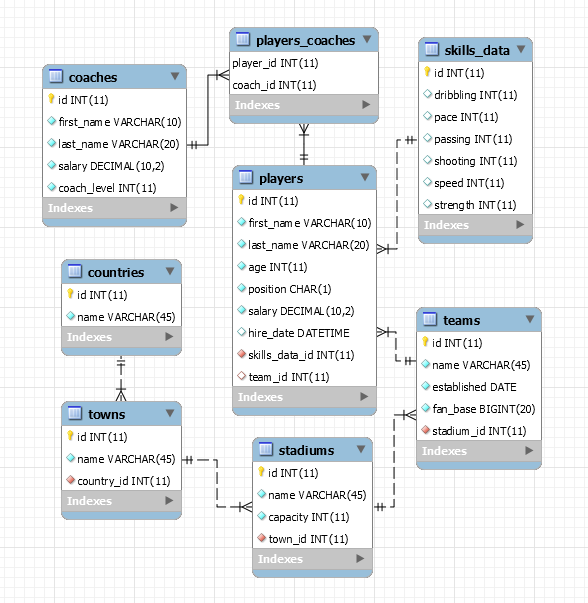
# MySQL Exam Football Scout Database

[Database Basics MySQL Exam - 9 Feb 2020 Link in Jud](https://judge.softuni.org/Contests/2027/Database-Basics-MySQL-Exam-9-Feb-2020)

## In a parallel reality, you have been selected to help the most famous football coaches to select the best players for their teams. Thanks to your knowledge of databases, you have been selected to create the structure of a brand new database, tailored to the requirements of your employers, and to fill it in with a huge amount of data collected for you by the most experienced football scouts. Once the base is ready, you will be able to respond without any problems to any information request from the coaches on the basis of certain criteria. As with other databases, it is most important first to become familiar with the structure you need to build, and then fill it with given data.

## Section 0: Database Overview

You have been given an Entity / Relationship Diagram of the Football Scout Database:



The **Football Scout Database (FSD)** needs to hold information about **players**, **skill**, **coaches**, **teams, stadiums, towns**, **countries**.

Your task is to create a database called fsd (**Football Scout Database**). Then you will have to create several **tables**.

* players – contains information about the **players**.
  + Each player has a skills data, team and coach.
* coaches – contains information about the **coaches**.
  + One coach can train many players
* players\_coaches – a **many** to **many** **mapping** table between the **players** and the **coaches**.
  + Have composite primary key from player\_id and coach\_id
* skill data – contains information about the **current player skills**.
* teams – contains information about the **teams**.
  + Each team has a stadium.
* stadiums – contains information about the stadiums.
  + Each stadium has a city.
* towns - contains information about the **towns**.
  + Each town has a country
* **countries** – contains information about current country.

## Section 1: Data Definition Language (DDL) – 40 pts

Make sure you implement the whole database correctly on your local machine, so that you could work with it.

The instructions you’ll be given will be the minimal required for you to implement the database.

### Table Design

You have been tasked to create the tables in the database by the following models:

players

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer,** from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| first\_name | A **string** containing a maximum of **10 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**.** |
| last\_name | A **string** containing a maximum of **20 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**.** |
| age | **Integer,** from **1** to **2,147,483,647.** | **DEFAULT value is 0.**  **NULL** is **NOT** permitted**.** |
| position | **Exactly 1 character – A** or **M** or **D** | **NULL** is **NOT** permitted**.** |
| salary | **DECIMAL**, up to **10 digits**, **2** of which after the **decimal point**. | **DEFAULT value is 0.**  **NULL** is **NOT** permitted**.** |
| hire\_date | The exact **time** and **date** of conclusion of the contract | **NULL is permitted.** |
| skills\_data\_id | **Integer,** from **1** to **2,147,483,647.** | Relationship with table skill\_datas.  **NULL** is **NOT** permitted. |
| team\_id | **Integer,** from **1** to **2,147,483,647.** | Relationship with table teams.  **NULL** **is** permitted. |

