Basic CRUD in MySQL

Create, Retrieve, Update, Delete using SQL queries

Mariyan Apostolov

mariyan.apostolov89@gmail.com

CODERDOJO
<BULGARIA>

Table of Content

- Query Basics
- Retrieving Data
- Writing Data in Tables
- Modifying Existing Records

Query Basics

SQL Introduction

CODERDOJO <BULGARIA>

Query Basics

- 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=‘2019-03-01';
```

- Inserting data into table:

```
INSERT INTO projects(name, start_date)
VALUES(‘Muzeiko', ‘2019-03-01');
```

Query Basics

- Update end date of specific projects:

```
UPDATE projects
  SET end_date = ‘2019-04-01'
 WHERE start_date = ‘2019-03-01';
```

- Delete specific projects:

```
DELETE FROM projects
     WHERE start_date = ‘2019-04-01';
```

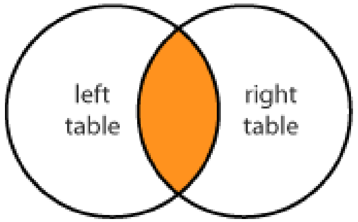
Retrieving Data

Using SQL SELECT

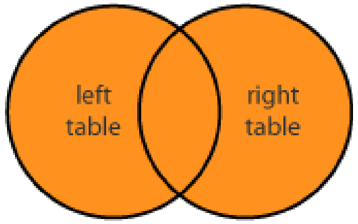
CODERDOJO <BULGARIA>

Capabilities of SQL SELECT

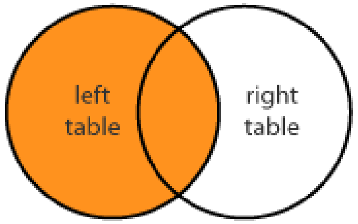
INNER JOIN



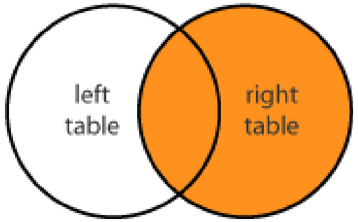
FULL JOIN



LEFT JOIN



RIGHT JOIN



CODERDOJO
<BULGARIA>

Column Aliases

- Aliases rename a table or a column heading

```
SELECT employee_id AS id, first_name, last_name
  FROM employees;
```

Display name

- You can shorten fields or clarify abbreviations

```
SELECT pr.name,
       pr.etc AS 'Elapsed Time Counter'
  FROM projects AS pr;
```

Concatenation

- You can concatenate column names or strings using the concat() function
  - 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`;
```

Filtering Selected Rows

- 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 greater control

```
… WHERE `salary` <= 20000;
```

Other Comparison Conditions

- Conditions ca be combined using NOT, OR, AND and brackets

```
… WHERE NOT (`manager_id` = 1 OR `manager_id` = 2);
```

- Using BETWEEN operator to specify a range:

```
… WHERE `salary` BETWEEN 20000 AND 22000;
```

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

```
… WHERE `manager_id` IN (17, 3);
```

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

```
... WHERE `manager_id` = NULL;
```

This is always false

```
... WHERE `manager_id` IS NULL;
```

```
... WHERE `manager_id` IS NOT NULL;
```

CODERDOJO <BULGARIA>

Sorting with ORDER BY

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

```
SELECT `last_name`, `hire_date`
FROM `employees`
ORDER BY `hire_date` LIMIT 1;
```

Greatest value first

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

Writing Data in Tables

Using SQL INSERT

CODERDOJO <BULGARIA>

Inserting Data

- The SQL INSERT command

```
INSERT INTO `towns` VALUES (12, 'Sofia');
```

```
INSERT INTO projects(`name`, `start_date`)
   VALUES ('Always Win', NOW())
```

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

```
INSERT INTO `employees_projects`
   VALUES (17, 1),
          (17, 2), …
```

Inserting Data

- You can use existing records to create a new table

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

- Or into an existing table

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

Modifying Existing Records

Using SQL UPDATE and DELETE

CODERDOJO <BULGARIA>

Updating Data

- The SQL UPDATE command

```
UPDATE `employees`
  SET `last_name` = 'Apostolov'
 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!**

Deleting Data

- Deleting specific rows from a table

```
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;
```

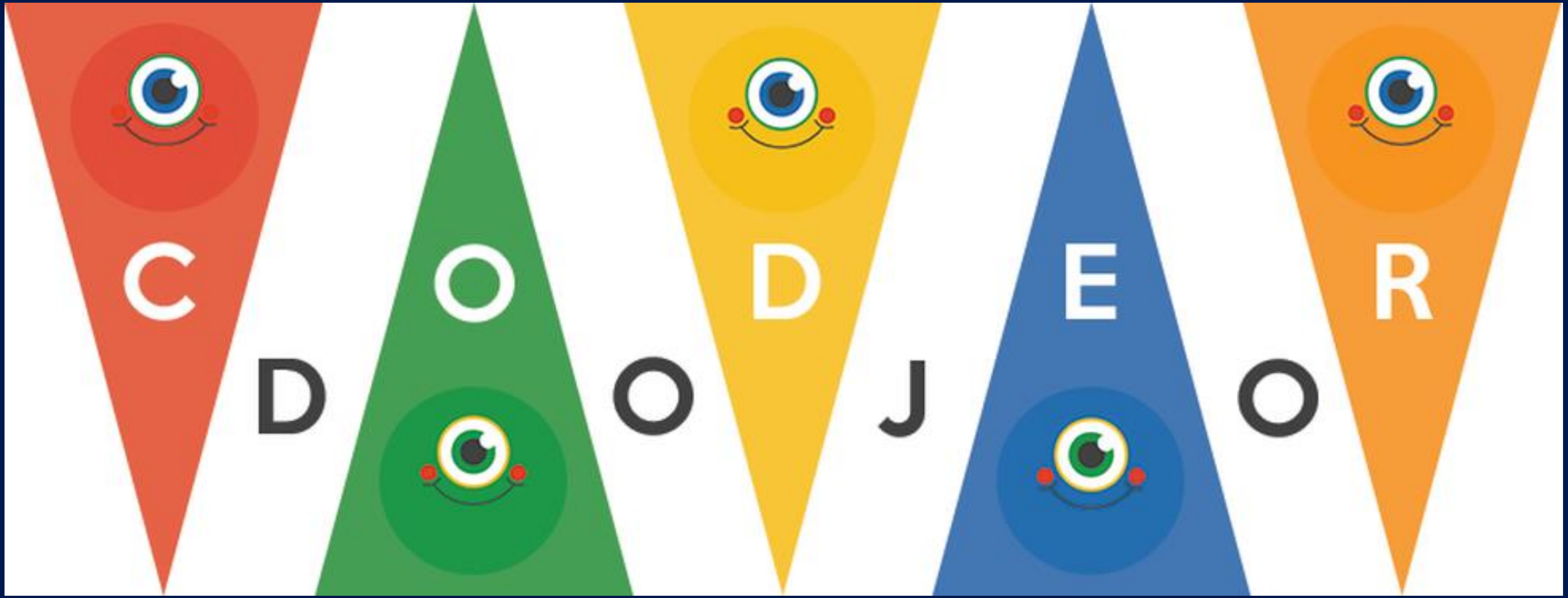
Thank You.

Mariyan Apostolov

mariyan.apostolov89@gmail.com

https://www.coderdojo.bg/

CODERDOJO <BULGARIA>



CODERDOJO
<BULGARIA>