# PostgreSQL Regular Exam

Exam problems for the [PostgreSQL course @ Software University](https://softuni.bg/trainings/4244/postgresql-september-2023).

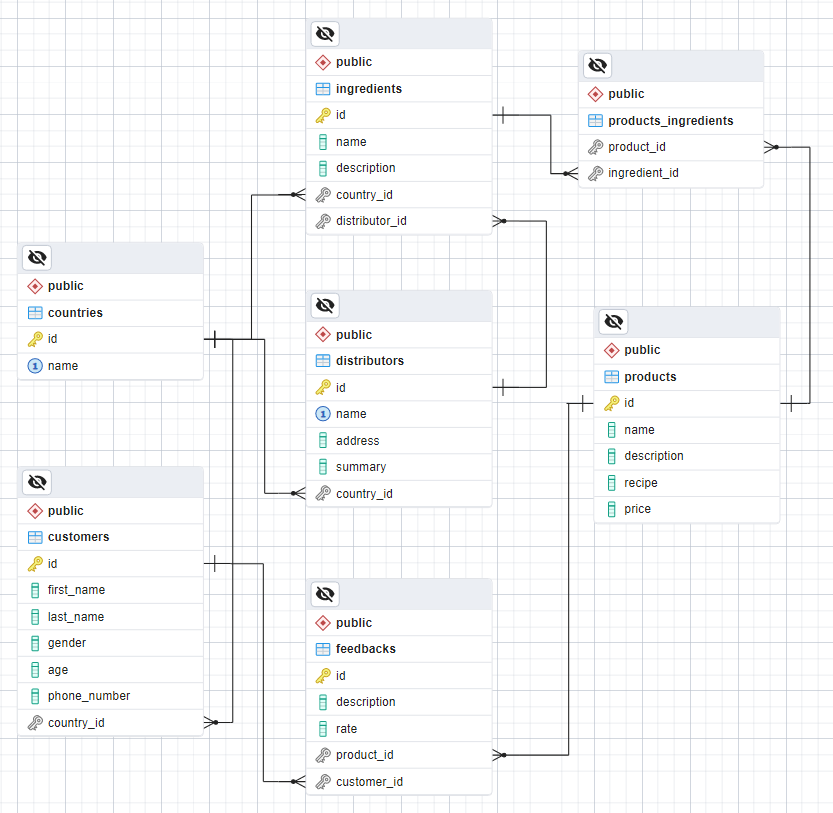
**Submit your solutions** in the SoftUni [Judge Contest](https://judge.softuni.org/Contests/4439/PostgreSQL-Retake-Exam-4-December-2023).

# Bio Bakery

# *In a small, charming town, Oliver, an aspiring baker, sets out to open his first Bio Bakery. He needs your help as a programmer to create an efficient database to help him keep track of ingredients for his healthy pastries and manage customer service. The bakery has become a hit, offering delicious and personalized treats to the delighted community. And your partnership brings both prosperity and useful pleasures to the city.*

# Section 1. Data Definition Language (DDL) - (30 pts)

You've been provided with the E/R Diagram representing the **Bio Bakery database**, showcasing how the various entities within the **Bio Bakery** are interconnected. This diagram provides a visual representation of the database's structure.



Create a PostgreSQL database named **"bio\_bakery\_db"** that comprises seven tables:

* **"countries"** - contains information about the place of origin (for ingredients), the central office (for distributors), or the home country (for customers);
* **"customers"** - stores data about our product users;
* **"products"** - contains information about the bakery's product offerings;
* **"feedbacks"** - stores details of customer feedback when evaluating certain products;
* **"distributors"** - contains information about organizations that supply ingredients;
* **"ingredients"** - holds details about fruits, vegetables, spices, and other components;
* **"products\_ingredients"** - serves as a many-to-many mapping table linking products and ingredients;

You are assigned with the responsibility of building the database tables based on the provided models:

### countries

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **id** | **Integer** from **0** to **2,147,483,647** | Primary Key, Unique table identification, Auto-increment |
| **name** | **String** up to **50** symbols | It has a **"UNIQUE"** constraint, **NULL** is **not** allowed |

