Basic CRUD in MySQL Server

Create, Retrieve, Update, Delete - using SQL queries

SoftUni Team

Technical Trainers

Software University

SoftUni Foundation

Software University

http://softuni.bg

Table of Content

Have a Question?

SoftUni Foundation

sli.do

#JavaDB

Query Basics

SQL Introduction

SQL Queries – Few Examples

SoftUni Foundation

- Select first, last name and job title about employees:

```
SELECT first_name, last_name, job_title FROM employees;
```

- Select projects which start on 01-06-2003:

```
SELECT * FROM projects WHERE start_date='2003-06-01';
```

- Inserting data into table:

```
INSERT INTO projects(name, start_date)
VALUES('Introduction to SQL Course', '2006-01-01');
```

SQL Queries – Few Examples

SoftUni Foundation

- Update end date of specific projects:

```
UPDATE projects
   SET end_date = '2006-08-31'
 WHERE start_date = '2006-01-01';
```

- Delete specific projects:

```
DELETE FROM projects
       WHERE start_date = '2006-01-01';
```

Retrieving Data

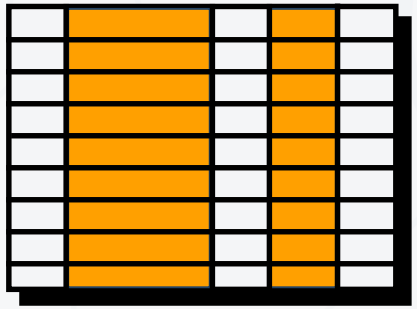
Using SQL SELECT

Capabilities of SQL SELECT

SoftUni Foundation

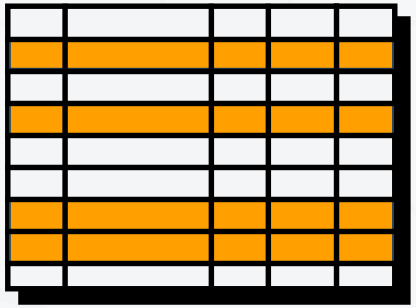
Projection

Take a subset of the columns



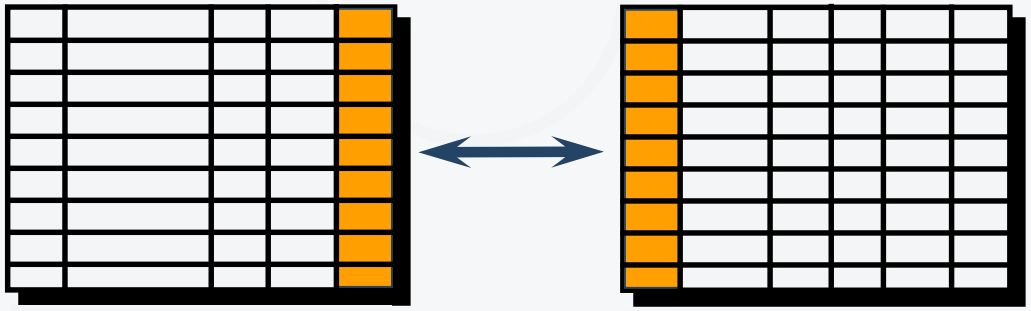
Selection

Take a subset of the rows



Join

Combine tables by some column



SELECT – Examples

SoftUni Foundation

- Selecting all columns from the "employees" table

id	first_name	last_name	job_title	department_id	salary
1	John	Smith	Therapist	1	900
2	John	Johnson	Acupuncturist	1	880
3	Smith	Johnson	Technician	2	1100
…	…	…	…	…	…

```
SELECT * FROM employees;
```

List of columns (* for all)

Table name

Problem: Select Employee Information

SoftUni Foundation

- Write a query to **select** all employees from "hospital" database
  - **Retrieve** information about their **id, first_name, last_name** and **job_title**
  - **Ordered** by id
- Note: Query **Hospital** database

id	first_name	last_name	job_title
1	John	Smith	Therapist
2	John	Johnson	Acupuncturist
3	Smith	Johnson	Technician
…	…	…	…

