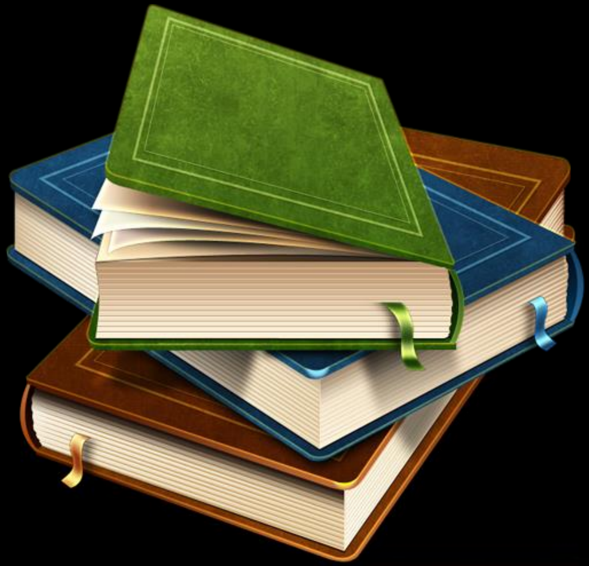
Using SQL

Connecting, Retrieving Data, Executing SQL Commands, …

Svetlin Nakov
Technical Trainer
www.nakov.com
Software University
http://softuni.bg

Table of Contents



What is Database?

What is database?

- Relational database is set of tables with defined relations between them
  - Each table has columns (fields) and rows
  - Some fields are called primary and foreign keys and define relation

Field

Row

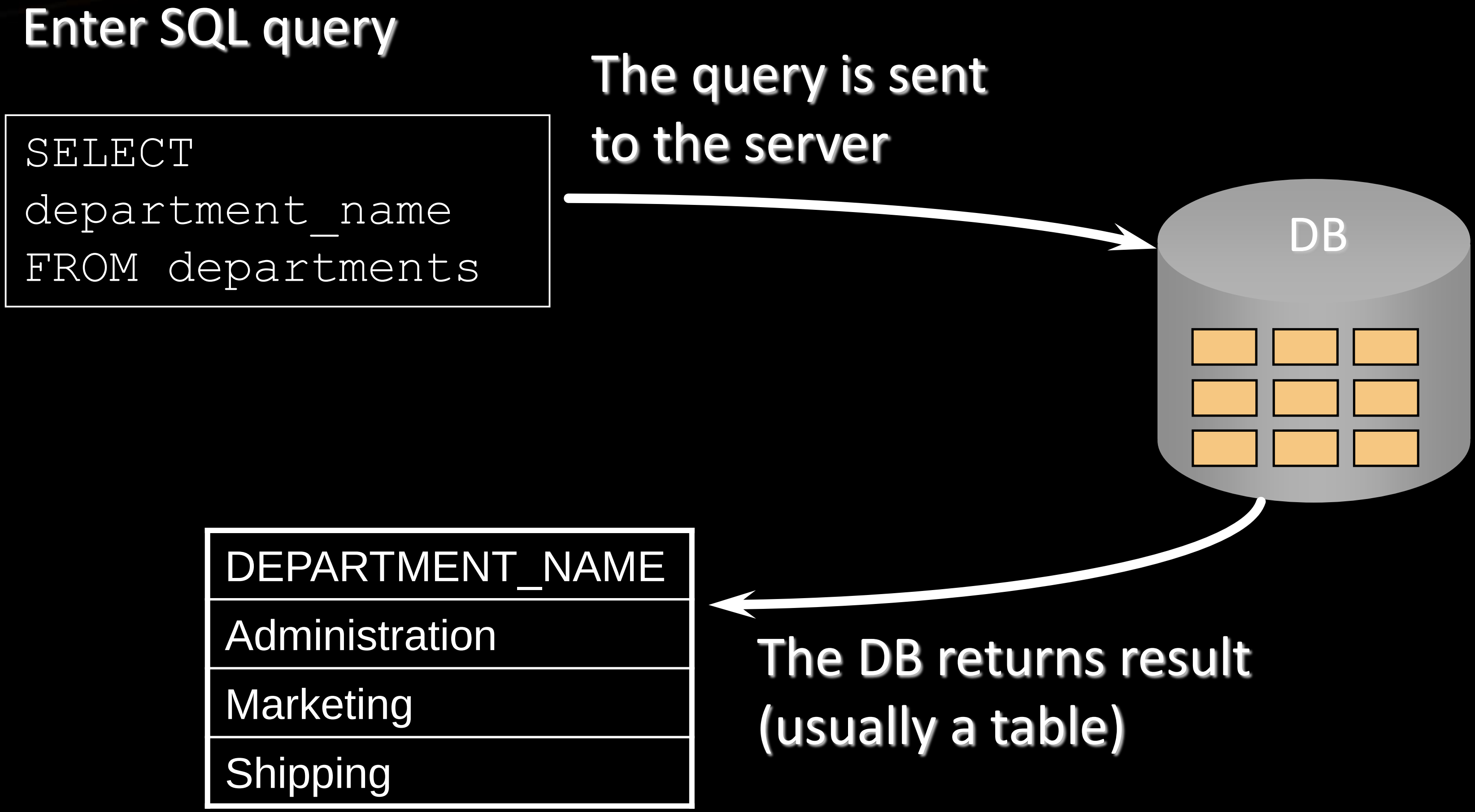
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	80
101	Neenah	Kochhar	17000	50
102	Lex	De Haan	(null)	90
103	Hunold	Alexander	9000	60
104	Ernst	Bruce	6000	90

