# Database Basics MS SQL Regular Exam – 09 Feb 2025

Exam problems for the ["Database Basics" course @ SoftUni](https://softuni.bg/trainings/4841/ms-sql-january-2025).  
Submit your solutions in the SoftUni Judge system at [Judge](https://judge.softuni.org/Contests/5359/MS-SQL-Regular-Exam-9-February-2025).

# Euro Leagues

# Section 1. DDL (30 pts)

You have been given the E/R Diagram of the **EuroLeagues** database.



Create a database called **EuroLeagues**. You need to create **6 tables**:

* **Leagues** – Contains information about football leagues. Each league represents a national competition (e.g., La Liga, Serie A, Premier League);
* **Teams** – Contains details about football teams participating in the leagues. Includes the team name, city, and the league they belong to;
* **Players** – Contains information about football players, including their names, and positions;
* **Matches**– Stores information about matches between teams, including the home team, away team, match date, and goals scored by each team;
* **PlayersTeams** – Мanages the relationship between **players** and **teams**, indicating which player is playing for specific team;
* **PlayerStats** – Tracks individual player performance such as goals scored and assists made;
* **TeamStats** – Tracks aggregate performance of teams in leagues, such as wins, draws, and losses;

**NOTE: Keep in mind that Judge doesn't accept "ALTER" statement and square brackets naming (when the names are not keywords).**

**NOTE: Please keep in mind that in case you have to work with a date, you have to use the exact same data type, described in the models tables. If you don't use the correct type, the Judge system won't accept your submission as correct.**

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

### ****Leagues****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Name** | **String** up to **50** symbols, **Unicode** | **Null** is **not** allowed |

### Teams

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Name** | **String** up to **50** symbols, **Unicode** | **Null** is **not** allowed, **Unique** |
| **City** | **String** up to **50** symbols, **Unicode** | **Null** is **not** allowed |
| **LeagueId** | **Integer** from **0** to **2,147,483,647** | **Null** is **not** allowed |

### Players

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Name** | **String** up to **100** symbols, **Unicode** | **Null** is **not** allowed |
| **Position** | **String** up to **20** symbols, **Unicode** | **Null** is **not** allowed |

### ****Matches****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **HomeTeamId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Teams**, **Null** is **not** allowed |
| **AwayTeamId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Teams**, **Null** is **not** allowed |
| **MatchDate** | **DateTime2** | **Null** is **not** allowed |
| **HomeTeamGoals** | **Integer** from **0** to **2,147,483,647** | **Default** is **0**, **Null** is **not** allowed |
| **AwayTeamGoals** | **Integer** from **0** to **2,147,483,647** | **Default** is **0**, **Null** is **not** allowed |
| **LeagueId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Leagues**, **Null** is **not** allowed |

### ****PlayersTeams****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **PlayerId** | **Integer** from **0** to **2,147,483,647** | **PK**, Unique table identification, Relationship with table **Players**, **Null** is **not** allowed |
| **TeamId** | **Integer** from **0** to **2,147,483,647** | **PK**, Unique table identification, Relationship with table **Teams**, **Null** is **not** allowed |

### ****PlayerStats****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **PlayerId** | **Integer** from **0** to **2,147,483,647** | **PK**, Unique table identification, Relationship with table **Players**, **Null** is **not** allowed |
| **Goals** | **Integer** from **0** to **2,147,483,647** | **Default** is **0**, **Null** is **not** allowed |
| **Assists** | **Integer** from **0** to **2,147,483,647** | **Default** is **0**, **Null** is **not** allowed |

### ****TeamStats****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **TeamId** | **Integer** from **0** to **2,147,483,647** | **PK**, Unique table identification, Relationship with table **Teams**, **Null** is **not** allowed |
| **Wins** | **Integer** from **0** to **2,147,483,647** | **Default** is **0**, **Null** is **not** allowed |
| **Draws** | **Integer** from **0** to **2,147,483,647** | **Default** is **0**, **Null** is **not** allowed |
| **Losses** | **Integer** from **0** to **2,147,483,647** | **Default** is **0**, **Null** is **not** allowed |

## Database design

Submit all of yours **CREATE** **statements** to Judge (only the creation of tables).

# Section 2. DML (10 pts)

**Before you start, you have to import "DataSet.sql". If you have created the structure correctly, the data should be successfully inserted.**

In this section, you have to do some data manipulations:

## Insert

Let's **insert** some sample data into the database. Write a **query to add the following records** into the corresponding tables. All IDs (**Primary Keys**) should be **auto-generated**.