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#0

Solution: Select Employee Information

```
SELECT id, first_name, last_name, job_title
FROM employees
ORDER BY id;
```

List of columns

Table name

Aliases rename a table or a column heading:

```
SELECT e.id AS 'No.',
e.first_name AS 'First_Name',
e.last_name AS 'Last_Name',
e.job_title AS 'Job_Title'
FROM employees AS e ORDER BY id;
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#0

Concatenation

SoftUni Foundation

- You can concatenate column names or strings using the **concat()** function
- **concat()** - returns the string that results from concatenating the arguments.
  - String literals are enclosed in ['](**single quotes**)
  - Table and column names containing special symbols use [`] (**backtick**)

```
SELECT concat(`first_name`,' ',`last_name`) AS 'full_name',
     `job_title` as  'Job Title',
      `id` AS 'No.'
  FROM `employees`;
```

Concatenation(2)

- Another function of concatenation is **concat_ws()** - stands for concatenate with **separator** and is a special form of CONCAT().

Separator

Arguments

```
SELECT concat_ws(' ', `first_name`, `last_name`, `job_title`)
AS 'full_name',
`job_title` AS  'Job Title',
`id` AS 'No.'
  FROM `employees`;
```

- Skip any **NULL** values after the separator argument.

Problem: Select Employees with Filter

- Find information about all employees, listing their:
  - **Full Name**
  - **Job title**
  - **Salary**

- Use **concatenation** to display first and last names as **one field**
- Note: Query **Hospital** database

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#1

Solution: Select Employees with Filter

Concatenation

```
SELECT concat(`first_name`,' ',`last_name`) AS
    'full_name',
    `job_title` AS 'job_title',
      `salary` AS 'salary'
  FROM `employees` WHERE salary > 1000;
```

Column alias

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#1

Filtering the Selected Rows

SoftUni Foundation

- Use **DISTINCT** to eliminate duplicate results

```
SELECT DISTINCT `department_id`
  FROM `employees`;
```

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

```
SELECT `last_name`, `department_id`
FROM `employees`
WHERE `department_id` = 1;
```

- Other **logical operators** can be used for better control

```
SELECT `last_name`, `salary`
FROM `employees`
WHERE `salary` <= 20000;
```

Other Comparison Conditions

SoftUni Foundation

- Conditions can be combined using **NOT**, **OR**, **AND** and brackets

```
SELECT `last_name` FROM `employees`
WHERE NOT (`manager_id` = 3 OR `manager_id` = 4);
```

- Using **BETWEEN** operator to specify a range:

```
SELECT `last_name`, `salary` FROM `employees`
WHERE `salary` BETWEEN 20000 AND 22000;
```

- Using **IN / NOT IN** to specify a set of values:

```
SELECT `first_name`, `last_name`, `manager_id`
FROM `employees`
WHERE `manager_id` IN (109, 3, 16);
```

Problem: Select Employees by Multiple Filters

SoftUni Foundation

- Write a query to **retrieve** information about employees, order by id
  - who are in **department 4**
  - have salary **higher or equal to 1600**

id	first_name	last_name	job_title	department_id	salary
7	Jack	Jackson	Epidemiologist	4	1800
9	Nikolay	Ivanov	Nutrition Technician	4	1600

```
SELECT * FROM employees AS e
WHERE e.department_id = 4 AND e.salary >= 1600;
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#4

Comparing with NULL

SoftUni Foundation

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

```
SELECT `last_name`, `manager_id`
FROM `employees`
WHERE `manager_id` = NULL;
```

This is always false!

```
SELECT `last_name`, `manager_id`
FROM `employees`
WHERE `manager_id` IS NULL;
```

```
SELECT `last_name`, `manager_id`
FROM `employees`
WHERE `manager_id` IS NOT NULL;
```

Sorting with ORDER BY

- Sort rows with the **ORDER BY** clause
  - **ASC**: ascending order, default
  - **DESC**: descending order

ASC is the default sorting order

```
SELECT `last_name`, `hire_date`
FROM `employees`
ORDER BY `hire_date`;
```

LastName	HireDate
	1998-07-31
	1999-02-26
Tamburello	1999-12-12
…	…

```
SELECT `last_name`, `hire_date`
FROM `employees`
ORDER BY `hire_date` DESC;
```

