**Lab: Built-in Functions**

This document defines the **lab** **exercise assignments** for the [PostgreSQL course @ Software University.](https://softuni.bg/trainings/4244/postgresql-september-2023)

Submit your solutions in the [SoftUni Judge System](https://judge.softuni.org/Contests/4104/Built-in-Functions-Lab).

Create database **book\_library** and open its **query tool.**

Download the file **book\_library.sql** from the course instance, import it into your database’s **query tab,** and **execute** the queries. Get familiar with the book\_library **tables**. You will use them in the following exercises.

1. **Find Book Titles**

Write a SQL query to find all **books** whose **titles start** with **"The"**. **Order** the result **by** id**.**

Submit your query statements**.**

**Example**

|  |
| --- |
| **title** |
| The Mysterious Affair at Styles |
| The Big Four |
| The Murder at the Vicarage |
| The Mystery of the Blue Train |
| The Ring |
| … |

1. **Replace Titles**

Write a SQL query to find all **books**, **which titles start with "The" and replace the substring with 3 asterisks.** Retrieve data about the updated titles. **Order the result by** id**.** Submit your query statements.

**Example**

|  |
| --- |
| **title** |
| \*\*\* Mysterious Affair at Styles |
| \*\*\* Big Four |
| \*\*\* Murder at the Vicarage |
| \*\*\* Mystery of the Blue Train |
| \*\*\* Ring |
| \*\*\* Alchemist |
| \*\*\* Fifth Mountain |
| \*\*\* Zahir |
| \*\*\* Dead Zone |
| \*\*\* Hobbit |
| \*\*\* Adventures of Tom Bombadil |

1. **Triangles on Bookshelves**

Write a SQL query to calculate the **area** of **triangles** with a given **side** and **height** from table **triangles**.

* Display the resulting table with columns **id** and **area. Order by id.**

Submit your query statements.

**Example**

|  |  |
| --- | --- |
| **id** | **area** |
| 1 | 4.0000000000000000 |
| 2 | 9.0000000000000000 |
| 3 | 6.7500000000000000 |
| 4 | 48.0000000000000000 |
| 5 | 7.5000000000000000 |

1. **Format Costs**

Write a SQL query to get each book’s **title** and **cost** (cost as **modified\_price)** and format the output to **3 digits after the decimal point**. **Order** by **id**.

Submit your query statements.

**Example**

|  |  |
| --- | --- |
| **title** | **modified\_price** |
| Unfinished Portrait | 15.990 |
| The Mysterious Affair at Styles | 17.990 |
| The Big Four | 14.990 |
| The Murder at the Vicarage | 13.990 |
| The Mystery of the Blue Train | 12.990 |
| … | … |

1. **Year of Birth**

Write a SQL query to get the **year of birth** for each **author**. Your query should return:

* **first\_name** – the first name of each author
* **last\_name** – the last name of each author
* **year** – the year of birth of each author

Submit your query statements.

**Example**

|  |  |  |
| --- | --- | --- |
| **first\_name** | **last\_name** | **year** |
| Agatha | Christie | 1890 |
| William | Shakespeare | 1564 |
| Danielle | Schuelein-Steel | 1947 |
| Joanne | Rowling | 1965 |
| Lev | Tolstoy | 1828 |
| … | … | … |

1. **Format Date of Birth**

Write a SQL query to display the **author’s** **last name** and **date of birth** in the format **15 (Mon) Sep 1890**.

* use date format: **DD (Dy) Mon YYYY**
* **born** field(formatted) as **Date of Birth**
* **last\_name** as **Last Name**

Submit your query statements.

**Example**

|  |  |
| --- | --- |
| **Last Name** | **Date of Birth** |
| Christie | 15 (Mon) Sep 1890 |
| Shakespeare | 26 (Sun) Apr 1564 |
| Schuelein-Steel | 14 (Mon) Jul 1947 |
| Rowling | 31 (Sat) Jul 1965 |
| Tolstoy | 09 (Tue) Sep 1828 |
| … | … |

1. **Harry Potter Books**

Write a SQL query to retrieve **titles** of all the **Harry Potter** books. **Order the information by** id**.**

Submit your query statements.

* Use the **WHERE** clause withthe **LIKE** operator.

**Example**

|  |
| --- |
| **title** |
| Harry Potter and the Philosophers Stone |
| Harry Potter and the Chamber of Secrets |
| Harry Potter and the Prisoner of Azkaban |
| … |