### customers

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **id** | **Integer** from **0** to **2,147,483,647** | Primary Key, Unique table identification, Auto-increment |
| **first\_name** | **String** up to **25** symbols | **NULL** is **not** allowed |
| **last\_name** | **String** up to **50** symbols | **NULL** is **not** allowed |
| **gender** | **String** limited to **1** character | The **gender** can only have one of two values: **'M'** (for male) or **'F'** (for female), **NULL** is permitted |
| **age** | **Integer** from **0** to **2,147,483,647** | The column must always have a value **greater than zero**  **NULL** is not allowed |
| **phone\_number** | **String** limited to **10** character | **NULL** is **not** allowed |
| **country\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **countries**, Cascade Operations, **NULL** is **not** allowed |

### products

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **id** | **Integer** from **0** to **2,147,483,647** | Primary Key, Unique table identification, Auto-increment |
| **name** | **String** up to **25** symbols | **NULL** is **not** allowed |
| **description** | **String** up to **250** symbols | **NULL** is permitted |
| **recipe** | **Text** with unlimited numbers of symbols | **NULL** is permitted |
| **price** | **Numeric** number with a precision of **10** digits, including **2** digits after the decimal point | The column must always have a value **greater than zero**,  **NULL** is not allowed |

### feedbacks

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **id** | **Integer** from **0** to **2,147,483,647** | Primary Key, Unique table identification, Auto-increment |
| **description** | **String** up to **255** symbols | **NULL** is permitted |
| **rate** | **Numeric** number with a precision of **4** digits, including **2** digits after the decimal point | The **rate** is between **0** to **10**, inclusive,  **NULL** is permitted |
| **product\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **products**, Cascade Operations, **NULL** is **not** allowed |
| **customer\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **customers**, Cascade Operations, **NULL** is **not** allowed |

### distributors

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **id** | **Integer** from **0** to **2,147,483,647** | Primary Key, Unique table identification, Auto-increment |
| **name** | **String** up to **25** symbols | It has a **"UNIQUE"** constraint, **NULL** is **not** allowed |
| **address** | **String** up to **30** symbols | **NULL** is **not** allowed |
| **summary** | **String** up to **200** symbols | **NULL** is **not** allowed |
| **country\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **countries**, Cascade Operations, **NULL** is **not** allowed |

### ingredients

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **id** | **Integer** from **0** to **2,147,483,647** | Primary Key, Unique table identification, Auto-increment |
| **name** | **String** up to **30** symbols | **NULL** is **not** allowed |
| **description** | **String** up to **200** symbols | **NULL** is permitted |
| **country\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **countries**, Cascade Operations, **NULL** is **not** allowed |
| **distributor\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **distributors**, Cascade Operations, **NULL** is **not** allowed |

### products\_ingredients

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **product\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **products**, Cascade Operations, **NULL** is permitted |
| **ingredient\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **ingredients**, Cascade Operations, **NULL** is permitted |

## Database Design

Submit only your **CREATE** statements for all tables to the Judge.

# Section 2. Data Manipulation Language (DML) - (10 pts)

**Prior to beginning, it is necessary to import "dataset.sql". If the structure has been properly created, the data should be inserted successfully.**

In this scenario, you have multiple tasks that involve manipulating the database, such as modifying data and adding new entries.

## Insert

Oliver wants to celebrate the 5th anniversary of his bakery by sending presents to his valued customers. To facilitate this, he requests the creation of a new table named **"gift\_recipients"** with the following columns:

* **"id"** - an automatically generated primary key;
* **"name"**- concatenation of customers’ **"first\_name"** and **"last\_name"**;
* **"country\_id"** - the country **"id"** of the individuals eligible to receive a gift;
* **"gift\_sent"** - a Boolean flag with a default value of **"False"**

As part of the data insertion process, you should update the **"gift\_sent"** column to **"True"** for customers whose country **"id"** matches one of the following values: **7, 8, 14, 17, 26**.