players\_coaches

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| player\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table players. |
| coach\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table coaches. |

coaches

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer,** from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| first\_name | A **string** containing a maximum of **10 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**.** |
| last\_name | A **string** containing a maximum of **20 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**.** |
| salary | **DECIMAL**, up to **10 digits**, **2** of which after the **decimal point**. | **DEFAULT value is 0.**  **NULL** is **NOT** permitted**.** |
| coach\_level | **Integer**, from **1** to **2,147,483,647.** | **DEFAULT value is 0.**  **NULL** is **NOT** permitted**.** |

skills\_data

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**, from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| dribbling | **Integer**, from **1** to **2,147,483,647.** | **DEFAULT value is 0** |
| pace | **Integer**, from **1** to **2,147,483,647.** | **DEFAULT value is 0** |
| passing | **Integer**, from **1** to **2,147,483,647.** | **DEFAULT value is 0** |
| shooting | **Integer**, from **1** to **2,147,483,647.** | **DEFAULT value is 0** |
| speed | **Integer**, from **1** to **2,147,483,647.** | **DEFAULT value is 0** |
| strength | **Integer**, from **1** to **2,147,483,647.** | **DEFAULT value is 0** |

teams

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**, from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| name | A **string** containing a maximum of **45 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted. |
| established | The **year** and **date** the club was founded | **NULL** is **NOT** permitted. |
| fan\_base | **Very** big **Integer**, from **1** to **264-1.** | **NOT** is **NULL** permitted,  **DEFAULT value** is 0 |
| stadium\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table stadiums.  **NULL** is **NOT** permitted**.** |

stadiums

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**, from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| name | A **string** containing a maximum of **45 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted. |
| capacity | **Integer**, from **1** to **2,147,483,647.** | **NULL** is **NOT** permitted**.** |
| town\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table towns.  **NULL** is **NOT** permitted**.** |

towns

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**, from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| name | A **string** containing a maximum of **45 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted. |
| country\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table countries.  **NULL** is **NOT** permitted**.** |

countries

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**, from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| name | A **string** containing a maximum of **45 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted. |

Submit your solutions in Judge on the first task. Submit **all** SQL table creation statements.

You will also be given a data.sql file. It will contain a **dataset** with random data which you will need to **store** in your **local database**. This data will be given to you so you don’t have to imagine it and lose precious time in the process. The data is in the form of **INSERT** statement queries.

## Section 2: Data Manipulation Language (DML) – 30 pts

Here we need to do several manipulations in the database, like changing data, adding data etc.

### Insert

You will have to **insert** records of data into the **coaches** table, based on the **players** table.

For players with age over 45(**inclusive**), **insert data** in the coaches table with the **following values**:

* first\_name –set it to **first name** of the **player**
* last\_name – set it to **last name** of the **player**.
* salary – set it to **double** as **player’s** salary.
* coach\_level – set it to be equals to count of the characters in player’s first\_name.

### Update

Update all **coaches,** who train **one or more** players **and** their **first\_name** **starts** with ‘**A**’. **Increase** their **level** with **1**.

### Delete

As you remember at the beginning of our work, we **promoted** several football players to coaches. Now you need to **remove all of them** from the table of players in order for our database to be updated accordingly.

**Delete** all players from table players**,** which are already added in table coaches.

## Section 3: Querying – 50 pts

And now we need to do some data extraction. **Note** that the **example results** from **this section** use a **fresh database**. It is **highly recommended** that you **clear** the **database** that has been **manipulated** by the **previous problems** from the **DML** **section** and **insert again** the **dataset** you’ve been given, to ensure **maximum consistency** with the **examples** given in this section.

### Players

Extract from the Football Scout Database (fsd) database, info about all of the **players**.

