Introduction to Databases

How do RDBMS work?

SoftUni Foundation

Java DB

SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Contents

Data Management

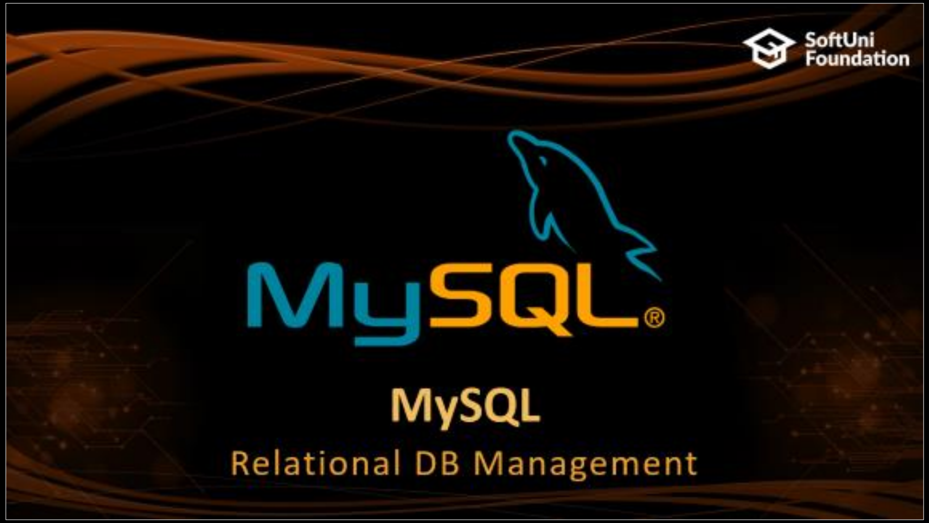
When Do We Need a Database?

Database Engine

Client-Server Model

Structured Query Language

Query Components



MySQL

Relational DB Management

Have a Question?

SoftUni Foundation

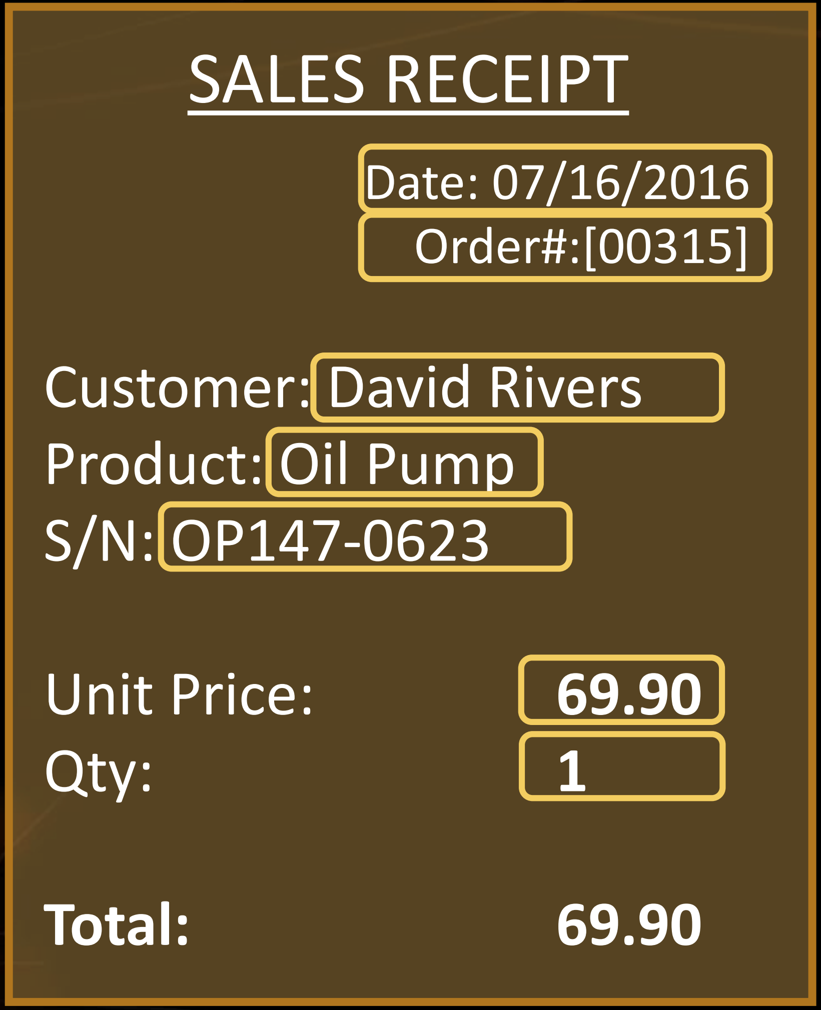
sli.do

#JavaDB

Data Management

When Do We Need a Database?

Storage vs. Management



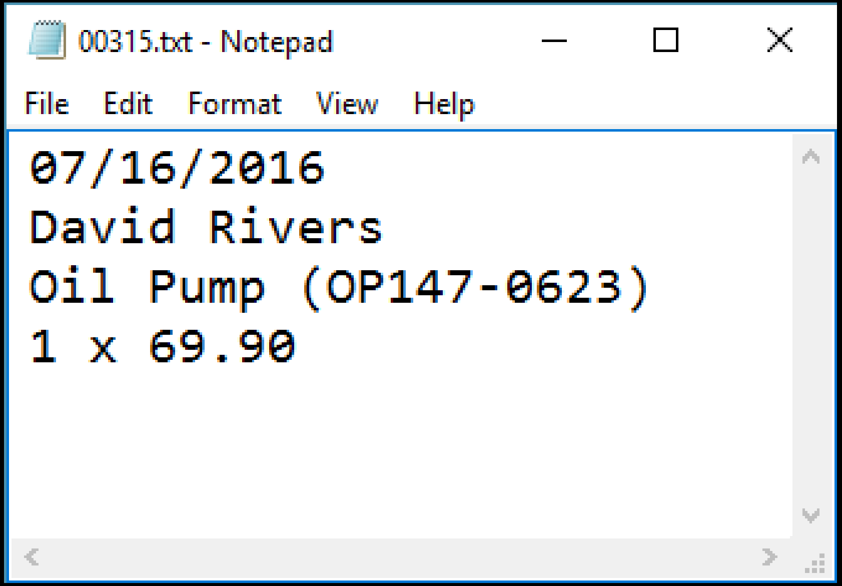
SALES RECEIPT

Date: 07/16/2016
Order#:[00315]