To successfully accomplish the specified task, make sure to submit **all SQL queries** to the Judge.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **name** | **country\_id** | **gift\_sent** |
| 1 | Betty Wallace | 17 | true |
| 2 | Rachel Bishop | 3 | false |
| 3 | Joan Peters | 17 | true |
| 4 | Jean Pierce | 8 | true |
| … | … | … | … |
| 9 | Theresa Riley | 5 | false |
| 10 | Harry Jones | 15 | false |
| … | … | … | … |
| 22 | Larry Torres | 2 | false |
| 23 | Richard Carroll | 14 | true |
| … | … | … | … |
| 38 | Paula Gonzalez | 26 | true |
| 39 | Robin Daniels | 21 | false |
| 40 | Ashley Bryant | 4 | false |

## Update

In the task ahead, write an SQL query to **increase** **"price"** by **10%** for **products** with a **"rate"** above **8**.

### Example

Before update

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **description** | **recipe** | **price** |
| 1 | Octinoxate | Octocrylene | Oxybenzone libero non mattis pulvinar, nulla pede ullamcorper augue a suscipit … | 13.00 |
| 2 | Tobacco Cake | Sed ante. Vivamus tortor. Duis mattis … | Aliquam augue quam sollicitudin vitae consectetuer eget rutrum … | 14.39 |
| 3 | Musaka | Praesent blandit lacinia erat … | Duis at velit eu est congue elementum. | 15.05 |
| … | … | … | … | … |
| 6 | Salad | Nulla ac enim.Duis aliquam convallis nunc. | Curabitur in libero ut mas | 3.31 |
| 7 | Panetone | Nulla justo. Aliquam quis turpis eget … | Morbi non quam nec dui luctus rutrum. Nulla tellus. | 8.67 |
| … | … | … | … | … |
| 29 | Atropinum Sulphuricum | Berberis Vulgaris | Bryonia | 1.21 |
| 30 | ALCOHOL | Suspendisse potenti. | Proin at turpis a pede posuere nonummy. Integer non velit. | 24.37 |

After update

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **description** | **recipe** | **price** |
| 1 | Octinoxate | Octocrylene | Oxybenzone libero non mattis pulvinar, nulla pede ullamcorper augue a suscipit … | 14.30 |
| 2 | Tobacco Cake | Sed ante. Vivamus tortor. Duis mattis … | Aliquam augue quam sollicitudin vitae consectetuer eget rutrum … | 14.39 |
| 3 | Musaka | Praesent blandit lacinia erat … | Duis at velit eu est congue elementum. | 16.56 |
| … | … | … | … | … |
| 6 | Salad | Nulla ac enim.Duis aliquam convallis nunc. | Curabitur in libero ut mas | 3.64 |
| 7 | Panetone | Nulla justo. Aliquam quis turpis eget … | Morbi non quam nec dui luctus rutrum. Nulla tellus. | 8.67 |
| … | … | … | … | … |
| 29 | Atropinum Sulphuricum | Berberis Vulgaris | Bryonia | 1.21 |
| 30 | ALCOHOL | Suspendisse potenti. | Proin at turpis a pede posuere nonummy. Integer non velit. | 26.81 |

## Delete

In this task, you are responsible for cleaning up the database by removing a **distributor** and **all related records** associated with them. This task should be executed when a distributor's **"name"** starts with the letter **'L'**.

To successfully accomplish the specified task, make sure to submit **all SQL queries** to the Judge.

