Databases Basics

How do RDBMS work?

SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Contents

1. Database Management
2. Database Engine
3. Structured Query Language
   - Creating Databases and Tables
   - Inserting Values
   - Queries
   - Deleting

Have a Question?

SoftUni Foundation

sli.do

#TECH-FUND

Database Management

When Do We Need a Database?

Storage vs. Management

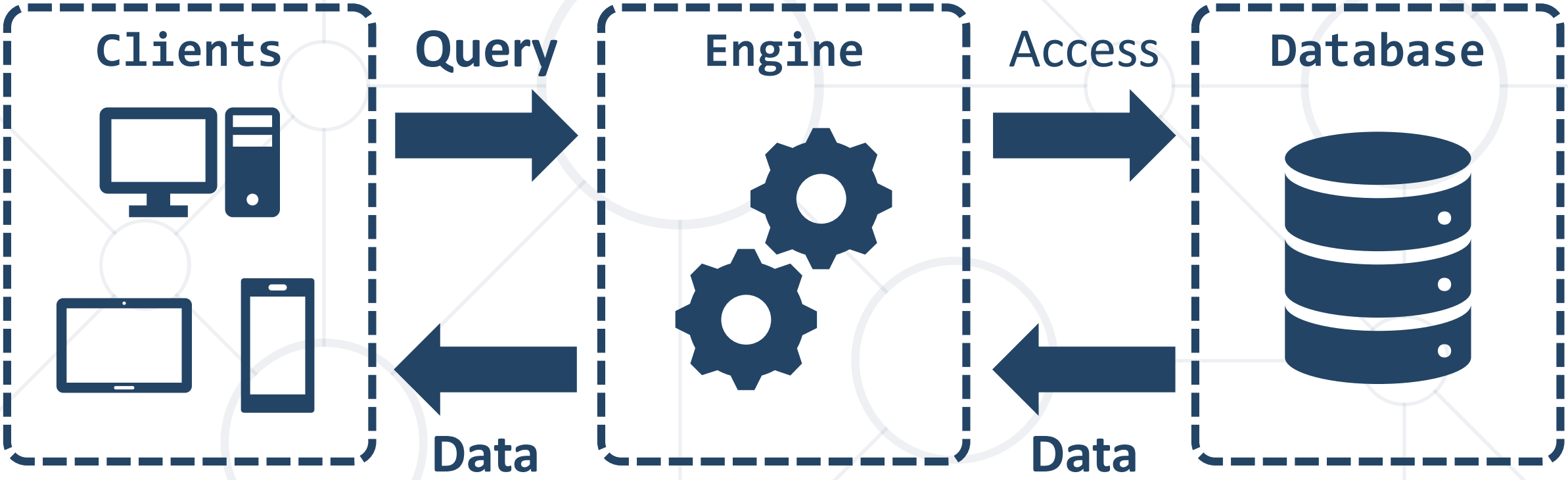
- Storing data is **not** the primary reason to use a Database
- Flat storage **eventually** runs into **issues** with
  - Size
  - Ease of updating
  - Accuracy
  - Security
  - Redundancy
  - Importance

