# PostgreSQL Regular Exam

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

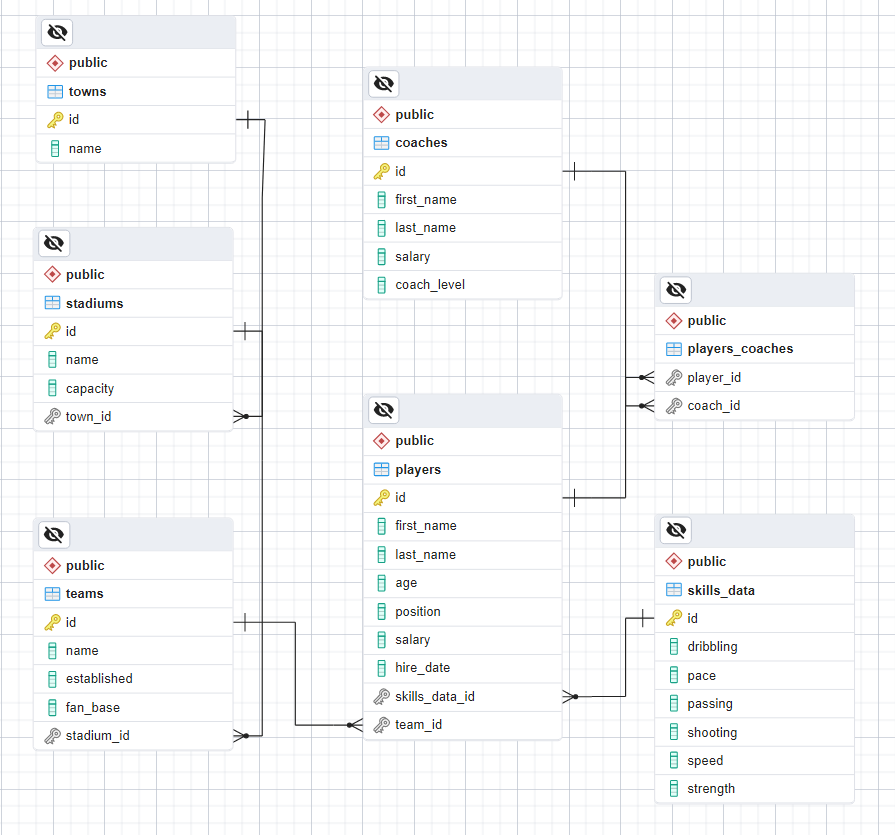
**Submit your solutions** in the SoftUni [Judge Contest](https://judge.softuni.org/Contests/4401/PostgreSQL-Exam-14-October-2023).

# Soccer Talent

# *You've been selected to help soccer coaches pick the best players for their teams. Using your database skills, you're creating Soccer Talent a custom database with expert scout data. Once done, you can easily answer coaches' queries and streamline player selection. Mastering the database structure comes first, then filling it with soccer talent data. Welcome to a world where your database skills shape the future stars of the beautiful game.*

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

The E/R Diagram for **Soccer Talent** has been supplied to you. This diagram illustrates the connections among different entities within **Soccer Talent**, offering a visual depiction of the database structure.



Create a PostgreSQL database named **"soccer\_talent\_db"** that comprises seven tables:

* **"towns"** - contains information about the players' towns;
* **"stadiums"** - holds data about the stadiums;
* **"teams"** - contains information about the teams;
* **"coaches"** - stores details about the coaches. A coach can train multiple players;
* **"skills\_data"** - holds information about the current player's skills;
* **"players"** - contains details about the players;
* **"players\_coaches"** - serves as a many-to-many mapping table between players and coaches;

**NOTE: Please ensure that you use the exact data types specified in the model tables when working with dates. For instance, if a column is of type "DATE," make sure to use the "DATE" type. Similarly, if a column is of type "TIMESTAMP" use the "TIMESTAMP" data type. If you use an incorrect type, the Judge system will not accept your submission as correct.**

Your first task is to set up the database tables following the provided models. Follow the specifications closely, maintaining the same column order as shown below.

### towns

|  |  |  |
| --- | --- | --- |
| **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 **45** symbols | **NULL** is **not** allowed |

### stadiums

|  |  |  |
| --- | --- | --- |
| **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 **45** symbols | **NULL** is **not** allowed |
| **capacity** | **Integer** from **0** to **2,147,483,647** | The **capacity** is a **positive number greater than 0**  **NULL** is **not** allowed |
| **town\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **towns**, Cascade Operations, **NULL** is **not** allowed |

### teams

|  |  |  |
| --- | --- | --- |
| **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 **45** symbols | **NULL** is **not** allowed |
| **established** | **DATE** | **NULL** is **not** allowed |
| **fan\_base** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is **not** allowed |
| **stadium\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **stadiums**, Cascade Operations, **NULL** is **not** allowed |