### Example

Before delete

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **address** | **summary** | **country\_id** |
| 1 | Loride | 1 Rieder Avenue | Zero defect delivering | 23 |
| 2 | Frova | 47450 Forster Place | Configurable clear-thinking delivery | 17 |
| … | … | … | … | … |
| 7 | Honey 4 Kids | 249 Dexter Plaza | Customer-focused zero defect | 15 |
| 8 | Levetiracetam | 15 Arkansas Center | Enhanced attitude-oriented business | 6 |
| 9 | Rabbitbush | 1 Northfield Court | Seamless interactive transport parallelism | 5 |
| … | … | … | … | … |
| 20 | Allopurinol | 20 Sachtjen Drive | Multi-lateral cohesive delivery | 2 |
| 22 | Lovastatin | 8 Katie Court | Exclusive interactive transport | 16 |
| 23 | Red Zone Collection | 60 High Crossing Terrace | Synergistic actuating project | 19 |
| … | … | … | … | … |
| 58 | Carvedilol | 6 Mcbride Trail | Multi-layered logistical capacity | 20 |
| 59 | Lisinopril | 9503 Autumn Leaf Way | Enterprise-wide dedicated | 14 |
| 60 | M3Modulator | 68233 Division Place | Networked clear-thinking transport | 8 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **description** | **country\_id** | **distributor\_id** |
| 1 | Allspice | It is a dark-brown pea-size berry. Comes from the evergreen pimento tree. | 5 | 4 |
| 2 | Basil | Member of the mint family. It has green leaves. | 12 | 3 |
| … | … | … | … | … |
| 19 | Mint | One of the most popular spice used. | 20 | 15 |
| 20 | Mustard | Comes in white yellow and brown seeds. | 19 | 22 |
| 21 | Nutmeg | Oval seeds from the nutmeg tree. Dark grey color. Mace is the spice … | 7 | 22 |
| 22 | Oregano | Member of the mint family related to marjoram and thyme | 15 | 8 |
| 23 | Paprika | Powdered dried red peppers | 2 | 4 |
| … | … | … | … | … |
| 43 | Fenugreek | One of the most popular spice used. | 32 | 15 |
| 44 | Turmeric | Comes in white yellow and brown seeds. | 32 | 22 |
| 45 | Garifalo | Garifalo (cloves) are an important ingredient in stifado and is also used in breads … | 31 | 22 |
| 46 | Kumino | Kumino (cumin) is used in soutzoukakia, the spicey meatballs served in tomato sauce. | 31 | 8 |
| 47 | Sousami | Sousami (sesame seeds) are used on breads and in halva and with honey to make … | 31 | 4 |
| … | … | … | … | … |
| 50 | Mahlab | Mahlab comes from the kernel of the sour cherry. On account of its intense aroma … | 31 | 4 |

|  |  |
| --- | --- |
| **product\_id** | **ingredient\_id** |
| 12 | 32 |
| 8 | 11 |
| 24 | 21 |
| 29 | 26 |
| … | … |
| 12 | 9 |
| 18 | 22 |
| 29 | 19 |
| … | … |
| 13 | 10 |
| 8 | 22 |
| 10 | 25 |
| … | … |
| 22 | 4 |

After delete

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **address** | **summary** | **country\_id** |
| 2 | Frova | 47450 Forster Place | Configurable clear-thinking delivery | 17 |
| … | … | … | … | … |
| 7 | Honey 4 Kids | 249 Dexter Plaza | Customer-focused zero defect | 15 |
| 9 | Rabbitbush | 1 Northfield Court | Seamless interactive transport parallelism | 5 |
| … | … | … | … | … |
| 20 | Allopurinol | 20 Sachtjen Drive | Multi-lateral cohesive delivery | 2 |
| 23 | Red Zone Collection | 60 High Crossing Terrace | Synergistic actuating project | 19 |
| … | … | … | … | … |
| 58 | Carvedilol | 6 Mcbride Trail | Multi-layered logistical capacity | 20 |
| 60 | M3Modulator | 68233 Division Place | Networked clear-thinking transport | 8 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **description** | **country\_id** | **distributor\_id** |
| 1 | Allspice | It is a dark-brown pea-size berry. Comes from the evergreen pimento tree. | 5 | 4 |
| 2 | Basil | Member of the mint family. It has green leaves. | 12 | 3 |
| … | … | … | … | … |
| 19 | Mint | One of the most popular spice used. | 20 | 15 |
| 23 | Paprika | Powdered dried red peppers | 2 | 4 |
| … | … | … | … | … |
| 43 | Fenugreek | One of the most popular spice used. | 32 | 15 |
| 47 | Sousami | Sousami (sesame seeds) are used on breads and in halva and with honey to make … | 31 | 4 |
| … | … | … | … | … |
| 50 | Mahlab | Mahlab comes from the kernel of the sour cherry. On account of its intense aroma … | 31 | 4 |

|  |  |
| --- | --- |
| **product\_id** | **ingredient\_id** |
| 12 | 32 |
| 8 | 11 |
| 29 | 26 |
| … | … |
| 12 | 9 |
| 29 | 19 |
| … | … |
| 13 | 10 |
| 10 | 25 |
| … | … |
| 22 | 4 |