LastName	HireDate
Valdez	2005-07-01
Tsoflias	2005-07-01
Abbas	2005-04-15
…	…

Views

- Views are **virtual tables** made from others tables, views or joins between them
- Usage:
  - To simplify writing complex queries
  - To limit access to data for certain users

Views (2)

Table 1		
Column 1	Column 2	Column 3

Table 2		
Column 1	Column 2	Column 3

v_table1_table2		
Column 1	Column 2	Column 3

Views - Example

SoftUni Foundation

- Get employee names and salaries, by department

```
CREATE VIEW `v_hr_result_set` AS
SELECT
    CONCAT(`first_name`,' ',`last_name`) AS 'Full Name', `salary`
FROM `employees` ORDER BY `department_id`;
```

```
SELECT * FROM `v_hr_result_set`;
```

Problem: Top Paid Employee

- Create a **view** that selects all information about the **top paid employee**
  - Name the view **`v_top_paid_employee`**
  - Note: Query **hospital** database

	id	first_name	last_name	job_title	department_id	salary
▶	8	Pedro	Petrov	Medical Director	3	2100

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#3

Solution: Top Paid Employee

```sql
CREATE VIEW `v_top_paid_employee`
AS
    SELECT * FROM `employees`
    ORDER BY `salary` DESC LIMIT 1;
```

Sorting column

Greatest value first

```sql
SELECT * FROM `v_top_paid_employee`;
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#3

Writing Data in Tables

Using SQL INSERT

Inserting Data

SoftUni Foundation

- The SQL **INSERT** command

```
INSERT INTO `towns` VALUES (33, 'Paris');
```

Values for all columns

```
INSERT INTO projects(`name`, `start_date`)
     VALUES ('Reflective Jacket', NOW())
```

Specify columns

- **Bulk data** can be recorded in a single query, separated by comma

```
INSERT INTO `employees_projects`
     VALUES (229, 1),
            (229, 2),
            (229, 3), …
```

Inserting Data (2)

SoftUni Foundation

- You can use existing records to create a **new table**

```
CREATE TABLE `customer_contacts`
AS SELECT `customer_id`, `first_name`, `email`, `phone`
FROM `customers`;
```

New table name

Existing source

- Or into an existing table

List of columns

```
INSERT INTO projects(name, start_date)
SELECT CONCAT(name,' ', ' Restructuring'), NOW()
FROM departments;
```

Modifying Existing Records

Using SQL UPDATE and DELETE

Updating Data

SoftUni Foundation

- The SQL **UPDATE** command

New values

```
UPDATE `employees`
   SET `last_name` = 'Brown'
 WHERE `employee_id` = 1;
```

```
UPDATE `employees`
   SET `salary` = `salary` * 1.10,
       `job_title` = CONCAT('Senior',' ', `job_title`)
 WHERE `department_id` = 3;
```

- Note: Don't forget the **WHERE** clause!

Problem: Update Employees Salary

- **Update** all employees salaries whose **job_title** is "**Therapist**" by **10%**.
  - **all salaries ordered ascending**

salary
880
990
1089
1100
1100
1500.23
1600
1800
2100

```
UPDATE employees
SET salary = salary + (salary * 0.1)
WHERE job_title = 'Therapist';
SELECT salary
FROM employees
ORDER BY salary ASC;
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#2

Deleting Data

- Deleting specific rows from a table

Condition

```
DELETE FROM `employees`
WHERE `employee_id` = 1;
```

  - Note: Don't forget the **WHERE** clause!