### ****Leagues****

|  |
| --- |
| **Name** |
| Eredivisie |

### ****Teams****

|  |  |  |
| --- | --- | --- |
| **Name** | **City** | **LeagueId** |
| PSV | Eindhoven | 6 |
| Ajax | Amsterdam | 6 |

### ****Players****

|  |  |
| --- | --- |
| **Name** | **Position** |
| Luuk de Jong | Forward |
| Josip Sutalo | Defender |

### ****Matches****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **HomeTeamId** | **AwayTeamId** | **MatchDate** | **HomeTeamGoals** | **AwayTeamGoals** | **LeagueId** |
| 98 | 97 | '2024-11-02 20:45:00' | 3 | 2 | 6 |

### ****PlayersTeams****

|  |  |
| --- | --- |
| **PlayerId** | **TeamId** |
| 2305 | 97 |
| 2306 | 98 |

### ****PlayerStats****

|  |  |  |
| --- | --- | --- |
| **PlayerId** | **Goals** | **Assists** |
| 2305 | 2 | 0 |
| 2306 | 2 | 0 |

### ****TeamStats****

|  |  |  |  |
| --- | --- | --- | --- |
| **TeamId** | **Wins** | **Draws** | **Losses** |
| 97 | 15 | 1 | 3 |
| 98 | 14 | 3 | 2 |

## Update

**Update Player Statistics for Forwards in La Liga**

In this task, you will update the PlayerStats table by **adding one goal** to the stats of **all** **Forwards** who **play in teams** that belong to **La Liga**.

## Delete

**Remove All Players in the Eredivisie and Handle Related Data**

Delete the players **Luuk de Jong** and **Josip Sutalo** from teams in the **Eredivisie** league. Ensure that any related data (e.g., player stats, team assignments) is handled appropriately to maintain database integrity.

# Section 3. Querying (40 pts)

**You need to start with a fresh dataset, so recreate your DB and import the sample data again ("DataSet.sql**"**).**

## Matches by Goals and Date

Select all **matches**  from the Matches table, where the **total number of goals** scored in the match is **5 or more**. The results should be ordered by the **total number of goals scored in the match** (**descending**), and then by the **match date** (**ascending**). The **match date** should be formatted in the **'yyyy-MM-dd'** format in the query results.

Required columns:

* **MatchDate** (formatted as **'yyyy-MM-dd'**)
* **HomeTeamGoals**
* **AwayTeamGoals**
* **TotalGoals** (calculated as HomeTeamGoals + AwayTeamGoals)

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **MatchDate** | **HomeTeamGoals** | **AwayTeamGoals** | **TotalGoals** |
| 2024-08-25 | 2 | 6 | 8 |
| 2024-08-26 | 4 | 3 | 7 |
| 2024-08-30 | 4 | 3 | 7 |
| 2024-08-31 | 7 | 0 | 7 |
| 2024-09-14 | 1 | 6 | 7 |
| 2024-09-22 | 4 | 3 | 7 |
| 2024-08-17 | 1 | 5 | 6 |
| ... | ... | ... | ... |

## Players with Common Part in Their Names

Write a query to retrieve the **names of** **all players** who have **"Aaron" as part of their name**, along with the name of the **city where their team is located**. The query should return the **player’s name** and the **city they are associated with**. The results must be sorted by the player’s **name in ascending order**.

Required columns:

* **Name**
* **City**

### Example

|  |  |
| --- | --- |
| **Name** | **City** |
| Aaron Cresswell | London |
| Aaron Hickey | London |
| Aaron Ramsdale | Southampton |
| Aaron Wan-Bissaka | London |
| Max Aarons | Bournemouth |

## Players in Teams Situated in London

Write a query to retrieve **all players** who are currently playing in **teams located in** **London**. The query should return the player’s **ID**, **name**, and **position**. Additionally, the results must be **ordered by the player’s** **name in ascending order**..

Required columns:

* **Id**
* **Name**
* **Position**

### Example

|  |  |  |
| --- | --- | --- |
| **Id** | **Name** | **Position** |
| 1296 | Aaron Cresswell | Defender |
| 1222 | Aaron Hickey | Defender |
| 1285 | Aaron Wan-Bissaka | Defender |
| 1309 | Adam Wharton | Midfielder |
| 1154 | Adama Traoré | Midfielder |
| 1158 | Alex Iwobi | Forward |
| ... | ... | ... |

## First 10 Matches in Early September

Write a query to retrieve the **first 10 matches** that were played between **1st September 2024** and **15th September 2024**, in leagues with **even-numbered IDs**.

* The query should return the match details, including the **home team**, **away team**, **league name**, and **match date**.
* The **MatchDate** must be **formatted** as **'yyyy-MM-dd'** (e.g., 2024-09-01)
* The results must be sorted in **ascending order** by MatchDate, and for matches on the same date, sort them by the HomeTeamName alphabetically.