Customer:	David Rivers
Product:	Oil Pump
S/N:	OP147-0623
Unit Price:	69.90
Qty:	1
Total:	**69.90**

00315 – 07/16/2016
David Rivers
Oil Pump (OP147-0623)
1 x 69.90

Storage vs. Management (2)



```
07/16/2016
David Rivers
Oil Pump (OP147-0623)
1 x 69.90
```

Order#	Date	Customer	Product	S/N	Qty
00315	07/16/2016	David Rivers	Oil Pump	OP147-063	1

Storage vs. Management (3)

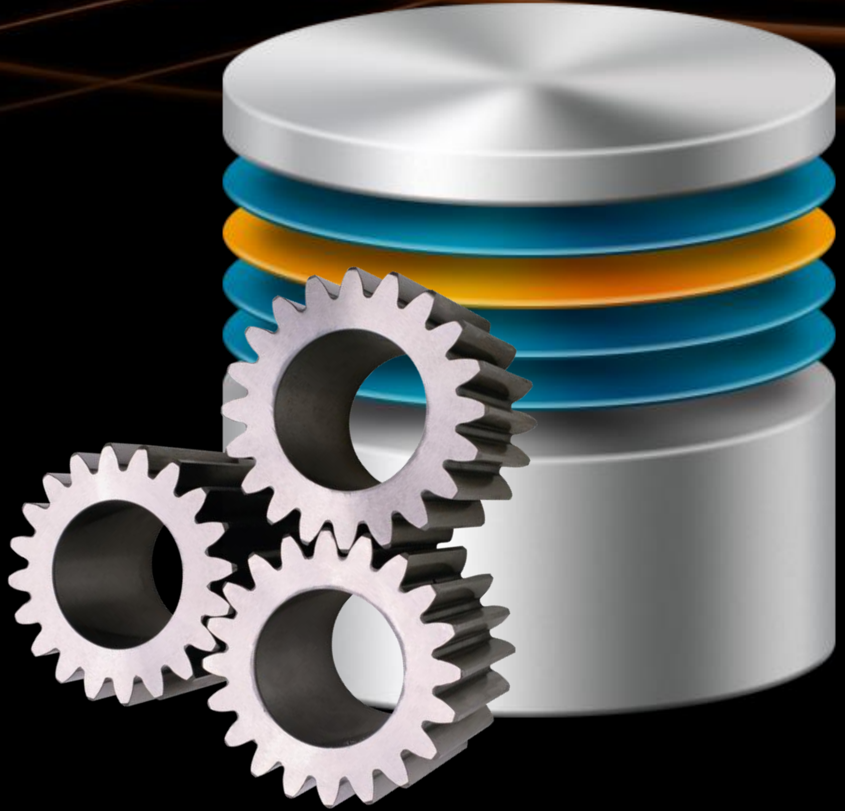
- Storing data is not the primary reason to use a database
- Flat storage eventually runs into issues with
  - Size
  - Ease of updating
  - Accuracy
  - Security
  - Redundancy
  - Importance

Databases

- A database is an organized collection of related information
  - It imposes rules on the contained data
  - Access to data is usually provided by a "system" (DBMS) database management
  - Relational storage first proposed by Edgar Codd in 1970

RDBMS

- Relational Data Base Management System
  - Database management
  - It parses requests from the user and takes the appropriate action
  - The user doesn't have direct access to the stored data
  - Data is presented by relations – collection of tables related by common fields
  - MS SQL Server, DB2, Oracle and MySQL