### coaches

|  |  |  |
| --- | --- | --- |
| **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 **10** symbols | **NULL** is **not** allowed |
| **last\_name** | **String** up to **20** symbols | **NULL** is **not** allowed |
| **salary** | **Numeric** number with a precision of **10** digits, including **2** digits after the decimal point | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is **not** allowed |
| **coach\_level** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is **not** allowed |

### skills\_data

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **id** | **Integer** from **0** to **2,147,483,647** | Primary Key, Unique table identification, Auto-increment |
| **dribbling** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is permitted |
| **pace** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is permitted |
| **passing** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is permitted |
| **shooting** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is permitted |
| **speed** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is permitted |
| **strength** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is permitted |

### players

|  |  |  |
| --- | --- | --- |
| **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 **10** symbols | **NULL** is **not** allowed |
| **last\_name** | **String** up to **20** symbols | **NULL** is **not** allowed |
| **age** | **Integer** from **0** to **2,147,483,647** | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is **not** allowed |
| **position** | **String** limited to **1** character | **NULL** is **not** allowed |
| **salary** | **Numeric** number with a precision of **10** digits, including **2** digits after the decimal point | The **DEFAULT** value is **0**, the column must always have a value **greater than or equal to zero**  **NULL** is **not** allowed |
| **hire\_date** | **TIMESTAMP** indicates when the player's contract starts | **NULL** is permitted |
| **skills\_data\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **skills\_data**, Cascade Operations, **NULL** is **not** allowed |
| **team\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **teams**, Cascade Operations, **NULL** is permitted |

### players\_coaches

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **player\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **players**, Cascade Operations, **NULL** is permitted |
| **coach\_id** | **Integer** from **0** to **2,147,483,647** | Relationship with table **coaches**, 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)

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

This section requires performing various data manipulations:

## Insert

Players hired **before** **'2013-12-13 07:18:46'** are eligible to apply for coaching positions. Your task is to update the **"coaches"** table by extracting information from the **"players"** table. Insert data into the **"coaches"** table with the following specifications:

* set the **"first\_name"** to the **player's first name**.
* set the **"last\_name"** to the **player's last name**.
* set the **"salary"** to **double** the **player's salary**.
* set the **"coach\_level"** to the **character count** of the player's **first name**.

### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **salary** | **coach\_level** |
| 1 | Anollie | Phelip | 578112.19 | 4 |
| 2 | Aster | Krolak | 876807.09 | 1 |
| 3 | Aesra | Simoneton | 336677.23 | 4 |
| … | … | … | … | … |
| 9 | Rudie | Gorgl | 460354.93 | 2 |
| 10 | Lewes | Dymocke | 898257.96 | 7 |
| 11 | Harlie | Sandells | 1580855.78 | 6 |
| 12 | Thor | Serrels | 911203.34 | 4 |
| … | … | … | … | … |
| 43 | Curtis | Lawrenceson | 74142.90 | 6 |
| 44 | Kate | Taylder | 991817.08 | 4 |
| 45 | Jorrie | Lumsden | 1016070.88 | 6 |

## Update

For the upcoming task**, update the salaries of coaches** whose **"first\_name"** starts with **'C'** and who **train one or more players**. Increase the **"salary"** by **multiplying** their current **"salary"** by their **"coach\_level"**.

### Example

Before update

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **salary** | **coach\_level** |
| 1 | Anollie | Phelip | 578112.19 | 4 |
| … | … | … | … | … |
| 7 | Mickey | Dabernott | 680019.08 | 7 |
| 8 | Chilton | Cookley | 56839.58 | 4 |
| 9 | Rudie | Gorgl | 460354.93 | 2 |
| … | … | … | … | … |
| 15 | Jorrie | Lumsden | 1016070.88 | 6 |

After update

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **salary** | **coach\_level** |
| 1 | Anollie | Phelip | 578112.19 | 4 |
| … | … | … | … | … |
| 7 | Mickey | Dabernott | 680019.08 | 7 |
| 8 | Chilton | Cookley | 227358.32 | 4 |
| 9 | Rudie | Gorgl | 460354.93 | 2 |
| … | … | … | … | … |
| 15 | Jorrie | Lumsden | 1016070.88 | 6 |

## Delete

As you may recall, at the beginning of our project, we promoted several football players to coaching roles. To ensure the accurate update of our database, your assignment is to **remove all instances of these promoted players** from the "**players**" table. More specifically, eliminate players **hired before** '2013-12-13 07:18:46'. Additionally, delete all associated records for these players from the "**players\_coaches**" table.