- Delete all rows from a table (**TRUNCATE** works faster than **DELETE**)

```
TRUNCATE TABLE users;
```

Problem: Delete From Table

- **Delete** all employees from the "**employees**" table who are in d epartment **2 or 1**.
- **Order** by id.

	id	first_name	last_name	job_title	department_id	salary
▶	4	Peter	Petrov	Supervisor	3	1100
	5	Peter	Ivanov	Dentist	4	1500.23
	7	Jack	Jackson	Epidemiologist	4	1800
	8	Pedro	Petrov	Medical Director	3	2100
	9	Nikolay	Ivanov	Nutrition Technician	4	1600

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#5

Solution: Delete From Table

Delete Data

```
DELETE FROM employees
WHERE department_id = 1
OR department_id = 2;
SELECT * FROM employees
```

OR Condition

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/731#5

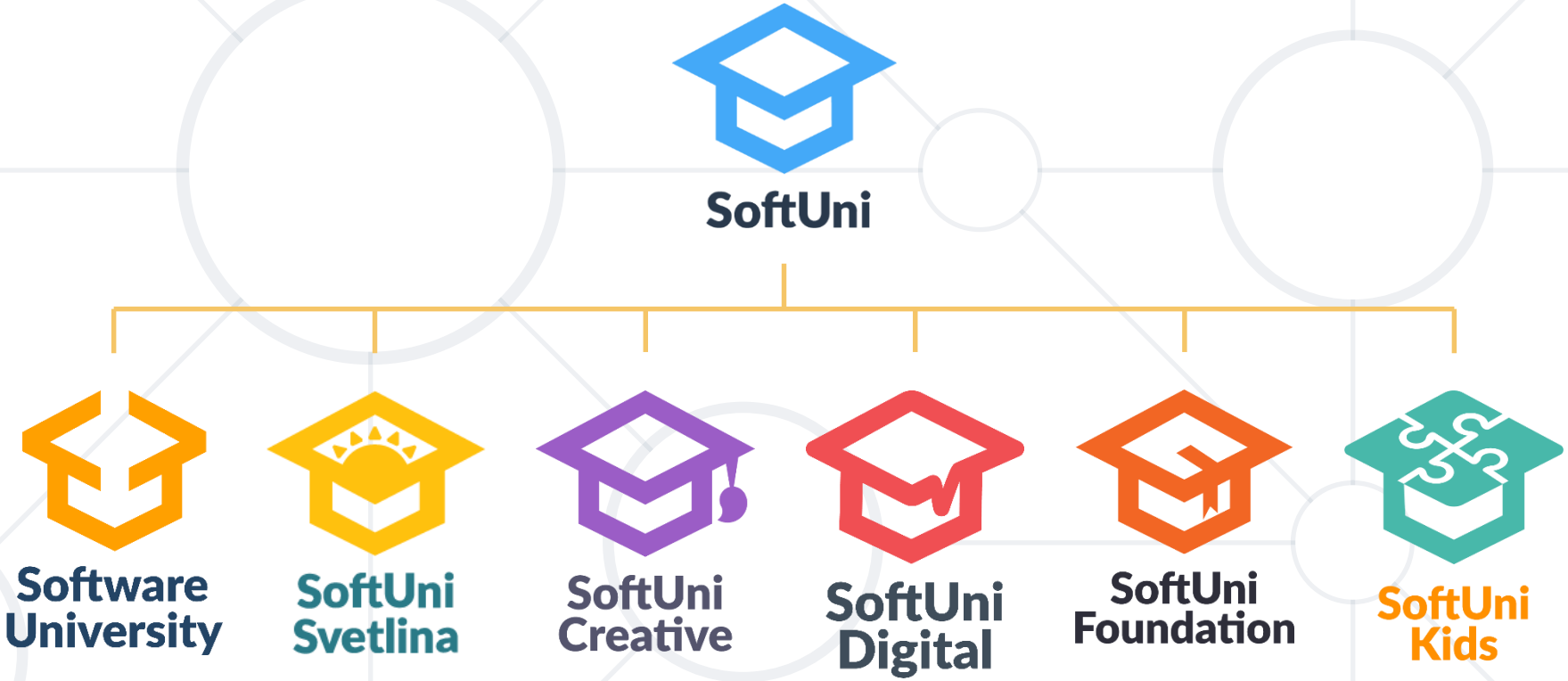
Summary

- We can easy manipulate our database with SQL queries

```
SELECT *
  FROM `projects`
 WHERE `start_date` = '2006-01-01';
```

- Queries provide a flexible and powerful method to manipulate records

Questions?



https://softuni.bg/trainings/courses

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech

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

One SOFTW

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