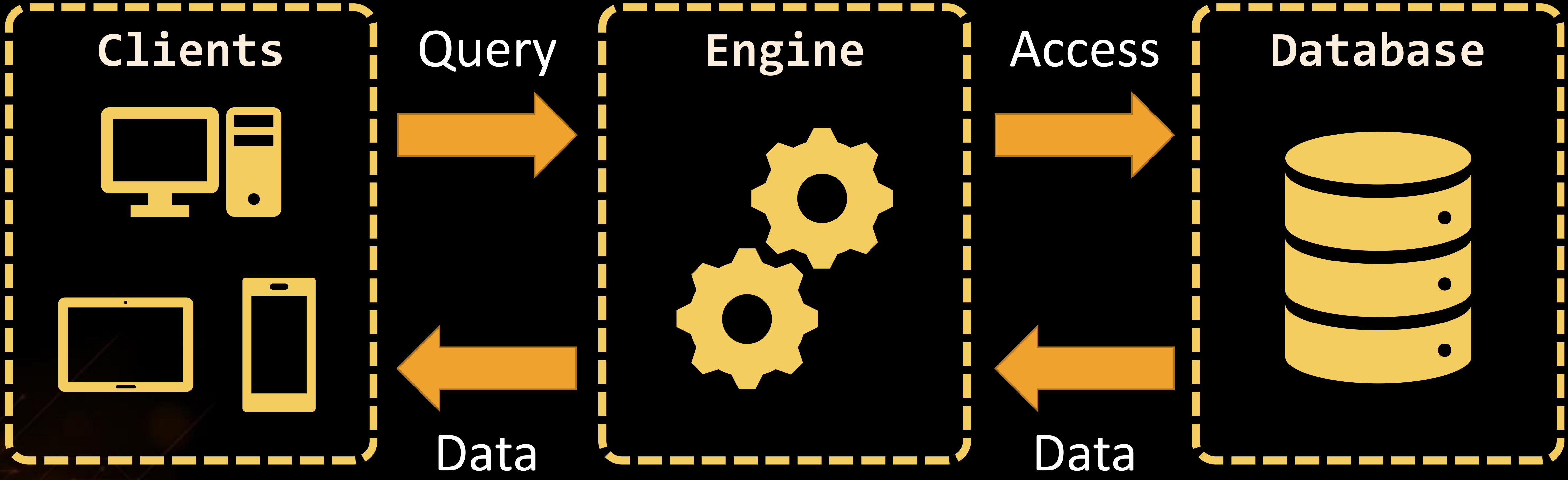


Database Engine

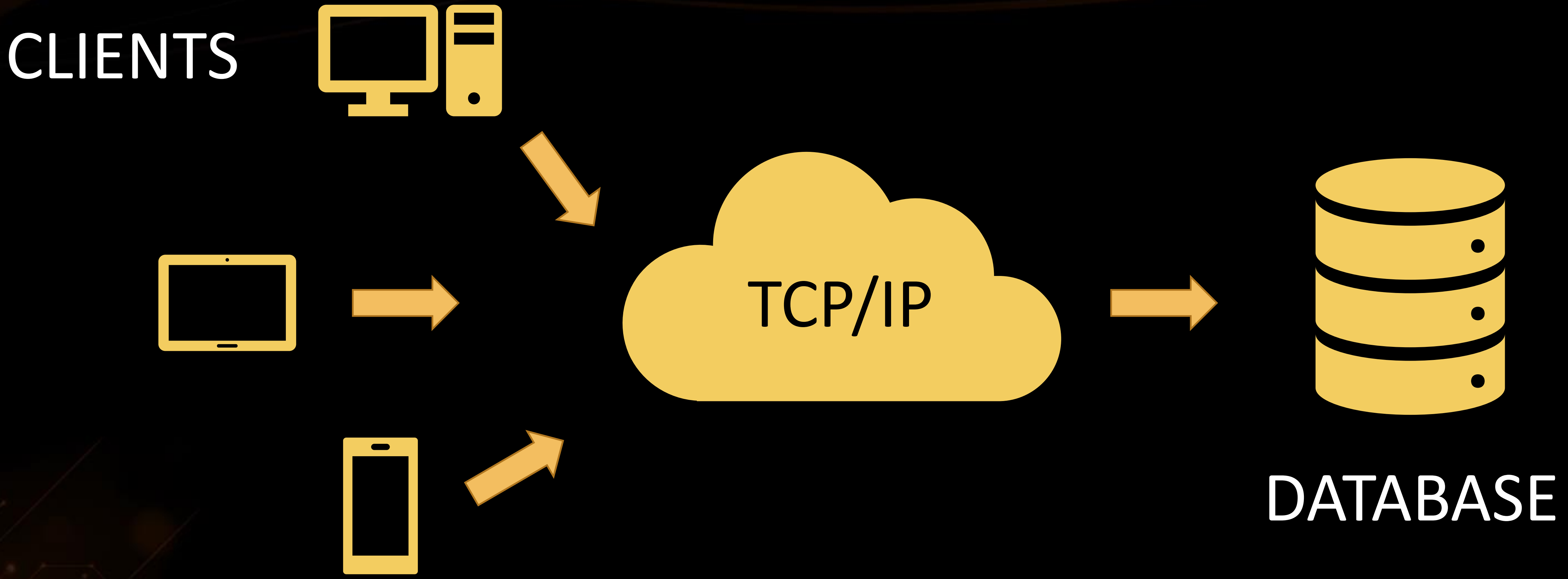
Client-Server Model

Database Engine Flow

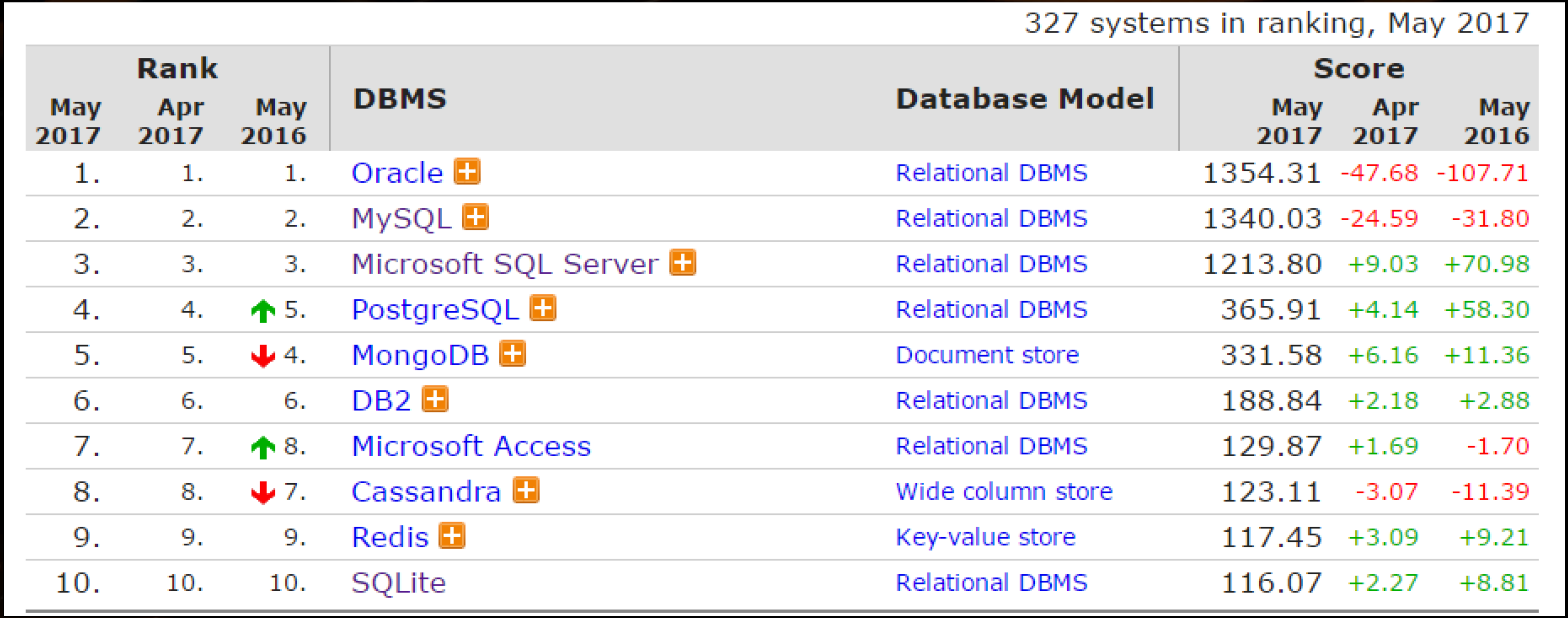
- SQL Server uses the Client-Server Model