What is SQL?

- Relational databases are manipulated using Structure Query Language (SQL)
  - Language for describing operations on structure and content of the database
  - Easy and straightforward to learn
  - Most databases follow the SQL standard 99 with little exceptions and additions
  - Uses English phrases and words:

```
SELECT department_name
FROM departments
```

Communication


Enter SQL query
The query is sent to the server
DB
The DB returns result (usually a table)

```
SELECT
department_name
FROM departments
```

DEPARTMENT_NAME
Administration
Marketing
Shipping

SQL

- SQL (Structured Query Language)
  - Language for describing and modifying database structure and data
  - Consists of DDL and DML
    - Data Definition Language (DDL) – defines the database structure – tables, fields and relations
    - Data Manipulation Language (DML) – modifies the data, stored in the tables – insert, delete, update or fetch rows

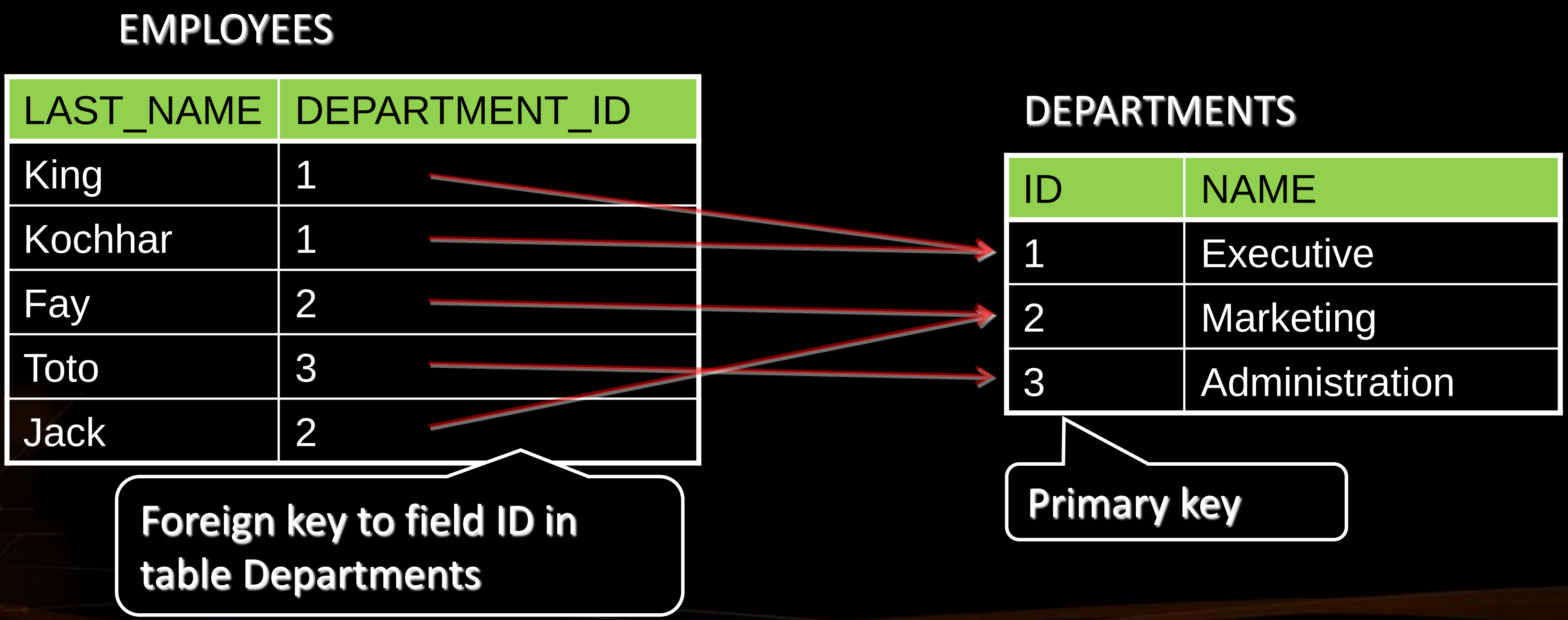
Keys and Table Relations

- Tables relations are defined by primary and foreign keys
  - Special properties of tables
  - Pair is formed by primary key in one table and linked foreign key in another
  - The values in a primary key field must be unique across the rows in the table
  - In a table there can be only one primary key but multiple foreign keys, pointing to other tables

Keys and Table Relations

Keys and Table Relations (2)

- Example of two tables with primary and foreign key
  - In table Employees we put the department id instead of all the information for the department
  - Data is not duplicated, less storage space required



EMPLOYEES

LAST_NAME	DEPARTMENT_ID
King	1
Kochhar	1
Fay	2
Toto	3
Jack	2

DEPARTMENTS

ID	NAME
1	Executive
2	Marketing
3	Administration

Types of Relations

