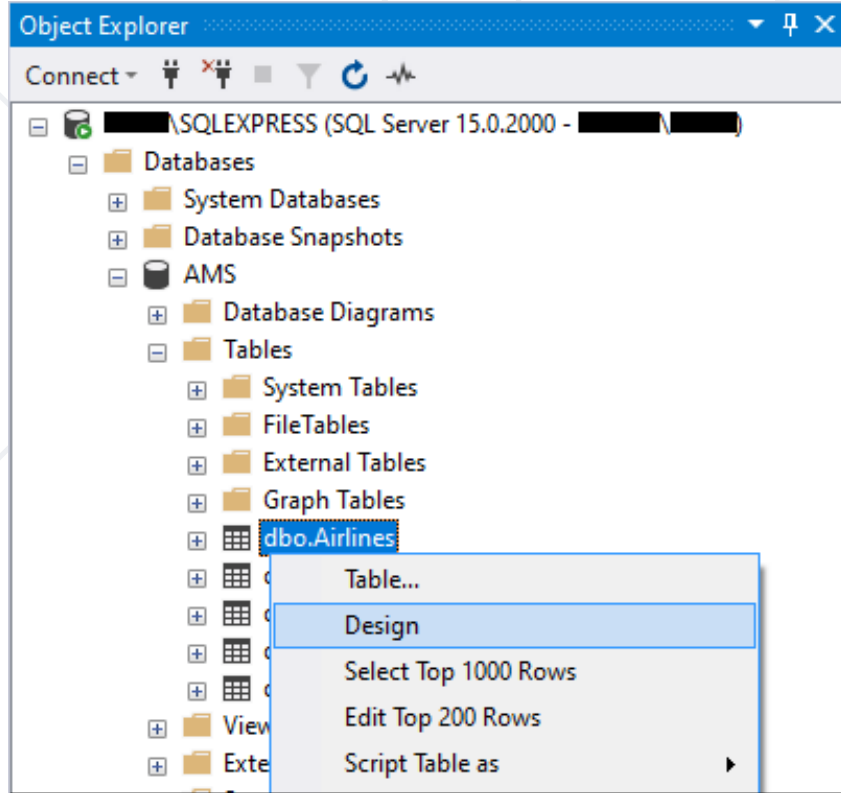


InvoiceId	CustomerId	InvoiceDate	BillingAddress	BillingCountry
1	1	2022-12-01	Theodor Heuss Strasse 34	Germany
2	4	2022-12-15	Ullevaalsveien 14	Norway
3	8	2023-01-03	Gretrystraat 63	Belgium
4	23	2023-01-11	69 Salem Stree	United States
5	14	2023-01-06	8210 111 ST NW	United States
6	37	2023-01-19	Berger 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



SQLSERVEREXPRESS...MS - dbo.Airlines			
	Column Name	Data Type	Allow Nulls
	AirlineId	int	<input type="checkbox"/>
	AirlineName	nvarchar(30)	<input type="checkbox"/>
	Nationality	nvarchar(3)	<input type="checkbox"/>
	Rating	int	<input checked="" type="checkbox"/>

Changes cannot conflict with existing rules!



# **Basic SQL Queries**

## Data Definition Using T-SQL

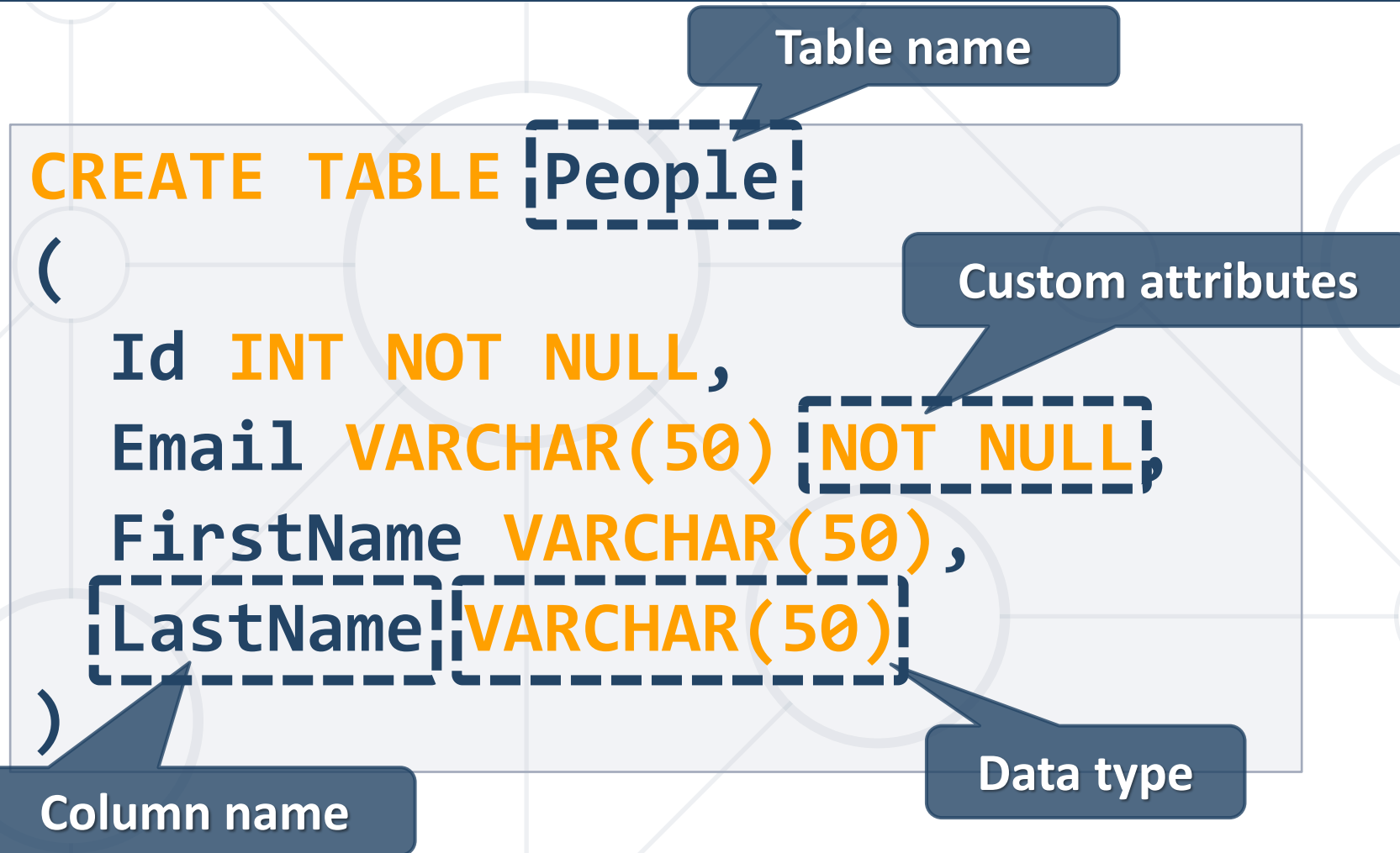
- We can communicate with the database engine using SQL
- Queries provide greater **control** and **flexibility**
- To create a database using SQL:

```
CREATE DATABASE Employees
```