Client-Server Model



Top Database Engines



327 systems in ranking, May 2017

Rank			DBMS	Database Model	Score		
May 2017	Apr 2017	May 2016			May 2017	Apr 2017	May 2016
1.	1.	1.	Oracle ⊞	Relational DBMS	1354.31	-47.68	-107.71
2.	2.	2.	MySQL ⊞	Relational DBMS	1340.03	-24.59	-31.80
3.	3.	3.	Microsoft SQL Server ⊞	Relational DBMS	1213.80	+9.03	+70.98
4.	4.	↑ 5.	PostgreSQL ⊞	Relational DBMS	365.91	+4.14	+58.30
5.	5.	↓ 4.	MongoDB ⊞	Document store	331.58	+6.16	+11.36
6.	6.	6.	DB2 ⊞	Relational DBMS	188.84	+2.18	+2.88
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	129.87	+1.69	-1.70
8.	8.	↓ 7.	Cassandra ⊞	Wide column store	123.11	-3.07	-11.39
9.	9.	9.	Redis ⊞	Key-value store	117.45	+3.09	+9.21
10.	10.	10.	SQLite	Relational DBMS	116.07	+2.27	+8.81

Source: http://db-engines.com/en/ranking

Structured Query Language

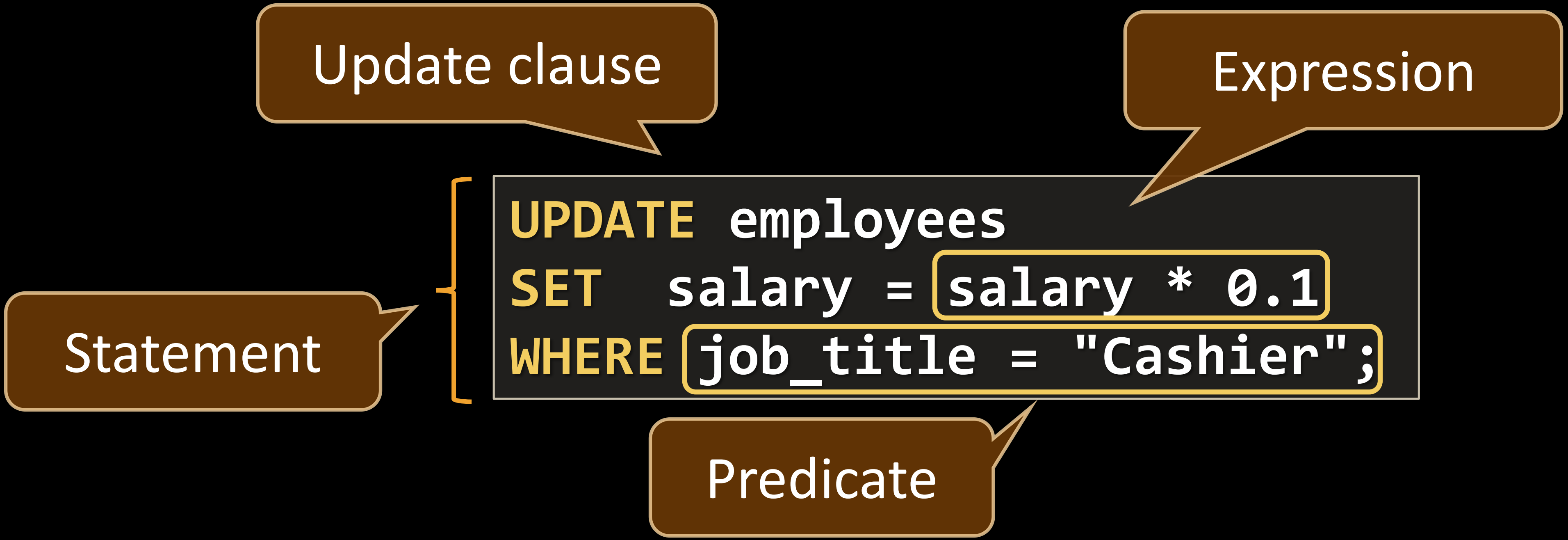
Query Components

Structured Query Language

- Programming language designed for managing data in a relational database
- Developed at IBM in the early 1970s
- To communicate with the Engine we use SQL

Structured Query Language (2)

- Subdivided into several language elements
  - Queries
  - Clauses
  - Expressions
  - Predicates
  - Statements



Update clause

Expression

Statement

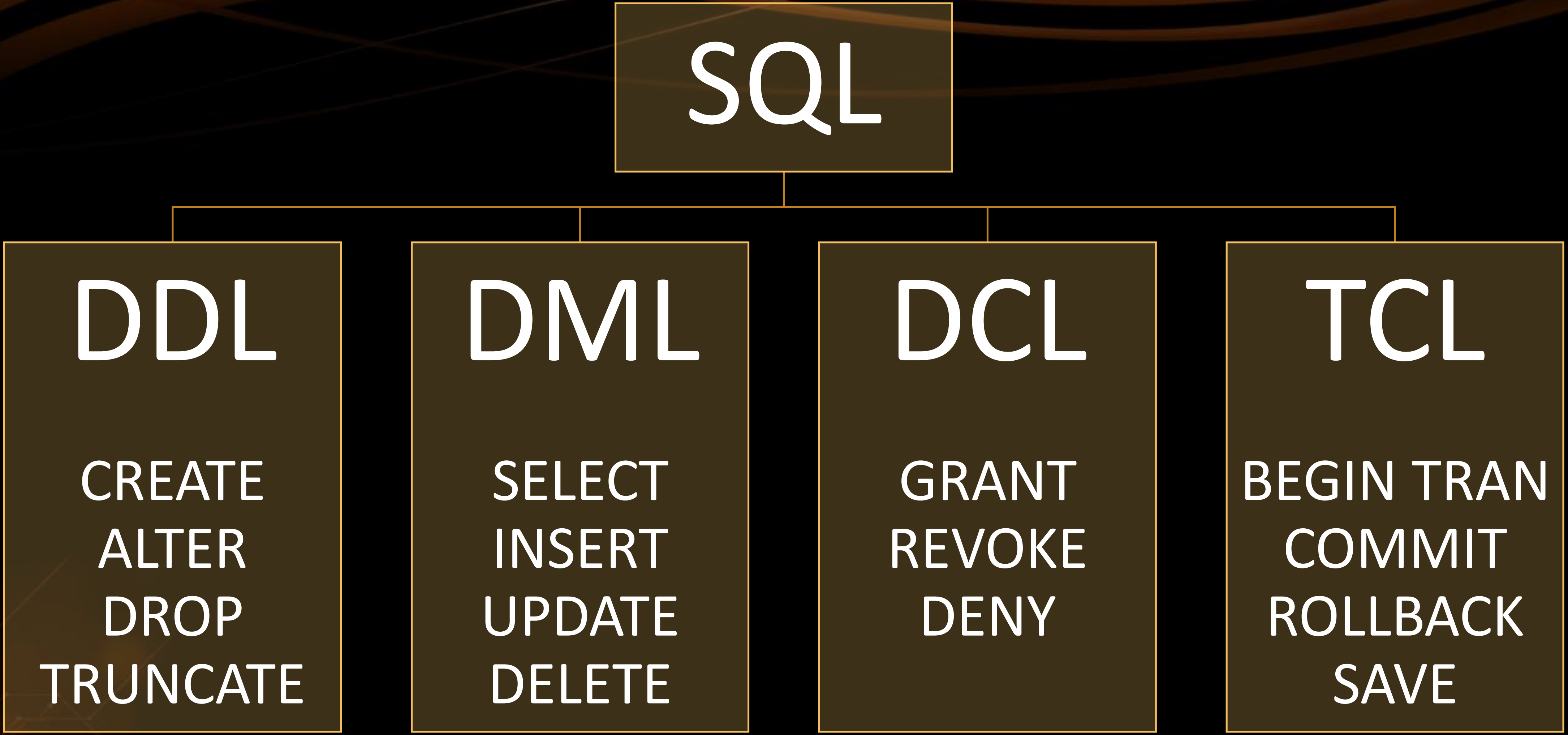
```
UPDATE employees
SET   salary = salary * 0.1
WHERE job_title = "Cashier";
```