Database Engines

Client-Server Model

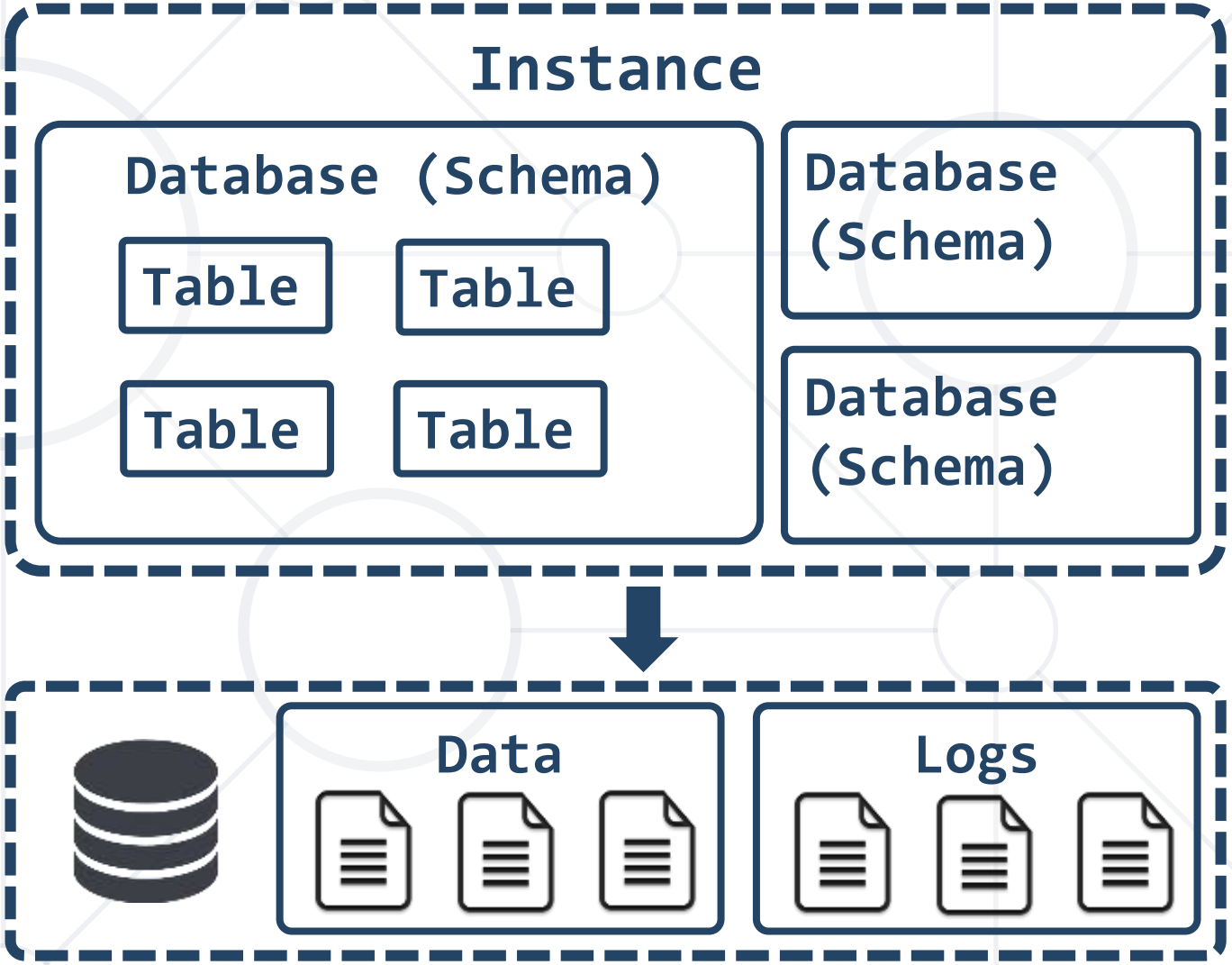
Database Engine Flow

- SQL Server uses the Client-Server Model



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

Column

Row

CustomerId	FirstName	Birthdate	CityId
1	August	03/12/1975	101
2	Bejnamin	04/04/1984	102
3	Denis	15/10/1988	103
4	Linda	07/01/1984	104

Cell

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

Structured Query Language

Query Components

Structured Query Language

- Programming language designed for managing data in a relational database
- To communicate with the Engine we use SQL
- 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**

MySQL

Working with Relational Databases

MySQL

- **Open-source** relational database management system
- Used in many **large-scale** websites
  - Amazon
  - Apple
  - Facebook
- Works on **many** system platforms – MAC OS, Windows, Linux

Developer Tools

- **XAMPP**
  - Web server stack
    - Apache + MariaDB + PHP + Perl
  - Cross-platform
- **HeidiSQL**
  - Tool using the popular MySQL, MSSQL and PostgreSQL
  - Can modify database
  - Easy to track database saves

Queries

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

```
CREATE DATABASE employees;
```

Database name

Creating Table and Inserting Values

SoftUni Foundation

- Table creation

Table name

```
CREATE TABLE people (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    email VARCHAR(40) NOT NULL,
    first_name VARCHAR(40) NOT NULL,
    last_name VARCHAR(40) NOT NULL
)
```

Custom attributes

Column name

Data type

- Inserting values

```
INSERT INTO `people` (`email`, `first_name`, `last_name`)
VALUES ('b@b.bg', 'John', 'Smith');
```

Retrieve Records

- Get all records from a table

```
SELECT * FROM people;
```

*** retrieves all columns**

- You can limit the number of columns

```
SELECT first_name, last_name FROM people;
```

List of columns

- You can limit the number of rows

```
SELECT TOP (5) fisrt_name, last_name FROM people;
```

Number of records

Update Records

SoftUni Foundation

- You can filter rows by specific conditions using the WHERE clause

```
SELECT last_name, age
FROM people
WHERE last_name = 'Pesho'
```

- Updating information

```
UPDATE people
SET last_name = 'Petrov'
WHERE first_name = 'Pesho'
```