Required columns:

* **HomeTeamName**
* **AwayTeamName**
* **LeagueName**
* **MatchDate**

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **HomeTeamName** | **AwayTeamName** | **LeagueName** | **MatchDate** |
| Genoa | Verona | Serie A | 2024-09-01 |
| Juventus | Roma | Serie A | 2024-09-01 |
| Udinese | Como | Serie A | 2024-09-01 |
| Bayer Leverkusen | RB Leipzig | Bundesliga | 2024-09-01 |
| Bayern Munich | SC Freiburg | Bundesliga | 2024-09-01 |
| Fiorentina | Monza | Serie A | 2024-09-01 |
| Heidenheim | Augsburg | Bundesliga | 2024-09-01 |
| Borussia Dortmund | Heidenheim | Bundesliga | 2024-09-13 |
| Como | Bologna | Serie A | 2024-09-14 |
| Milan | Venezia | Serie A | 2024-09-14 |

## Best Guest Teams

Write a query to find **all teams** that have scored **at least 6 goals** **as the away team** in **all leagues and matches**. Order the results by:

* **Total away goals** in **descending** order
* If two teams have the **same number of goals**, order them **alphabetically by team name**

Required columns:

* **Id**
* **Name**
* **TotalAwayGoals**

### Example

|  |  |  |
| --- | --- | --- |
| **Id** | **Name** | **TotalAwayGoals** |
| 61 | Bayern Munich | 14 |
| 80 | Marseille | 8 |
| 62 | Bayer Leverkusen | 7 |
| 44 | Chelsea | 7 |
| … | … | … |

## Average Scoring Rate

Write a query to calculate the **average number of goals scored per match** for each league. The scoring rate includes goals scored by both home and away teams in all matches within the league. Display the name of the league and the average scoring rate, rounded to two decimal places. The results must be sorted in **descending order** by the average scoring rate

**HINT: Use ROUND to format the average scoring rate to two decimal places**

**HINT: In SQL, when dividing two integers, the result is also an integer (whole number) because SQL performs integer division by default. To get a decimal (floating-point) result, at least one value in the division must be explicitly converted to FLOAT or DECIMAL. Use CAST() to ensure correct calculations.**

Required columns:

* **LeagueName**
* **AvgScoringRate**

### Example

|  |  |
| --- | --- |
| **LeagueName** | **AvgScoringRate** |
| Bundesliga | 3.5 |
| Ligue 1 | 2.94 |
| Premier League | 2.65 |
| Serie A | 2.5 |
| La Liga | 2.38 |

# Section 4. Programmability (20 pts)

## League Top Scorrer

Create a user-defined function named **udf\_LeagueTopScorer** that receives the **name of a league** as input.

The function should return the **name(s) of the top scorer(s)** in the given league along with the **number of goals** they scored.

* If **multiple players share the highest goal count** in the league, the function should **return all of them**.
* **HINT: To identify the player(s) with the most goals, think about how to filter only those who have the highest goal count in the league.** 
  + **One approach is to compare each player's goal count to the maximum recorded value**
  + **Another approach involves ranking all players**

### Examples

|  |  |
| --- | --- |
| **Query** | |
| **SELECT dbo.udf\_LeagueTopScorer('Serie A')** | |
| **Example Output 1** | |
| **PlayerName** | **TotalGoals** |
| **Mateo Retegui** | **14** |
| **Query** | |

## Additionally, you can test the function by editing the goal statistics of Erling Haaland and Alexander Isak to match the current top scorer, Mohamed Salah, who has 18 goals:

*-- Update goals for Erling Haaland*

UPDATE PlayerStats

SET Goals = 18

WHERE PlayerId = (SELECT p.Id FROM Players p WHERE p.Name = 'Erling Haaland');

*-- Update goals for Alexander Isak*

UPDATE PlayerStats

SET Goals = 18

WHERE PlayerId = (SELECT p.Id FROM Players p WHERE p.Name = 'Alexander Isak');

|  |  |
| --- | --- |
| **Query** | |
| **SELECT dbo.udf\_LeagueTopScorer('Premier League')** | |
| **Example Output 1** | |
| **PlayerName** | **TotalGoals** |
| Mohamed Salah | **18** |
| Alexander Isak | **18** |
| Erling Haaland | **18** |

## Search for Teams from a Specific City

Create a stored procedure named **usp\_SearchTeamsByCity** that receives a **city name** as input. The procedure should return the following information about all teams located in the specified city:

* TeamName
* LeagueName
* City

**Order** the results **by Team Name** (**ascending**)

### Example

|  |  |  |
| --- | --- | --- |
| **Query** | | |
| **EXEC usp\_SearchTeamsByCity 'London'** | | |
| **Output** | | |
| **TeamName** | **LeagueName** | **City** |
| Arsenal | Premier League | London |
| Brentford | Premier League | London |
| Chelsea | Premier League | London |
| … | … | … |