Predicate

Structured Query Language (3)

- 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

Structured Query Language (4)



SQL

DDL

CREATE
ALTER
DROP
TRUNCATE

DML

SELECT
INSERT
UPDATE
DELETE

DCL

GRANT
REVOKE
DENY

TCL

BEGIN TRAN
COMMIT
ROLLBACK
SAVE

MySQL

Relational DB Management

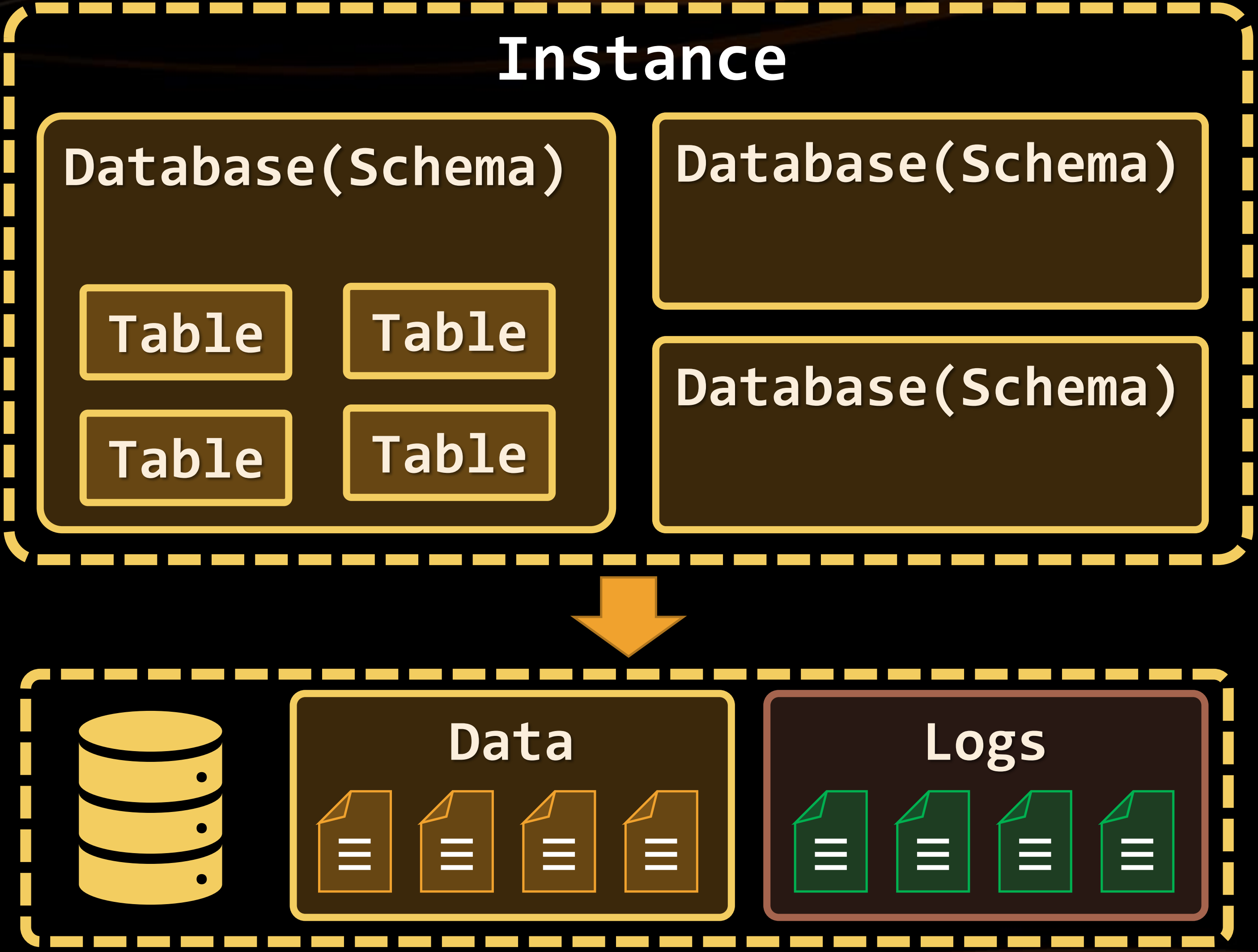
MySQL

- Open-source relational database management system
- Used in many large-scale websites like including Google Facebook, YouTube etc.
- Works on many system platforms –

  MAC OS, Windows, Linux
- Download MySQL Server
  - Windows: dev.mysql.com/downloads/windows/installer/
  - Ubuntu/Debian: dev.mysql.com/downloads/repo/apt/