Updates the last name of person

Comparing with NULL

- NULL is a special value that means missing value
  - Not the same as 0 or a blank space
- Checking for NULL values

This is always false!

```
SELECT last_name FROM Employees
WHERE email = NULL
```

```
SELECT last_name FROM Employees
WHERE email IS NULL
```

```
SELECT last_name FROM Employees
WHERE email IS NOT NULL
```

Deleting

- Deleting structures is called dropping
  - Table

```
TRUNCATE TABLE people;
```

Delete all records in a table

```
DROP TABLE people;
```

Delete the data and the structure

  - Entire Database

```
DROP DATABASE employees;
```

Delete entire database

- Both of these actions **cannot be undone**

NoSQL

Using MongoDB

NoSQL Database

- SQL query is **not used** in NoSQL systems
- More **scalable** and provide **superior performance**
- Such databases are **MongoDB**, **Cassandra**, **Redis**, etc
- Key-value **stores**

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

MongoDB

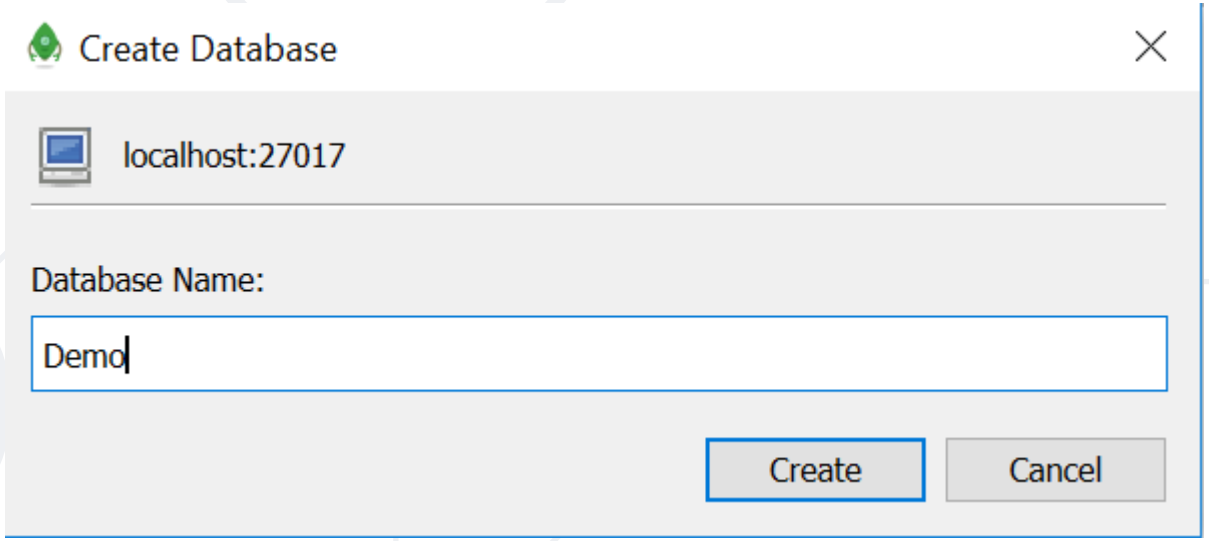
- Free **open-source** cross-platform document-oriented program
- Uses **JSON**-like documents with schemata.
- Good for **e-commerce** product catalog, blogs, evolving data requirements
- **Loosely coupled** objectives – the design **may change** by over time.

Developer Tools

- **Robo 3T**
  - Fully featured IDE with embedded shell
  - Visual Query Builder
  - IntelliShell with Auto-Completion
- Alternatives (**NoSQLBooster**)
  - Shell-centric cross platform GUI
  - Fluent Query Builder

Creating a Database

- Creating a database is done using the GUI
- Right click on **New Connection** and select **Create Database**



Creating Collection and Inserting Values

SoftUni Foundation

- Create collection

Collection name

```
db.createCollection('people')
```

- Inserting values

```
db.getCollection('people')
.insert({
    firstName: 'Michael',
    lastName: 'Smith',
    email: 'michael@gmail.com',
})
```

Data is inserted as **an object**

Retrieve Entries

SoftUni Foundation

- Get **all entries** from a collection

```
db.getCollection('people').find({})
```

- Filter elements by **given criteria**

```
db.getCollection('people').find({ firstName: 'Michael' })
```

- Return **specified** field

```
db.getCollection('people').find(
  { firstName: 'Michael' },
  { firstName: 1 }
)
```