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

### Example

Before delete

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **age** | **position** | **salary** | **hire\_date** | **skills\_data\_id** | **team\_id** |
| 1 | Alden | Wrettum | 21 | A | 32283.83 | [null] | 87 | [null] |
| 2 | Dayna | Halesworth | 23 | M | 897853.69 | 2020-01-15 07:26:33 | 88 | 77 |
| … | … | … | … | … | … | … | … | … |
| 11 | Curry | Brando | 17 | M | 511422.90 | 2019-08-16 16:02:01 | 34 | 36 |
| 12 | Thor | Serrels | 24 | D | 455601.67 | 2013-03-19 15:23:23 | 47 | 36 |
| 13 | Berkie | Maryin | 45 | A | 698230.79 | 2016-12-27 13:45:05 | 65 | 77 |
| … | … | … | … | … | … | … | … | … |
| 16 | Eldin | Gravet | 35 | D | 821422.57 | 2019-03-23 11:24:11 | 69 | 53 |
| 17 | Glory | Crosetti | 28 | A | 394462.27 | 2010-08-05 19:01:14 | 13 | 1 |
| 18 | Doretta | Rignold | 42 | M | 665969.43 | 2015-09-18 11:43:44 | 40 | 66 |
| … | … | … | … | … | … | … | … | … |
| 19 | Gwendolen | Semple | 17 | D | 407582.09 | [null] | 94 | [null] |
| 20 | Launce | Perchard | 44 | A | 899242.30 | 2013-09-13 14:16:39 | 70 | 13 |
| 21 | Vasili | Grigorescu | 45 | M | 46428.66 | 2013-12-13 07:18:46 | 91 | 76 |
| … | … | … | … | … | … | … | … | … |
| 99 | Miranda | Frichley | 45 | A | 307130.04 | 2019-06-20 16:31:41 | 53 | 22 |
| 100 | Jorrie | Lumsden | 50 | M | 508035.44 | 2010-09-19 11:52:56 | 4 | 54 |

|  |  |
| --- | --- |
| **player\_id** | **coach\_id** |
| 1 | 1 |
| 54 | 2 |
| 17 | 5 |
| 33 | 4 |
| 24 | 8 |
| 66 | 10 |
| 90 | 5 |
| 4 | 6 |
| 39 | 10 |
| 71 | 8 |

After delete

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **age** | **position** | **salary** | **hire\_date** | **skills\_data\_id** | **team\_id** |
| 1 | Alden | Wrettum | 21 | A | 32283.83 | [null] | 87 | [null] |
| 2 | Dayna | Halesworth | 23 | M | 897853.69 | 2020-01-15 07:26:33 | 88 | 77 |
| … | … | … | … | … | … | … | … | … |
| 11 | Curry | Brando | 17 | M | 511422.90 | 2019-08-16 16:02:01 | 34 | 36 |
| 13 | Berkie | Maryin | 45 | A | 698230.79 | 2016-12-27 13:45:05 | 65 | 77 |
| … | … | … | … | … | … | … | … | … |
| 16 | Eldin | Gravet | 35 | D | 821422.57 | 2019-03-23 11:24:11 | 69 | 53 |
| 18 | Doretta | Rignold | 42 | M | 665969.43 | 2015-09-18 11:43:44 | 40 | 66 |
| … | … | … | … | … | … | … | … | … |
| 19 | Gwendolen | Semple | 17 | D | 407582.09 | [null] | 94 | [null] |
| 21 | Vasili | Grigorescu | 45 | M | 46428.66 | 2013-12-13 07:18:46 | 91 | 76 |
| … | … | … | … | … | … | … | … | … |
| 99 | Miranda | Frichley | 45 | A | 307130.04 | 2019-06-20 16:31:41 | 53 | 22 |

|  |  |
| --- | --- |
| **player\_id** | **coach\_id** |
| 1 | 1 |
| 33 | 4 |
| 24 | 8 |
| 90 | 5 |
| 4 | 6 |
| 39 | 10 |

# 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.**

## Players

Extract information about all the **"players"** with their **"full\_name"** (concatenation of **"first\_name"** and **"last\_name"**), **"age"**, and **"hire\_date"**. Select only the players whose **"first\_name"** starts with **'M%'**. Sort the list by **"age"** in **descending order**. If there is more than one player with the same age, the results should be further sorted by their **"full\_name"** in **ascending order**.