**Order** the results by **players - salary descending.**

#### Required Columns

* first\_name
* age
* salary

#### Example

|  |  |  |
| --- | --- | --- |
| **first\_name** | **age** | **salary** |
| Renault | 24 | 984113.71 |
| Akim | 31 | 982188.88 |
| Mollie | 24 | 966079.07 |
| … | … | … |

### Young offense players without contract

One of the coaches wants to know more about all the **young players** (under age of **23**) who can strengthen his team in the offensive (played on **position ‘A’**). As he is not paying a transfer amount, he is looking only for those who have **not signed** a contract so far (haven’t **hire\_date**) and have **strength of** more than **50**. **Order** the results **ascending** by salary, then by **age**.

#### Required Columns

* id (player)
* full\_name
* age
* position
* hire\_date

#### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **full\_name** | **age** | **position** | **hire\_date** |
| 40 | Carlen Hadny | 18 | A | NULL |
| 23 | Kalvin Bewley | 19 | A | NULL |
| .. | .. | .. | .. | .. |

### Detail info for all teams

Extract from the database all of the **teams** and the **count** of the **players** that they have.

**Order** the results **descending** **by count of players**, then by fan\_base **descending**.

#### Required Columns

* team\_name
* established
* fan\_base
* count\_of\_players

#### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **team\_name** | established | fan\_base | players\_count |
| Ailane | 1963-08-20 | 6711237100133852778 | 10 |
| Ntags | 1981-06-05 | 3508984270641351110 | 9 |
| Skyble | 1953-11-14 | 5381600486852672412 | 8 |
| … | … | … | … |
| Miboo | 1971-08-02 | 30087062078800256 | 0 |

### The fastest player by towns

Extract from the database, the **fastest player (having max speed)**, in terms of **towns** where their team played.

**Order** players by **speed descending**, then by **town name**.

**Skip** players that played in team **‘Devify’**

#### Required Columns

* max\_speed
* town\_name

#### Example

|  |  |
| --- | --- |
| **max\_speed** | **town\_name** |
| 97 | Smolensk |
| 92 | Bromma |
| 92 | Lühua |
| … | … |
| NULL | Zavolzh’ye |

### Total salaries and players by country

And like everything else in this world, everything is ultimately about finances. Now you need to extract detailed information on the amount of **all salaries** given to football players by the **criteria of the country in which they played**.

**If there are no players in a country**, display **NULL**. Order the results by **total count of players** in **descending** order, then by **country name** alphabetically.

#### Required Columns

* name (country)
* total\_sum\_of\_salaries
* total\_count\_of\_players

#### Example

|  |  |  |
| --- | --- | --- |
| **name** | total\_count\_of\_players | total\_sum\_of\_salaries |
| Sweden | 28 | 14968947.79 |
| Brazil | 18 | 8352732.65 |
| China | 13 | 7042890.51 |
| Russia | 7 | 2230759.71 |
| … | … | … |
| Thailand | 0 | NULL |
| United States | 0 | NULL |

## Section 4: Programmability – 30 pts

The time has come for you to prove that you can be a little more dynamic on the database. So, you will have to write several procedures.

### Find all players that play on stadium

Create a **user defined function** with the name **udf\_stadium\_players\_count (stadium\_name VARCHAR(30))** that receives a **stadium’s name** and returns the number of players that play home matches there.

#### Example

|  |
| --- |
| **Query** |
| SELECT udf\_stadium\_players\_count ('Jaxworks') as `count`; |
| count |
| **14** |

|  |
| --- |
| **Query** |
| SELECT udf\_stadium\_players\_count ('Linklinks') as `count`; |
| count |
| **0** |

### Find good playmaker by teams

Create a stored procedure udp\_find\_playmaker which accepts the following parameters:

* min\_dribble\_points
* team\_name (with max length 45)

And extracts data about the **players** with the given **skill stats (more than min\_dribble\_points)**, played for **given team** (team\_name) and have **more than average** **speed** for **all** **players**. **Order** players by **speed** descending. Select **only the best one**.

Show all needed info for this player: full\_name, age, salary, dribbling, speed, team name.

CALL udp\_find\_playmaker (20, ‘**Skyble**’);

#### Result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **full\_name** | **age** | **salary** | **dribbling** | **speed** | **team\_name** |
| Royal Deakes | 19 | 49162.77 | 33 | 92 | Skyble |