Will retrieve an **entity** with the desired **field only**

Updating Entries

SoftUni Foundation

- Update **first** entry

```
db.getCollection('people').update(
  { firstName: 'Anne' },
  { firstName: 'George' }
)
```

Old value

New value

```
db.getCollection('people').update(
  { firstName: 'Anne' },
  { firstName: 'George' },
  { $multi: true }
)
```

Updates **all entries** with the given value

Deleting Entries

- Delete the **first entry** that matches given criteria

```
db.getCollection('people').deleteOne(
    { firstName: 'George' }
)
```

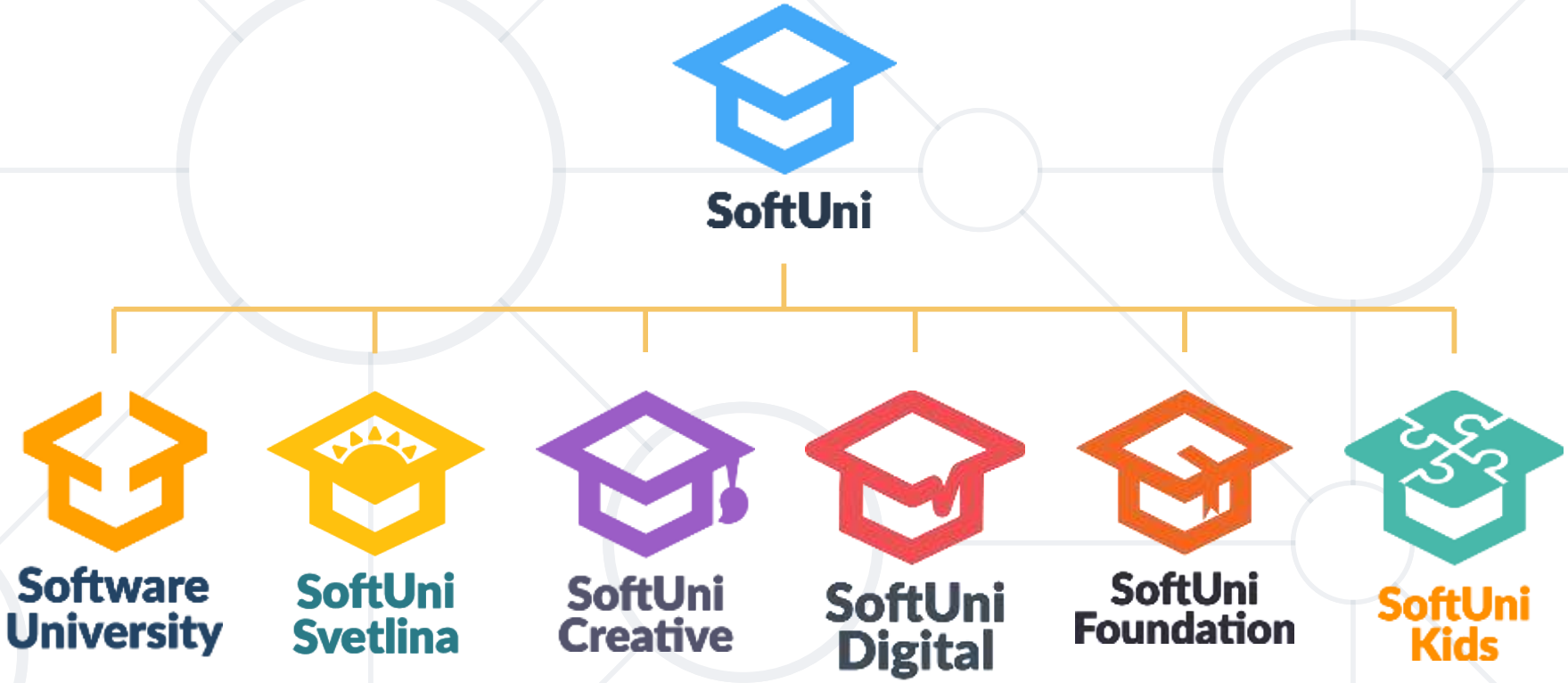
- Delete **all entries** that match given criteria

```
db.getCollection('people').deleteMany(
    { firstName: 'George' }
)
```

Summary

- RDBMS **store** and **manage data**
- We **communicate** with the DB engine via SQL
- **MySQL** is a multiplatform RDBMS **using SQL**
- **NoSQL** Databases are more **scalable**
- **MongoDB** stores entries in **JSON** format

Questions?



https://softuni.bg/courses/technology-fundamentals

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech
we know sports

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

codexio

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

WORLD OF MYTHS

Lukanet.com

codeXio

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license