- There are three types of relations between two tables
  - One-to-one – one row in the first table corresponds to single row in the other
  - One-to-many – one row in the first table corresponds to many rows in the other
  - Many-to-many – many rows in one table correspond to many rows in the other
    - Third table is needed to be achieved
    - Sum of two one-to-many relations

Fields Properties

- There are additional properties of the fields that change their behavior
  - Unique – requires the values in that field to be unique
    - Inserting or modifying value that already exists raises error
  - Index – modifies the internal work of the storage engine – speeds up searching for value in that field
    - Requires storage space

Fields Properties (2)

- Autoincrement – usually used for primary key fields; if the inserted value is NULL a new value is generated and used instead
- Not null fields – require the inserted value to be distinct from NULL
  - Raises error otherwise
  - All primary keys are not null
- MySQL supports also full text index – index for string fields

Data Manipulation Language

Select Query

Projection
Choosing set of columns

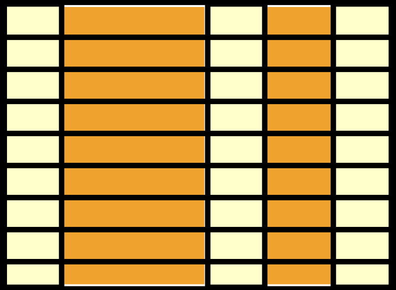


Table 1

Filtering
Choosing set of rows

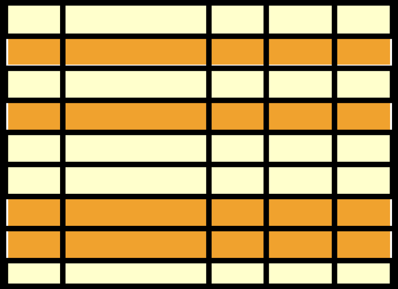


Table 1

Joining
Combining data from two or more tables

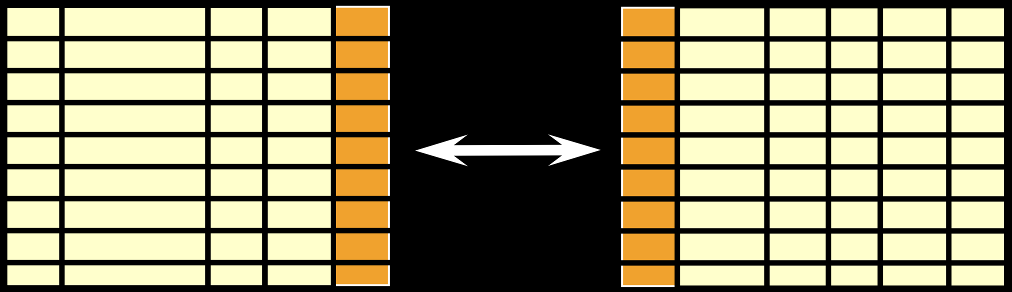


Table 1

Table 2

Select Query (2)

- Example select query:

```
SELECT
       EMPLOYEE_ID, FIRST_NAME as NAME,
SALARY
FROM    EMPLOYEES
WHERE EMPLOYEE_ID > 180
```

  - `EMPLOYEE_ID`, `FIRST_NAME`, `SALARY` – fields we are selecting
  - `as` sets name of the field in the result table
  - `From` defines the tables we are gathering the data from
  - `Where` filters the rows