Database name

- SQL keywords are traditionally **capitalized**

# Table Creation 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

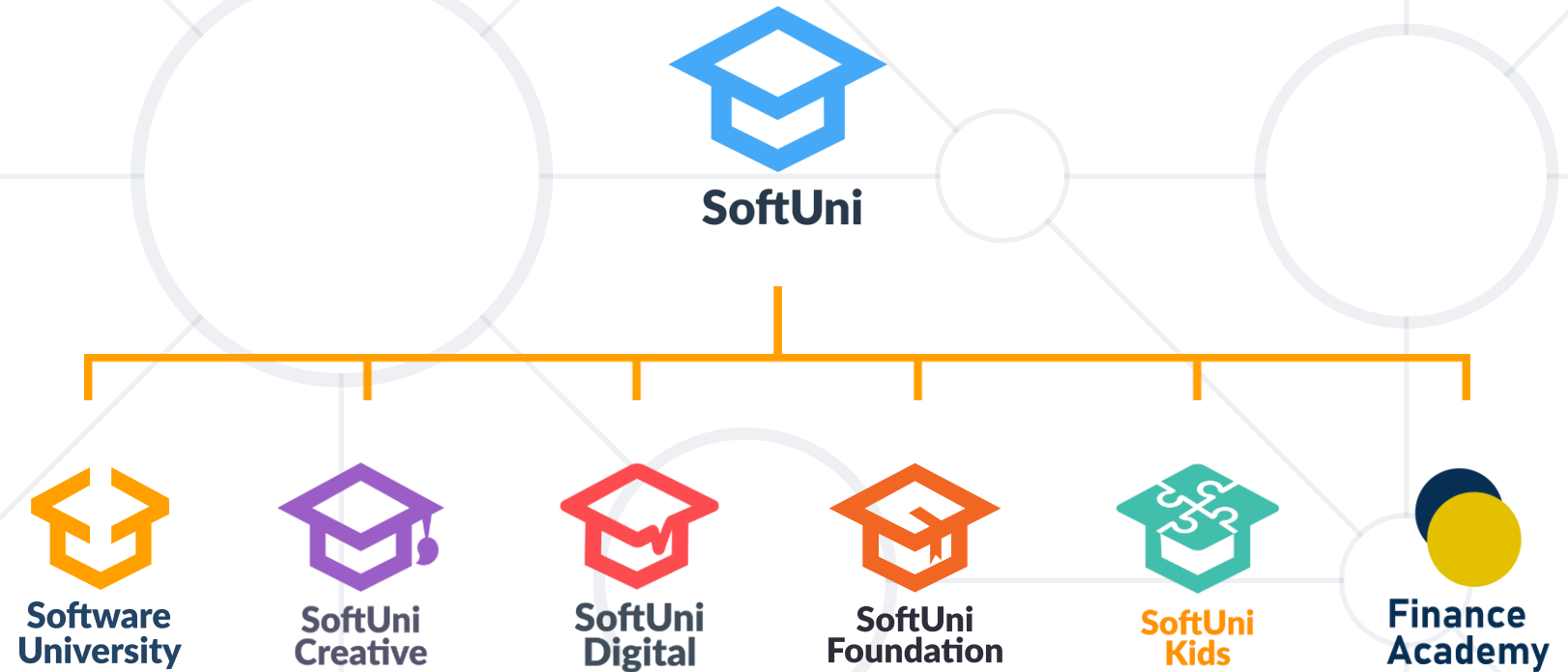
```
SELECT TOP (5) FirstName, LastName  
FROM Employees
```



- RDBMS stores and manages data
- Table relations reduce repetition and complexity
- Table columns have **fixed types**
- We can use Management Studio to **create** and **customize** tables
- SQL provides **greater control** over actions



# Questions?



# SoftUni Diamond Partners

**SUPER  
HOSTING  
.BG**



**Coca-Cola HBC  
Bulgaria**

 **Flutter**<sup>TM</sup>  
International

**INDEAVR**  
Serving the high achievers



**AMBITIONED**

 **DRAFT  
KINGS**



**BOSCH**

 **Postbank**  
*Решения за твоето утре*

 **PHAR  
VISION**



**SmartIT**

**DXC**  
TECHNOLOGY

createX  


- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg>
- © Software University – <https://softuni.bg>

