# Lab: Working with Queries in Django

Submit your solutions to the SoftUni [**Judge system**](https://alpha.judge.softuni.org/Contests/Working-with-Queries-in-Django-Lab/4325).

For this lab, you are given an **ORM project skeleton** (you can download it from the current lesson's resources) with **three models** called **"Author"**, **"Book"**, and **"Review"**:

* The **"Author"** model consists of the fields: **"first\_name"**, **"last\_name"**, **"birth\_date"**, **"nationality"**, and **"biography"**.
* The **"Book"** model consists of the fields: **"title"**, **"author"**, **"publication\_year"**, **"genre"**, **"language"**, and **"page\_count"**.
* The **"Review"** model consists of the fields: **"reviewer\_name"**, **"book\_title"**, **"author\_name"**, **"rating"**, **"comment"**, and **"created\_on"**.

In the **caller.py** file, you will find a function called **"add\_records\_to\_database"** on line 61. **Execute the function** to populate the tables with the needed data. Remember to **make it like a comment again**, so you do not overpopulate the tables while solving the problems.

## Books Finder

Create a function called **"find\_books\_by\_genre\_and\_language"** that:

* **Receives a book genre and a book language** as arguments.
* **Returns a queryset** of all books that concurrently satisfy **both** specified criteria - **genre** **and** **language**.

### Examples

|  |
| --- |
| **Test Code** |
| print(find\_books\_by\_genre\_and\_language("Romance", "English"))  print(find\_books\_by\_genre\_and\_language("Poetry", "Spanish"))  print(find\_books\_by\_genre\_and\_language("Mystery", "English")) |
| **Output** |
| <QuerySet [<Book: Love in Paris by Sarah Lee>]>  <QuerySet [<Book: Poems of the Heart by Maria Garcia>, <Book: Soulful Verses by Maria Garcia>, <Book: Whispers in the Wind by Maria Garcia>]>  <QuerySet [<Book: The Mystery of the Lost Key by John Smith>, <Book: The Enigmatic Riddle by John Smith>, <Book: Murder in the Mansion by Jane Johnson>]> |

## Find Authors' Nationalities

Create a function called **"find\_authors\_nationalities"** that:

* Finds all **authors** whose **nationalities** are **NOT null**.
* **Returns** **information** about each of them in the format:

**"{first\_name} {last\_name} is {nationality}"**

**...**

**"{first\_name} {last\_name} is {nationality}"**

### Examples

|  |
| --- |
| **Test Code** |
| print(find\_authors\_nationalities()) |
| **Output** |
| John Smith is American  Jane Johnson is British  Sarah Lee is Australian  Emily White is American  Laura Hall is American  John Grisham is American  Robert Miller is British |

## Order Books by Year

Create a function called **"order\_books\_by\_year"** that:

* **Orders** all **books** by their **publication year** in **ascending** **order**. If **two or more books are published** in the **same year**, order them by **title in ascending order** (alphabetically).
* **Returns** **information** about each book in the format:

**"{publication\_year} year: {title} by {author}"**

**...**

**"{publication\_year} year: {title} by {author}"**

### Examples

|  |
| --- |
| **Test Code** |
| print(order\_books\_by\_year()) |
| **Output** |
| 1813 year: Pride and Prejudice by Jane Austen  1988 year: The Alchemist by Paulo Coelho  2005 year: Fantasy World: The Quest Begins by Jane Johnson  2008 year: Poems of the Heart by Maria Garcia  2010 year: The Mystery of the Lost Key by John Smith  2012 year: Love in Paris by Sarah Lee  2013 year: The Enigmatic Riddle by John Smith  2015 year: Soulful Verses by Maria Garcia  2016 year: Murder in the Mansion by Jane Johnson  2018 year: Whispers in the Wind by Maria Garcia  2019 year: The Magic Kingdom by Alice Roberts  2021 year: 1984 by Anonymous Writer  2021 year: The Red Planet Expedition by Michael Brown  2022 year: To Kill a Mockingbird by Harper Lee |

## Delete Review by ID

Create a function called **"delete\_review\_by\_id"** that:

* **Receives a review's ID** as an argument.
* **Deletes the review's record** by the given ID.
* **Returns** **information** about the deleted review in the format:

**"Review by {reviewer\_name} was deleted"**

### Examples

|  |
| --- |
| **Test Code** |
| print(delete\_review\_by\_id(4))  print(delete\_review\_by\_id(1))  print(delete\_review\_by\_id(8)) |
| **Output** |
| Review by Samuel White was deleted  Review by Alice Johnson was deleted  Review by Daniel Harris was deleted |

