# Exercises: Built-in Functions

This document defines the **exercise assignments** for the ["Databases Basics - MSSQL" course @ Software University.](https://softuni.bg/courses/databases-basics-ms-sql-server)

# Part I – Queries for SoftUni Database

## Find Names of All Employees by First Name

Write a SQL all employees whose **first name starts with** **“SA”.**

SELECT FirstName, LastName

FROM Employees

WHERE FirstName LIKE 'SA%'

### Example

|  |  |
| --- | --- |
| **FirstName** | **LastName** |
| Sariya | Harnpadoungsataya |
| Sandra | Reategui Alayo |
| … | … |

## Find Names of All employees by Last Name

Write a SQL query to find **first** and **last names** of all employees whose **last name contains “ei”.**

SELECT FirstName, LastName

FROM Employees

WHERE LastName LIKE '%ei%'

### Example

|  |  |
| --- | --- |
| **FirstName** | **LastName** |
| Kendall | Keil |
| Christian | Kleinerman |
| … | … |

## Find First Names of All Employees

Write a SQL query to find the **first names** of all employees in the **departments** with **ID 3 or 10** and whose **hire year** is **between 1995 and 2005 inclusive**.

SELECT FirstName

FROM Employees

WHERE DepartmentID IN (3, 10)

AND

DATEPART(YEAR, HireDate) BETWEEN 1995 AND 2005

### Example

|  |
| --- |
| **FirstName** |
| Deborah |
| Wendy |
| Candy |
| … |

## Find All Employees Except Engineers

Write a SQL query to find the **first** and **last names** of all employees whose **job titles does not contain** “**engineer**”.

SELECT FirstName, LastName

FROM Employees

WHERE JobTitle NOT LIKE '%engineer%'

### Example

|  |  |
| --- | --- |
| **FirstName** | **LastName** |
| Guy | Gilbert |
| Kevin | Brown |
| Rob | Walters |
| … | … |

## Find Towns with Name Length

Write a SQL query to find town names that are **5** or **6 symbols long** and **order** them **alphabetically by town name**.

SELECT [Name]

FROM Towns

WHERE LEN([Name]) IN (5, 6)

ORDER BY [NAME] ASC

### Example

|  |
| --- |
| **Name** |
| Berlin |
| Duluth |
| Duvall |
| … |

## Find Towns Starting With

Write a SQL query to find all towns that **start with** letters **M**, **K**, **B** or **E**. Order them **alphabetically** by town name.

SELECT TownID, [Name]

FROM Towns

WHERE LEFT([Name], 1) IN ('M', 'K', 'B', 'E')

ORDER BY [NAME] ASC

ИЛИ

SELECT TownID, [Name]

FROM Towns

WHERE [Name] LIKE '[MKBE]%'

ORDER BY [NAME]

### Example

|  |  |
| --- | --- |
| **TownID** | **Name** |
| 5 | Bellevue |
| 31 | Berlin |
| 30 | Bordeaux |
| … | … |

## Find Towns Not Starting With

Write a SQL query to find all towns that **does not start with** letters **R**, **B** or **D**. Order them **alphabetically** by name.

SELECT TownID, [Name]

FROM Towns

WHERE [Name] LIKE '[^R,B,D]%'

ORDER BY [NAME]

ИЛИ

SELECT TownID, [Name]

FROM Towns

WHERE LEFT([Name], 1) NOT IN ('R', 'D', 'B')

ORDER BY [NAME]

### Example

|  |  |
| --- | --- |
| **TownID** | **Name** |
| 2 | Calgary |
| 23 | Cambridge |
| 15 | Carnation |
| … | … |

## Create View Employees Hired After 2000 Year

Write a SQL query to create view **V\_EmployeesHiredAfter2000** with **first and last name** to all employees **hired after 2000 year.**

CREATE VIEW V\_EmployeesHiredAfter2000 AS

SELECT FirstName, LastName

FROM Employees

WHERE DATEPART(Year, HireDate) > 2000

### Example

|  |  |
| --- | --- |
| **FirstName** | **LastName** |
| Steven | Selikoff |
| Peter | Krebs |
| Stuart | Munson |
| ... | ... |

## Length of Last Name

Write a SQL query to find the **names of all employees** whose **last name** is **exactly** **5 characters long.**

### Example

|  |  |
| --- | --- |
| **FirstName** | **LastName** |
| Kevin | Brown |
| Terri | Duffy |
| Jo | Brown |
| Diane | Glimp |
| … | … |

## Rank Employees by Salary

Write a query that **ranks** all employees using **DENSE\_RANK**. In the DENSE\_RANK function, employees need to be **partitioned** by **Salary** and **ordered** by **EmployeeID**. You need to find **only** the employees whose **Salary** is between 10000 and 50000 and **order** them by **Salary** in **descending** **order**.