Selecting all Fields

- Instead of list of fields to select `*` can be used to specify all fields
  - Example: table `employees`:

EMPL_ID	FIRST_NAME	LAST_NAME	SALARY
10	Larry	King	900
20	John	Kochhar	800
30	Papa	De Haan	850
50	Mimi	Tochkova	1200

```
SELECT * FROM EMPLOYEES
```

Is similar to query:

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
salary FROM EMPLOYEES
```

Selecting Fields

Live Demo

Filtering Rows

- To select from the employees table all employees with salary less than 1000:

```
SELECT FIRST_NAME, LAST_NAME, SALARY
FROM EMPLOYEES
WHERE SALARY < 1000
```

Produces result:

LAST_NAME	FIRST_NAME	SALARY
King	Larry	900
Kochhar	John	800
De Haan	Papa	850

Filtering Rows

Live Demo

The `null` Value

- The special value null means there is no value
  - Similar to PHP null
  - Different from zero or empty string
  - All operations with null produce null
    - Including comparison!

Strings

- Strings are enclosed in quotes
  - Some RDBMS support strings, enclosed in double-quotes
  - Example: selecting string

```
SELECT LAST_NAME, 'foo' AS FOO FROM EMPLOYEES
```

Produces result:

LAST_NAME	FOO
King	foo
Kochhar	foo
De Haan	foo
Mimi	foo

Selecting Only Distinct Rows

- The keyword distinct sets the database engine to return only distinct rows as result

```
SELECT MANAGER_ID,
SALARY
FROM EMPLOYEES
```

MANAGER_ID	SALARY
102	9000.00
103	4800.00
103	4800.00
103	4200.00

```
SELECT DISTINCT
MANAGER_ID,
SALARY
FROM EMPLOYEES
```

MANAGER_ID	SALARY
102	9000.00
103	4800.00
103	4200.00

Selecting Distinct Rows

Live Demo

Arithmetic Operations

- Arithmetic operations: - + * / ( )
- Example using in select query:

```
SELECT LAST_NAME, SALARY, SALARY + 300,
2*(SALARY + 300) AS BIG_SALARY
FROM EMPLOYEES WHERE SALARY < 1000
```

LAST_NAME	SALARY	SALARY + 300	BIG_SALARY
King	900	1200	2400
Kochhar	800	1100	2200
De Haan	850	1150	2300

String Operations

- Concatenation (joining) of strings is done by CONCAT()

```
SELECT concat(FIRST_NAME,' ',LAST_NAME) AS
Employees, SALARY
FROM EMPLOYEES
```

Employees	SALARY
Larry King	900
John Kochhar	800
Papa De Haan	850
Mimi Tochkova	1200

Comparison Operations

- Used in the where clause
  - Comparisons - <, >, <=, >=, <>
  - `BETWEEN value AND value` – similar to combination of comparisons
  - `IN (value, …)` – specifying if value is in a list
  - `LIKE, RLIKE` – simple and extended string comparison with regular expressions
  - `IS NULL, IS NOT NULL` – check if value is (not) null

Boolean Operations

- Used in where clauses
  - Logical operations – or, and, xor, not
  - Used to build complex filters for select query

```
SELECT
        MANAGER_ID,
        DEPARTMENT_NAME
FROM DEPARTMENTS
WHERE
        MANAGER_ID < 200 AND
        NOT (DEPARTMENT_NAME = 'SALES')
```

Boolean Operations

Live Demo

Sorting the Data

- Result of select query can be sorted via the `ORDER BY` clause
  - Syntax is:
    order by {column [asc|desc],...}

```
SELECT LAST_NAME, HIRE_DATE
FROM EMPLOYEES
ORDER BY HIRE_DATE, SALARY ASC
```

  - The asc and desc modifiers sort in ascending and descending order, respectively
  - By default sorting is ascending

Inserting Data Into Table

- The `insert` query has multiple forms:
  - `Insert into <table> values (<values>)`

```
INSERT INTO COUNTRIES
VALUES ('BG', 'Bulgaria', '1')

INSERT INTO COUNTRIES
(COUNTRY_ID,COUNTRY_NAME,REGION_ID)
VALUES ('BG', 'Bulgaria', '1')
```

Inserting Data Into Table

Live Demo

Modifying Data

- The update query modifies single or multiple rows in a table
  - The syntax is

```
update <table> set <column>=<value>,…
where <condition>
```

```
UPDATE EMPLOYEES SET
        FIRST_NAME = 'Updated Name',
        DEPARTMENT_ID = 90
WHERE EMPLOYEE_ID = 100
```

Modifying Data

Live Demo

Deleting Data

- The delete query deletes single or multiple rows from a table
  - Syntax is