## Filter Authors by Nationalities

Create a function called **"filter\_authors\_by\_nationalities"** that:

* **Receives a nationality** as an argument.
* **Filters** only the **authors** with the **given nationality** and **orders them by first name**, and then by **last name**.
* **Returns** **information** about each found **author's biography** in the format:

**"{biography1}"**

**...**

**"{biographyN}"**

If there is **NO biography added** for an author, return **information** **about their** **full name** in the format:  
**"{first\_name} {last\_name}"**

### Examples

|  |
| --- |
| **Test Code** |
| print("American authors:")  print(filter\_authors\_by\_nationalities('American'))  print()  print("British authors:")  print(filter\_authors\_by\_nationalities('British'))  print()  print("Authors with no nationalities:")  print(filter\_authors\_by\_nationalities(None)) |
| **Output** |
| American authors:  Emily White is a young adult fiction author, known for her coming-of-age stories.  John Grisham  John Smith  Laura Hall  British authors:  Jane Johnson is a renowned fantasy writer, famous for her epic fantasy series.  Robert Miller is a historical fiction writer, often exploring medieval themes.  Authors with no nationalities:  John Steinbeck was a renowned American author, famous for his classic novels.  Maria Garcia is a poet and writer, celebrated for her lyrical style.  Michael Brown is a science fiction author with a passion for space exploration. |

## Filter Authors by Birth Year

Create a function called **"filter\_authors\_by\_birth\_year"** that:

* **Receives two years** as two arguments.
* **Filters** the **authors who are born between the two given years (both inclusive)** and **order** **them** **by birth date** in **descending** **order**.
* **Returns** **information** about each found **author** in the format:

**"{birth\_date}: {first\_name} {last\_name}"**

**...**

**"{birth\_date}: {first\_name} {last\_name}"**

### Examples

|  |
| --- |
| **Test Code** |
| print("Authors born between 1980 and 2000:")  print(filter\_authors\_by\_birth\_year(1980, 2000))  print()  print("Authors born between 1950 and 1960:")  print(filter\_authors\_by\_birth\_year(1950, 1960))  print()  print("Authors born between 2000 and 2010:")  print(filter\_authors\_by\_birth\_year(2000, 2010)) |
| **Output** |
| Authors born between 1980 and 2000:  1992-03-12: Emily White  1990-02-10: Michael Brown  1982-08-04: Laura Hall  1980-05-15: John Smith  Authors born between 1950 and 1960:  1955-02-08: John Grisham  Authors born between 2000 and 2010: |

## Change Reviewer's Name

Create a function called **"change\_reviewer\_name"** that:

* Receives the **reviewer's name** as a first argument and a **new name** as a second argument.
* Changes **all occurrences** of the **reviewer's name** with the **new name**.
* **Returns** a queryset of all **reviews**.

### Examples

|  |
| --- |
| **Test Code** |
| print("Change Alice Johnson to A.J.:")  print(change\_reviewer\_name("Alice Johnson", "A.J."))  print()  print("Change Bob Wilson to Bobby W.:")  print(change\_reviewer\_name("Bob Wilson", "Bobby W."))  print()  print("Change A.J. to A. Johnson:")  print(change\_reviewer\_name("A.J.", "A. Johnson")) |
| **Output** |
| Change Alice Johnson to A.J.:  <QuerySet [<Review: Review by A.J.>, <Review: Review by Bob Wilson>, <Review: Review by A.J.>, <Review: Review by Samuel White>, <Review: Review by A.J.>, <Review: Review by A.J.>, <Review: Review by Carol Adams>, <Review: Review by Daniel Harris>, <Review: Review by Samuel White>]>  Change Bob Wilson to Bobby W.:  <QuerySet [<Review: Review by A.J.>, <Review: Review by Bobby W.>, <Review: Review by A.J.>, <Review: Review by Samuel White>, <Review: Review by A.J.>, <Review: Review by A.J.>, <Review: Review by Carol Adams>, <Review: Review by Daniel Harris>, <Review: Review by Samuel White>]>  Change A.J. to A. Johnson:  <QuerySet [<Review: Review by A. Johnson>, <Review: Review by Bobby W.>, <Review: Review by A. Johnson>, <Review: Review by Samuel White>, <Review: Review by A. Johnson>, <Review: Review by A. Johnson>, <Review: Review by Carol Adams>, <Review: Review by Daniel Harris>, <Review: Review by Samuel White>]> |