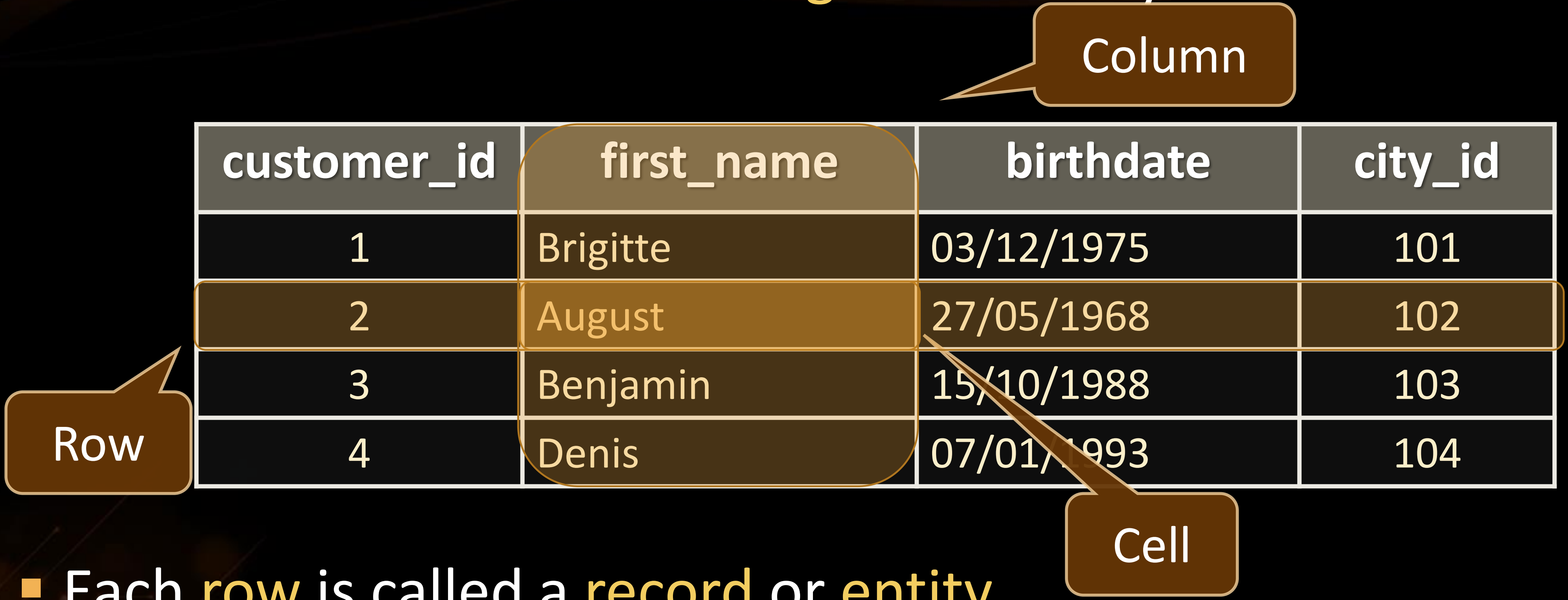
MySQL Server Architecture

- 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



customer_id	first_name	birthdate	city_id
1	Brigitte	03/12/1975	101
2	August	27/05/1968	102
3	Benjamin	15/10/1988	103
4	Denis	07/01/1993	104

Column

Row

Cell

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

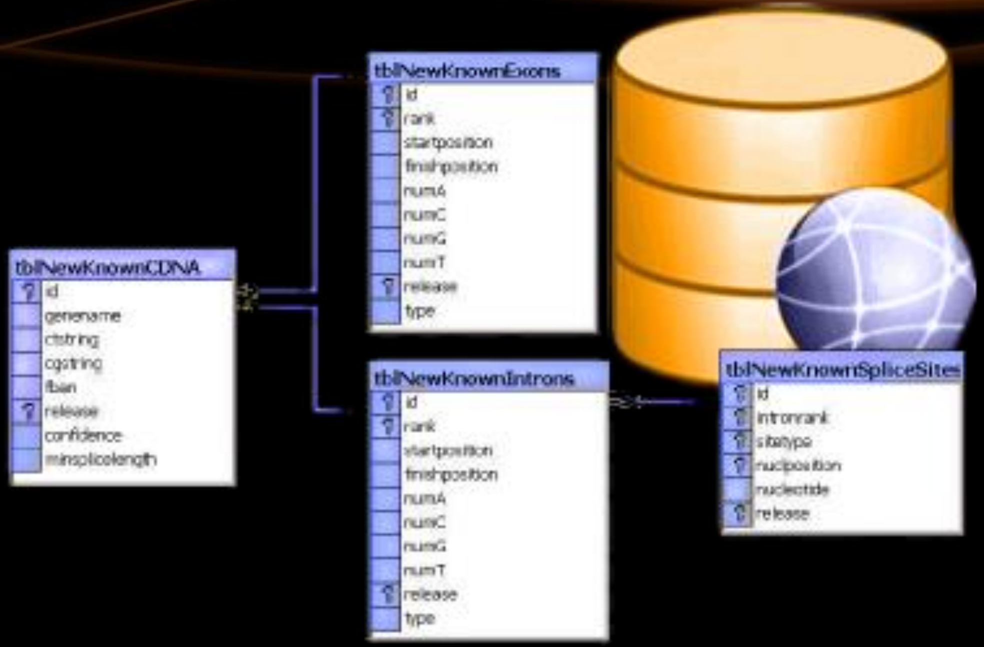


Table Relationships

Splitting data in tables

Why Split Related Data?