### Example

|  |  |  |
| --- | --- | --- |
| **full\_name** | **age** | **hire\_date** |
| Meredith Duffett | 46 | 2015-10-30 16:20:42 |
| Miranda Frichley | 45 | 2019-06-20 16:31:41 |
| Marni McDonald | 42 | 2012-04-08 23:10:52 |
| … | … | … |
| Malissa Paylie | 27 | 2012-05-01 07:52:22 |
| Marquita Sigert | 27 | 2017-02-19 23:07:14 |
| … | … | … |
| Melodee McVey | 22 | 2018-02-11 08:41:37 |
| Myer Daenen | 16 | 2017-06-19 08:25:13 |

## Offensive Players without Team

A coach has requested assistance in identifying players in offensive **"position"** (**'A'**) who are currently **not part of any team**. The coach aims to create a team consisting of players with strong offensive abilities, specifically those whose **combined score** in **"pace"** and **"shooting"** is above **130**. Required columns:

* **"id"** - players’ id;
* **"full\_name"** - concatenation of players’ **"first\_name"** and **"last\_name"**;
* **"age"**;
* **"position"**;
* **"salary"**;
* **"pace"**;
* **"shooting"**;

### Example

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **full\_name** | **age** | **position** | **salary** | **pace** | **shooting** |
| 97 | Gianni Morrow | 16 | A | 762456.74 | 82 | 68 |

## Teams with Player Count and Fan Base

Write an SQL query to retrieve information about **"teams"**, focusing on the **"player\_count"** for each team and selecting only those with a **"fan\_base"** **greater than** **30000**. Arrange the results by **"player\_count"** in **descending order**. If teams have the same **"player\_count"**, further order them by **"fan\_base"** in **descending order**. The output should include columns for **"team\_id"**, **"team\_name"**, **"player\_count"**, and **"fan\_base"**.

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **team\_id** | **team\_name** | **player\_count** | **fan\_base** |
| 51 | Ailane | 10 | 32000 |
| 78 | Skipstorm | 2 | 32000 |
| 66 | Yombu | 1 | 32000 |
| 29 | Voolia | 1 | 31000 |
| … | … | … | … |
| 61 | Tagopia | 0 | 31000 |
| 52 | Zoombox | 0 | 31000 |

## Coaches, Players Skills and Teams Overview

Retrieve information about coaches, their players, and their respective skills, along with the team each player belongs to. The goal is to obtain details such as the **"coach\_full\_name"**, the **"player\_full\_name"** (formed by combining **"first\_name"** and **"last\_name"**), and the **"name"** of the **team** each player is a part of. Additionally, select the **player's skills** (**"passing"**, **"shooting"**, and **"speed"**). Arrange the results by the **coach's full name** in **ascending order**. If a coach has multiple players, also sort the results by the **player's full name** in **descending order**.

### Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **coach\_full\_name** | **player\_full\_name** | **team\_name** | **passing** | **shooting** | **speed** |
| Acad Clyne | Ada Doumic | Edgetag | 84 | 44 | 88 |
| Arcos Chettle | Melodee McVey | Feedmix | 50 | 73 | 87 |
| Arcos Chettle | Glory Crosetti | Skyble | 61 | 73 | 41 |
| … | … |  | … | … | … |
| Lewes Dymocke | Camey Michurin | Pixonyx | 94 | 47 | 4 |
| Reynard Gravenor | Bibbye O'Lunney | Ntags | 39 | 72 | 49 |

# Section 4. Programmability - (20 pts)

## Stadium Teams Information

You have been assigned the creation of a user-defined function called **fn\_stadium\_team\_name()**. This function is designed to take a **stadium's name** as a parameter (**"stadium\_name"** of type **VARCHAR(30)**) and returns details about the **names of teams** playing home matches at that particular stadium. In cases where **multiple teams** share the same stadium, the function ensures they are **ordered alphabetically**.

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

### Examples

|  |
| --- |
| **Query** |
| **SELECT fn\_stadium\_team\_name('BlogXS')** |

|  |
| --- |
| Output |
| **fn\_stadium\_team\_name** |
| **Fiveclub** |

|  |
| --- |
| **Query** |
| **SELECT fn\_stadium\_team\_name('Quaxo')** |

|  |
| --- |
| Output |
| **fn\_stadium\_team\_name** |
| **Divavu** |
| **Photobug** |

|  |
| --- |
| **Query** |
| **SELECT fn\_stadium\_team\_name('Jaxworks')** |

|  |
| --- |
| Output |
| **fn\_stadium\_team\_name** |
| **Ailane** |
| **Feedmix** |
| **Jabbercube** |
| **Skipstorm** |

## Player Team Finder

Your last assignment is to create a stored procedure named **sp\_players\_team\_name()**. This procedure is designed to accept the **full name of a player** as input (**"player\_name"** of type **VARCHAR(50)**) and extract the **name of the team** to which the player currently belongs as output (**"team\_name"** of type **VARCHAR(45)**). In cases where the player is not **associated with any team**, the output should be **"The player currently has no team"**.

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

### Example

|  |  |
| --- | --- |
| **Query** | **Output** |
| **CALL sp\_players\_team\_name('Thor Serrels', '')** | **Ntags** |
| **CALL sp\_players\_team\_name('Walther Olenchenko', '')** | **The player currently has no team** |
| **CALL sp\_players\_team\_name('Isaak Duncombe', '')** | **Thoughtstorm** |