SELECT EmployeeID, FirstName, LastName, Salary,

DENSE\_RANK() OVER(PARTITION BY Salary ORDER BY EmployeeId) AS [Rank]

FROM Employees

WHERE Salary BETWEEN 10000 AND 50000

ORDER BY Salary DESC

### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **LastName** | **Salary** | **Rank** |
| 268 | Stephen | Jiang | 48100.00 | 1 |
| 284 | Amy | Alberts | 48100.00 | 2 |
| 288 | Syed | Abbas | 48100.00 | 3 |
| … | … | … | … | … |

## Find All Employees with Rank 2 \*

Use the query from the **previous** problem and **upgrade** it, so that it finds **only** the employees whose **Rank** is 2 and again, **order** them by **Salary (descending)**.

SELECT \* FROM (SELECT EmployeeID, FirstName, LastName, Salary,

DENSE\_RANK() OVER(PARTITION BY Salary ORDER BY EmployeeId) AS [Rank]

FROM Employees

WHERE Salary BETWEEN 10000 AND 50000) AS temp

WHERE temp.[Rank] = 2

ORDER BY temp.[Salary] DESC

### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **LastName** | **Salary** | **Rank** |
| 284 | Amy | Alberts | 48100.00 | 2 |
| 292 | Martin | Kulov | 48000.00 | 2 |
| 71 | Wendy | Kahn | 43300.00 | 2 |
| … | … | … | … | … |

# Part II – Queries for Geography Database

## Countries Holding ‘A’ 3 or More Times

Find all countries that holds the letter 'A' in their name at least 3 times (case insensitively), sorted by ISO code. Display the country name and ISO code.

SELECT CountryName, IsoCode

FROM Countries

WHERE CountryName LIKE '%a%a%a%'

ORDER BY IsoCode ASC

### Example

|  |  |
| --- | --- |
| **Country Name** | **ISO Code** |
| Afghanistan | AFG |
| Albania | ALB |
| … | … |

## Mix of Peak and River Names

Combine all peak names with all river names, so that the **last letter** of each **peak name** is the **same** **as** the **first letter** of its corresponding **river** **name**. Display the peak names, river names, and the obtained mix (mix should be in lowercase). **Sort** the results **by** the **obtained mix**.

SELECT p.PeakName, r.RiverName, LOWER(CONCAT(LEFT(p.PeakName, LEN(p.PeakName) -1), r.RiverName)) AS [Mix]

FROM Peaks AS p, Rivers AS r

WHERE RIGHT(p.PeakName, 1) = LEFT(r.RiverName, 1)

ORDER BY [Mix]

### Example

|  |  |  |
| --- | --- | --- |
| **PeakName** | **RiverName** | **Mix** |
| Aconcagua | Amazon | aconcaguamazon |
| Aconcagua | Amur | aconcaguamur |
| Banski Suhodol | Lena | banski suhodolena |
| … | … | … |

# Part III – Queries for Diablo Database

## Games from 2011 and 2012 year

Find the top 50 games ordered by start date, then by name of the game. Display only games from 2011 and 2012 year. Display start date in the format “**yyyy-MM-dd**”.

SELECT TOP(50) [Name], Convert(varchar(10), [Start], 120) as [Start date]

FROM Games

WHERE DATEPART(YEAR, [Start]) IN (2011, 2012)

ORDER BY [Start date], [Name]

ИЛИ

SELECT TOP 50 Name, FORMAT(Start,'yyyy-MM-dd') AS [Start Date]

FROM Games

WHERE (SELECT YEAR(Start)) IN (2011,2012)

ORDER BY [Start Date], Name

ИЛИ

SELECT TOP(50) [Name], FORMAT([Start], 'yyyy-MM-dd') AS [Start date]

FROM Games

WHERE DATEPART(YEAR, [Start]) IN (2011, 2012)

ORDER BY [Start date], [Name]

### Example

|  |  |
| --- | --- |
| **Name** | **Start** |
| Rose Royalty | 2011-01-05 |
| London | 2011-01-13 |
| Broadway | 2011-01-16 |
| … | … |

## User Email Providers

Find all users along with information about their email providers. Display the username and email provider. Sort the results by email provider alphabetically, then by username.