```
delete from <table>
where <condition>
```

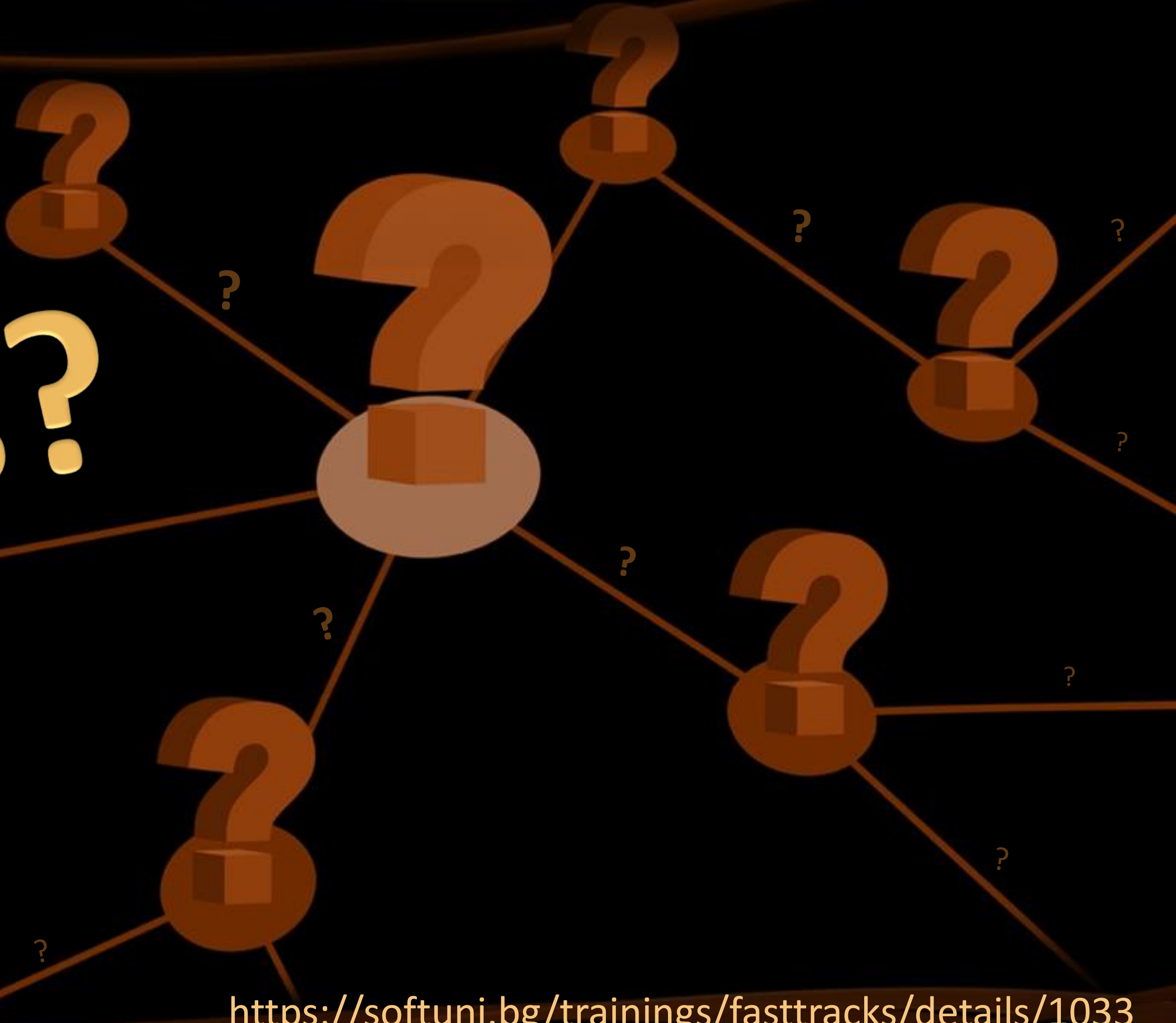
```
DELETE FROM EMPLOYEES WHERE EMPLOYEE_ID = 1

DELETE FROM EMPLOYEES WHERE FIRST_NAME LIKE
'S%'
```

- The `truncate` query empties table

```
TRUNCATE TABLE EMPLOYEES
```

SOFTWARE UNIVERSITY FOUNDATION



Questions?

https://softuni.bg/trainings/fasttracks/details/1033

License

Free Trainings @ Software University

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