Empty records

first	last	registered	email	email2
David	Rivers	05/02/2016	drivers@mail.cx	david@homedomain.cx
Sarah	Thorne	07/17/2016	sarah@mail.cx	*NULL*
Michael	Walters	11/23/2015	walters_michael@mail.cx	*NULL*

Redundant information

order_id	date	customer	product	s/n	price
00315	07/16/2016	David Rivers	Oil Pump	OP147-0623	69.90
00315	07/16/2016	David Rivers	Accessory Belt	AB544-1648	149.99
00316	07/17/2016	Sarah Thorne	Wiper Fluid	WF000-0001	99.90
00317	07/18/2016	Michael Walters	Oil Pump	OP147-0623	69.90

Related Tables

- We split the data and introduce relationships between the tables to avoid repeating information

user_id	first	last	registered
203	David	Rivers	05/02/2016
204	Sarah	Thorne	07/17/2016
205	Michael	Walters	11/23/2015

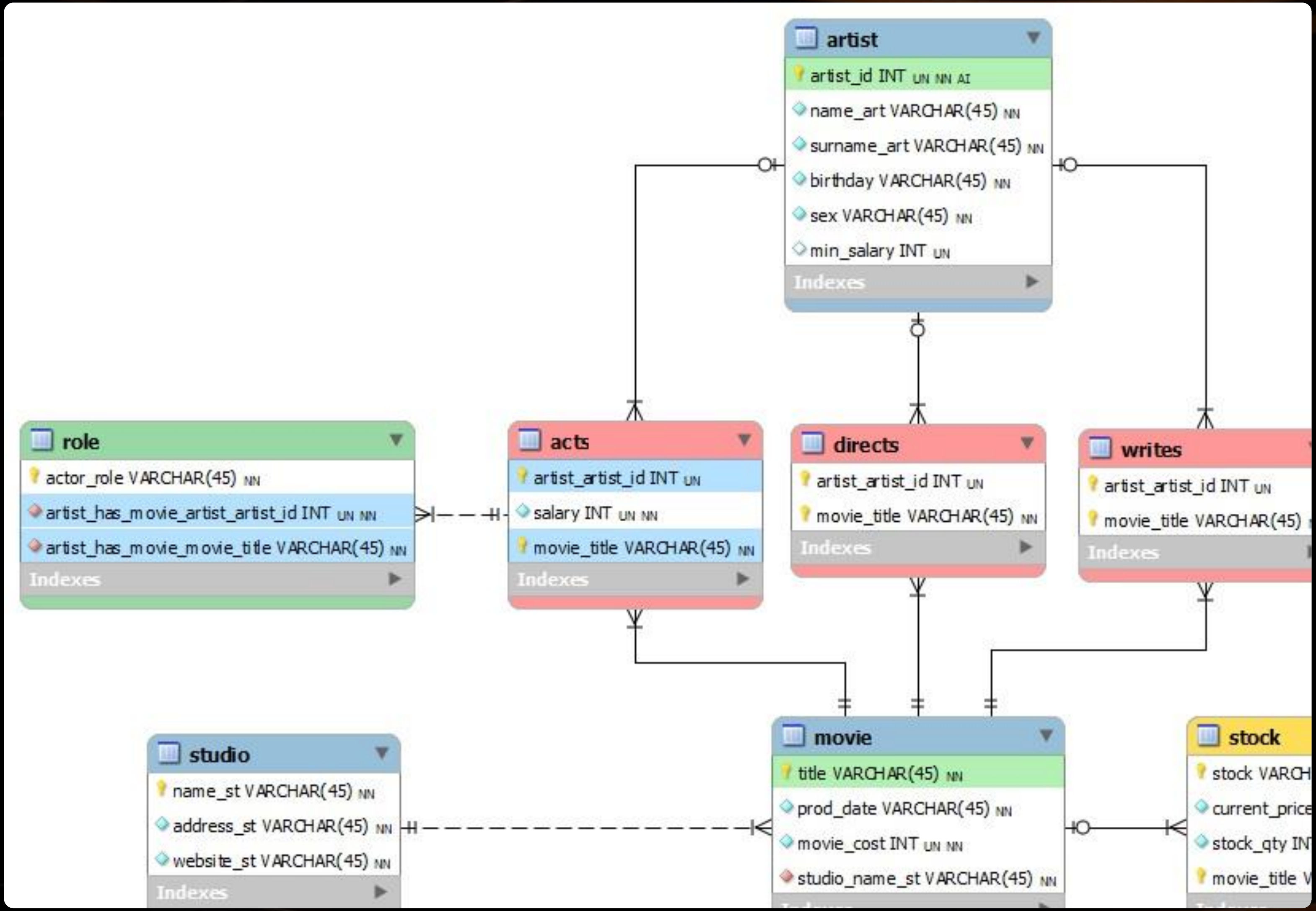
Primary Key

user_id	email
203	drivers@mail.cx
204	sarah@mail.cx
205	walters_michael@mail.cx
203	david@homedomain.cx

Foreign Key

- Connection via Foreign Key in one table pointing to the Primary Key in another

E/R Diagrams



artist
artist_id INT UN NN AI
name_art VARCHAR(45) NN
surname_art VARCHAR(45) NN
birthday VARCHAR(45) NN
sex VARCHAR(45) NN
min_salary INT UN
Indexes

role
actor_role VARCHAR(45) NN
artist_has_movie_artist_artist_id INT UN NN
artist_has_movie_movie_title VARCHAR(45) NN
Indexes