# Section 3. Querying - (40 pts)

**Now we will perform some data extraction tasks. Please note that the example results provided in this section are based on a fresh database. It is highly recommended to clear the database that was manipulated in the previous problems from the DML section and insert the given dataset again to ensure consistency with the examples in this section.**

## Products

Retrieve all products with **prices in the range of** **10** **to** **20**, and order the results by **"price"** in **descending order**. The required columns are **"name"**, **"recipe"**, and **"price"**.

### Example

|  |  |  |
| --- | --- | --- |
| **name** | **recipe** | **price** |
| Fish burger | Nulla facilisi. Cras non velit nec nisi vulputate nonu | 16.33 |
| Rock | Duis aliquam convallis nunc. Proin at turpis a pede posuere nonummy. Integer non velit. | 15.82 |
| Musaka | Duis at velit eu est congue elementum. | 15.05 |
| … | … | … |
| Fire in kitchen | diam id ornare imperdiet | 12.10 |
| Bread | Praesent blandit. Nam nulla. | 11.10 |

## Negative Feedback

Select feedback records alongside the customers who provided them. Filter the feedbacks with a **rating below** **5.0** and **customers who are females** (**"gender"** is **'F'**) with an **age greater than** **30**. Finally, order the results by **"product\_id"** in **descending order**. The required columns are **"product\_id"**, **"rate"**, **"description"**, **"customer\_id"**, **"age"**, and **"gender"**.

### Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **product\_id** | **rate** | **description** | **customer\_id** | **age** | **gender** |
| 27 | 4.16 | Meh.. | 20 | 57 | F |
| 21 | 3.36 | Way too salty | 17 | 48 | F |
| 6 | 3.60 | First food was not ok. Second I do not like sombreros. Third the music was bad. | 14 | 37 | F |
| 5 | 0.86 | You should kill yourselves. | 26 | 47 | F |

## High Average Price and Multiple Feedbacks

Create an SQL query to fetch products with a **price exceeding $15** and having **more than one feedback**. Calculate the **average price for each group of products, rounding that average to two decimal places**, and assign the result the label **"average\_price"**. Arrange the results in **ascending order** based on **"total\_feedbacks"**. In case of products having the same count of feedback, order the results by **"average\_price"** in **descending order**. Include the columns **"product\_name"**, **"average\_price"**, and **"total\_feedbacks"**.

### Example

|  |  |  |
| --- | --- | --- |
| **product\_name** | **average\_price** | **total\_feedbacks** |
| Bread | 27.39 | 2 |
| Pizza(small) | 27.27 | 2 |
| ALCOHOL | 24.37 | 2 |
| iloperidone | 21.86 | 2 |
| Fried Eggs bread | 26.61 | 3 |

## Specific Ingredients

Write an SQL query to retrieve a **list of ingredients** and the **corresponding products** that contain them, and the **names of the distributors** for these ingredients. The query should only include products with **'mustard'**(it should be **case-insensitive**) in their **ingredients** **"name"** and **distributors** from a specific country (**"country\_id" = 16**). Sort the results by **"product\_name"** in ascending order.