SELECT Username, SUBSTRING(Email, CHARINDEX('@', Email)+1, Len(Email)) AS [Email Provider]

FROM Users

ORDER BY [Email Provider], Username

### Example

|  |  |
| --- | --- |
| **Username** | **Email Provider** |
| Pesho | abv.bg |
| monoxidecos | astonrasuna.com |
| bashsassafras | balibless |
| … | … |

## Get Users with IPAdress Like Pattern

Find all users along with their IP addresses sorted by username alphabetically. Display only rows that IP address matches the pattern: “**\*\*\*.1^.^.\*\*\***”.

Legend: **\*** - one symbol, **^** - one or more symbols

SELECT Username, IpAddress

FROM Users

WHERE IpAddress LIKE '\_\_\_.1%.%.\_\_\_'

ORDER BY Username

### Example

|  |  |
| --- | --- |
| **Username** | **IP Address** |
| bindbawdy | 192.157.20.222 |
| evolvingimportant | 223.175.227.173 |
| inguinalself | 255.111.250.207 |
| … | … |

## Show All Games with Duration and Part of the Day

Find all games with part of the day and duration sorted by game name alphabetically then by duration (alphabetically, not by the timespan) and part of the day (all ascending). **Parts of the day** should be **Morning** (time is >= 0 and < 12), **Afternoon** (time is >= 12 and < 18), **Evening** (time is >= 18 and < 24). **Duration** should be **Extra** **Short** (smaller or equal to 3), **Short** (between 4 and 6 including), **Long** (greater than 6) and **Extra Long** (without duration).

SELECT [Name] as [Game],

CASE

WHEN

DATEPART(HOUR, Start) BETWEEN 0 AND 11 THEN 'Morning'

WHEN

DATEPART(HOUR, Start) BETWEEN 12 AND 17 THEN 'Afternoon'

WHEN

DATEPART(HOUR, Start) BETWEEN 18 AND 23 THEN 'Evening'

END AS [Part of the Day],

CASE

WHEN

Duration <= 3 THEN 'Extra Short'

WHEN

Duration >= 4 AND Duration <= 6 THEN 'Short'

WHEN

Duration > 6 THEN 'Long'

WHEN

Duration IS NULL THEN 'Extra Long'

END as [Duration]

FROM GAMES

ORDER BY [Game], [Duration]

### Example

|  |  |  |
| --- | --- | --- |
| **Game** | **Part of the Day** | **Duration** |
| Ablajeck | Morning | Long |
| Ablajeck | Afternoon | Short |
| Abregado Rae | Afternoon | Long |
| Abrion | Morning | Extra Short |
| Acaeria | Evening | Long |
| … | … | … |

# Part IV – Date Functions Queries

## Orders Table

You are given a table **Orders(Id, ProductName, OrderDate)** filled with data. Consider that the **payment** for that order must be accomplished **within 3 days after the order date**. Also the **delivery date is up to 1 month**. Write a query to show each product’s **name**, **order date**, **pay and deliver due dates**.

SELECT ProductName,

OrderDate,

DATEADD(Day, 3, OrderDate) AS [Pay Due],

DATEADD(Month, 1, OrderDate) AS [Deliver Due]

FROM Orders

### Original Table

|  |  |  |
| --- | --- | --- |
| **Id** | **ProductName** | **OrderDate** |
| 1 | Butter | 2016-09-19 00:00:00.000 |
| 2 | Milk | 2016-09-30 00:00:00.000 |
| 3 | Cheese | 2016-09-04 00:00:00.000 |
| 4 | Bread | 2015-12-20 00:00:00.000 |
| 5 | Tomatoes | 2015-12-30 00:00:00.000 |
| … | … | … |

### Output

|  |  |  |  |
| --- | --- | --- | --- |
| **ProductName** | **OrderDate** | **Pay Due** | **Deliver Due** |
| Butter | 2016-09-19 00:00:00.000 | 2016-09-22 00:00:00.000 | 2016-10-19 00:00:00.000 |
| Milk | 2016-09-30 00:00:00.000 | 2016-10-03 00:00:00.000 | 2016-10-30 00:00:00.000 |
| Cheese | 2016-09-04 00:00:00.000 | 2016-09-07 00:00:00.000 | 2016-10-04 00:00:00.000 |
| Bread | 2015-12-20 00:00:00.000 | 2015-12-23 00:00:00.000 | 2016-01-20 00:00:00.000 |
| Tomatoes | 2015-12-30 00:00:00.000 | 2016-01-02 00:00:00.000 | 2016-01-30 00:00:00.000 |
| … | … | … | … |