acts
artist_artist_id INT UN
salary INT UN NN
movie_title VARCHAR(45) NN
Indexes

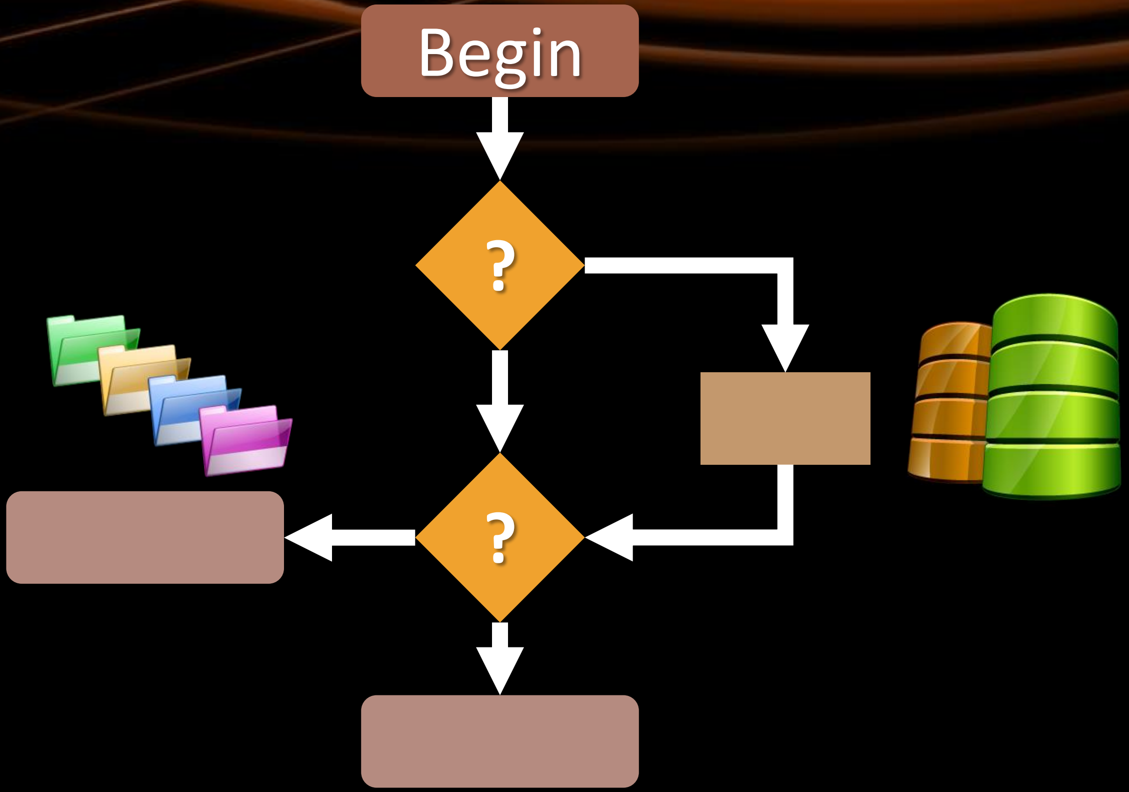
directs
artist_artist_id INT UN
movie_title VARCHAR(45) NN
Indexes

writes
artist_artist_id INT UN
movie_title VARCHAR(45)
Indexes

studio
name_st VARCHAR(45) NN
address_st VARCHAR(45) NN
website_st VARCHAR(45) NN
Indexes

movie
title VARCHAR(45) NN
prod_date VARCHAR(45) NN
movie_cost INT UN NN
studio_name_st VARCHAR(45) NN

stock
stock VARCH
current_price
stock_qty IN
movie_title V

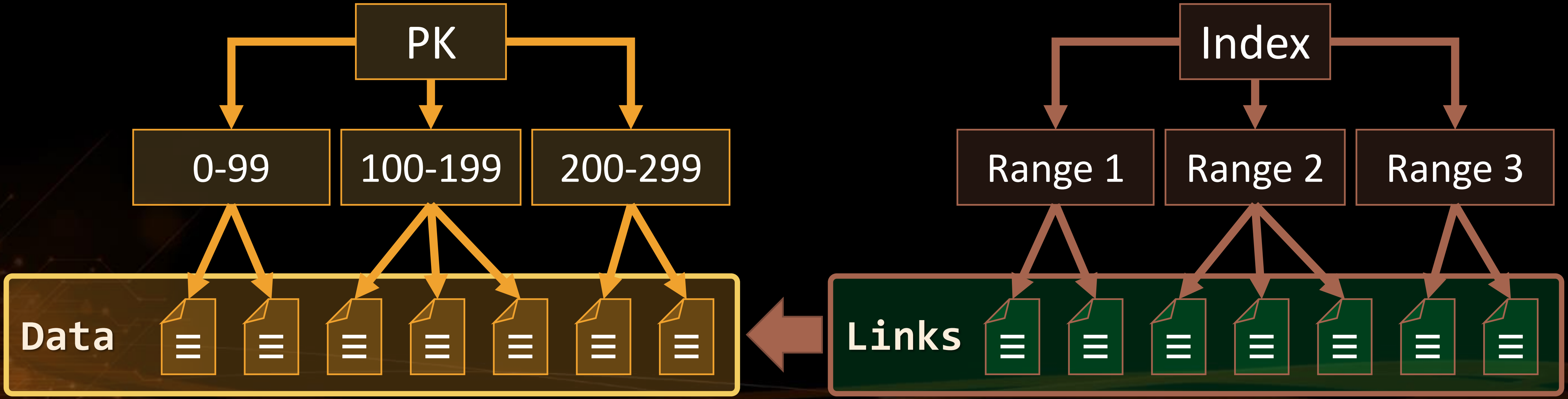


Programmability

Customizing Database Behavior

Indices

- Indices make data lookup faster
  - Clustered – bound to the primary key, physically sorts data
  - Non-Clustered – can be any field, references the primary index
- Structured as an ordered tree



Views

- Views are prepared queries for displaying sections of our data

```
CREATE VIEW v_employee_names AS
       SELECT e.employee_id,
              e.first_name,
              e.last_name
       FROM   soft_uni.employees AS e
```

```
SELECT * FROM v_employee_names
```

- Evaluated at run time – they do not increase performance

Procedures, Functions and Triggers

- A database can further be customized with reusable code
- Procedures – carry out a predetermined action
  - E.g. get all employees with salary above 35000
- Functions – receive parameters and return a result
  - E.g. get the age of a person using their birthdate and current date
- Triggers – watch for activity in the database and react to it
  - E.g. when a record is deleted, write it to an archive

Procedures

```
CREATE PROCEDURE udp_get_employees_salary_above_35000()
BEGIN
      SELECT first_name, last_name FROM employees
      WHERE salary > 35000;
END
```

```
CALL udp_get_employees_salary_above_35000
```

Functions

```
CREATE FUNCTION udf_get_age (dateValue DATE)
RETURNS INT
        BEGIN
        DECLARE result INT;
        SET result = TIMESTAMPDIFF(YEAR, dateValue, NOW());
        RETURN result;
        END
```

```
SELECT udf_get_age('1988-12-21');
```

Summary

- RDBMS store and manage data
- We communicate with the DB engine via SQL
- MySQL is a multiplatform RDBMS using SQL
- Table relations reduce repetition and complexity
- Databases can be customized with functions and procedures

Database Basics MySQL – Course Intro

SoftUni Foundation

xssoftware

SmartIT

NETPEAK
SEO and PPC for Business

SUPERHOSTING®.BG

INDEAVR
Serving the high achievers

telenor

SOFTWARE GROUP

INFRAGISTICS™
DESIGN / DEVELOP / EXPERIENCE

Questions?

https://softuni.bg/courses/databases-basics-mysql

License

Trainings @ Software University (SoftUni)

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

Software University

SoftUni Foundation

FORUM