### Example

|  |  |  |
| --- | --- | --- |
| **ingredient\_name** | **product\_name** | **distributor\_name** |
| Mustard | Musaka | Lovastatin |
| Mustard | Panetone | Lovastatin |
| Mustard | Tobacco Cake | Lovastatin |

## Middle Range Distributors

Select all **distributors** that distribute ingredients used in the **manufacturing process of products with an average rating between** **5** and **8** (**inclusive**). Order them first by **"distributor\_name"**. If there are one or more distributors with the same name, then order them by **"ingredient\_name"**. If, in the rare case, there are still duplicates, order by **"product\_name"**, all in **ascending order**.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **distributor\_name** | **ingredient\_name** | **product\_name** | **average\_rate** |
| Alprazolam | Cinnamon | Nicotine Pleasure | 5.2600000000000000 |
| Arinase | Peppercorn | Panetone | 5.4000000000000000 |
| Capex | Clove | Root | 5.0466666666666667 |
| … | … | … | … |
| Rabbitbush | Dill Weed | Banitsa | 5.5400000000000000 |
| Rabbitbush | Marjoram | Panetone | 5.4000000000000000 |
| Rabbitbush | Marjoram | Root | 5.0466666666666667 |

# Section 4. Programmability - (20 pts)

## Customer Feedback

You have been tasked with creating a user-defined function named **fn\_feedbacks\_for\_product()** that retrieves **customer feedback for a specific product**. This function is designed to accept the product's name as a parameter (**"product\_name"** of type **VARCHAR(25)**) and should return a table with the following columns:

* **"customer\_id" (INT)** - representing the unique identifier of the customer who provided the feedback;
* **"customer\_name" (VARCHAR(75))** - denoting the first name of the customer who supplied the feedback;
* **"feedback\_description" (VARCHAR(255))** - describing the content or text of the feedback;
* **"feedback\_rate" (NUMERIC(4, 2))** - indicating the numerical rating provided in the feedback;

The results should be ordered by customer **"id"** in **ascending order**.

For this task, please only submit your **user-defined function** in the Judge system.

### Examples

|  |
| --- |
| **Query** |
| **SELECT \* FROM fn\_feedbacks\_for\_product('Banitsa');** |

|  |  |  |  |
| --- | --- | --- | --- |
| Output | | | |
| **customer\_id** | **customer\_name** | **feedback\_description** | **feedback\_rate** |
| **18** | **Edward** | **Well- it is overpriced** | **5.54** |

|  |
| --- |
| **Query** |
| **SELECT \* FROM fn\_feedbacks\_for\_product('ALCOHOL');** |

|  |  |  |  |
| --- | --- | --- | --- |
| Output | | | |
| **customer\_id** | **customer\_name** | **feedback\_description** | **feedback\_rate** |
| **15** | **Randy** | **Let's find some flavours** | **8.94** |
| **23** | **Richard** | **I did not like the product** | **2.04** |

|  |
| --- |
| **Query** |
| **SELECT \* FROM fn\_feedbacks\_for\_product('Bread');** |

|  |  |  |  |
| --- | --- | --- | --- |
| Output | | | |
| **customer\_id** | **customer\_name** | **feedback\_description** | **feedback\_rate** |
| **11** | **Jerry** | **My greetings - it could be more enjoyable if ingredients were fresh** | **6.17** |
| **14** | **Lisa** | **We had an amazing time.** | **7.77** |
| **17** | **Rachel** | **Way too salty** | **3.36** |
| **22** | **Larry** | **Total garbage food.** | **1.11** |
| **30** | **Amanda** | **Best ingredients - worst cooking** | **7.23** |

## Customer’s Country

Your final task is to create a stored procedure called **sp\_customer\_country\_name()** designed to retrieve the country name associated with a **given customer's full name**. This procedure will take the customer's full name as input (**"customer\_full\_name"** of type **VARCHAR(50)**) and extract the name of the country from which they originate (**"country\_name"** of type **VARCHAR(50)**).

For this task, please only submit your **stored procedure** in the Judge system.

### Example

|  |  |
| --- | --- |
| **Query** | **Output** |
| **CALL sp\_customer\_country\_name('Betty Wallace', '')** | **South Korea** |
| **CALL sp\_customer\_country\_name('Rachel Bishop', '')** | **Japan** |
| **CALL sp\_customer\_country\_name('Jerry Andrews', '')** | **Bangladesh** |