# Exam Preparation

A problem for exam preparation for the ["Databases Basics - MSSQL" course @ Software University](https://softuni.bg/trainings/3965/ms-sql-january-2023)

Submit your solutions in the SoftUni Judge system [here](https://judge.softuni.org/Contests/3533/Databases-MSSQL-Server-Exam-19-June-2022)

# Zoo

*Your childhood dream came true and you are invited to work in the local Zoo. Noticing your potential, you are asked to design a management system, so that they can keep track of the animals and the people, who are involved in the zoo.*

# Section 1. DDL (30 pts)

You have been given the E/R Diagram of the **Zoo**

Diagram

Description automatically generated

Create a database called **Zoo**. You need to create **7 tables**:

* **Owners** – contains information about the **owners of the animals**;
* **AnimalTypes** – contains information about the different **animal types** in the zoo;
* **Cages** – contains information about the animal **cages**;
* **Animals** – contains information about the **animals**;
* **AnimalsCages** – a many-to-many mapping table between the animals and the cages;
* **VolunteersDepartments** – contains information about the **departments of the volunteers**;
* **Volunteers** – contains information about the **volunteers**.

**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. For example, data type Date means that you have to use Date, DateTime means that you have to use DateTime. If you don't use the correct type, the Judge system won't accept your submission as correct.**

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

**NOTE: Use VARCHAR for strings, not NVARCHAR.**

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

### Owners

|  |  |  |
| --- | --- | --- |
| **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 | **Null** is **not** allowed |
| **PhoneNumber** | **String** up to **15** symbols | **Null** is **not** allowed |
| **Address** | **String** up to **50** symbols | **Null** isallowed |

### AnimalTypes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Identity** |
| **AnimalType** | **String** up to **30** symbols | **Null** is **not** allowed |

### Cages

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Identity** |
| **AnimalTypeId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **AnimalTypes**, **Null** is **not** allowed |

### Animals

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Identity** |
| **Name** | **String** up to **30** symbols | **Null** is **not** allowed |
| **BirthDate** | **Date** | **Null** is **not** allowed |
| **OwnerId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Owners**, **Null** isallowed |
| **AnimalTypeId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **AnimalTypes**, **Null** is **not** allowed |

### AnimalsCages

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **CageId** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Relationship** with table **Cages**, **Null** is **not** allowed |
| **AnimalId** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Relationship** with table **Animals**, **Null** is **not** allowed |

### VolunteersDepartments

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Identity** |
| **DepartmentName** | **String** up to **30** symbols | **Null** is **not** allowed |

### Volunteers

|  |  |  |
| --- | --- | --- |
| **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 | **Null** is **not** allowed |
| **PhoneNumber** | **String** up to **15** symbols | **Null** is **not** allowed |
| **Address** | **String** up to **50** symbols | **Null** isallowed |
| **AnimalId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Animals**; **Null** is allowed |
| **DepartmentId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **VolunteersDepartments,** **Null** is **not** allowed. |

## Database design

Submit all of your **created** **statements** to Judge (only creation of tables).

# Section 2. DML (10 pts)

**Before you start you have to import "*01.* *DDL\_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 should be **auto-generated**.

****Volunteers****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **PhoneNumber** | **Address** | **AnimalId** | **DepartmentId** |
| Anita Kostova | 0896365412 | Sofia, 5 Rosa str. | 15 | 1 |
| Dimitur Stoev | 0877564223 | null | 42 | 4 |
| Kalina Evtimova | 0896321112 | Silistra, 21 Breza str. | 9 | 7 |
| Stoyan Tomov | 0898564100 | Montana, 1 Bor str. | 18 | 8 |
| Boryana Mileva | 0888112233 | null | 31 | 5 |

****Animals****

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **BirthDate** | **OwnerId** | **AnimalTypeId** |
| Giraffe | 2018-09-21 | 21 | 1 |
| Harpy Eagle | 2015-04-17 | 15 | 3 |
| Hamadryas Baboon | 2017-11-02 | null | 1 |
| Tuatara | 2021-06-30 | 2 | 4 |

## Update

**Kaloqn Stoqnov** (a current owner, present in the database) came to the zoo to adopt all the **animals**, who **don't** **have an owner**. **Update** the records by putting to those animals **the correct OwnerId.**

## Delete

The Zoo decided to close one of the **Volunteers Departments - Education program assistant**. Your job is to **delete** this **department** from the database.

**NOTE:** Keep in mind that there could be **foreign key constraint conflicts**!

# Section 3. Querying (40 pts)

**You need to start with a fresh dataset, so recreate your DB and import the sample data again (01. DDL\_Dataset.sql). DO NOT CHANGE OR INCLUDE DATA FROM DELETE, INSERT AND UDATE TASKS!!!**

## Volunteers

Extract information about all the **Volunteers** – **name**, **phone number**, **address**, **id** of the **animal**, they are responsible to and **id** of the **department** they are involved into. **Order** the result by **name** of the **volunteer** (**ascending**), then by the **id** of the **animal** (**ascending**) and then by the **id** of the **department** (**ascending**).

### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **PhoneNumber** | **Address** | **AnimalId** | **DepartmentId** |
| Anton Antonov | 0877456123 | Varna, 2 Dobrotitsa str. | 11 | 3 |
| Boyan Boyanov | 0896321546 | Plovdiv, 15 Arda str. | 14 | 1 |
| Darina Petrova | 0889654236 | Sofia, 39 Bratya Buxton str. | 31 | 3 |
| Dilyana Stoeva | 0889412025 | Sofia, 15 Lyulyak str. | NULL | 2 |
| Dimitrichka Stateva | 0888632123 | Sofia, 26 Vasil Levski str. | 7 | 8 |
| Gabriel Radkov | 0889745102 | Sliven, 6 Krim str. | 18 | 5 |
| … | … | … | … | … |

## Animals data

Select all **animals** and their **type**. Extract **name**, **animal type** and **birth date (**in format **'dd.MM.yyyy')**. **Order** the result by animal's **name (ascending)**.

### Example

|  |  |  |
| --- | --- | --- |
| **Name** | **AnimalType** | **BirthDate** |
| African Penguin | Birds | 17.07.2017 |
| African Spurred Tortoise | Reptiles | 26.09.2009 |
| American Kestrel | Birds | 27.04.2019 |
| Anaconda | Reptiles | 13.07.2016 |
| Axolotl | Amphibians | 21.01.2019 |
| Bald Eagle | Birds | 29.06.2014 |
| … | … | … |

## Owners and Their Animals

Extract the **animals** for each **owner**. **Find** the **top 5 owners**, who have the **biggest count of animals**. Select the **owner's** **name** and the **count** of the **animals** he owns. **Order** the result by the **count** of **animals** owned (**descending**) and then by the **owner's** **name**.

### Example

|  |  |
| --- | --- |
| **Owner** | **CountOfAnimals** |
| Kaloqn Stoqnov | 4 |
| Kiril Peshev | 4 |
| Kamelia Yancheva | 3 |
| Martin Genchev | 3 |
| Metodi Dimitrov | 3 |

## Owners, Animals and Cages

Extract information about the **owners** of **mammals**, the **name** of their **animal** and in which **cage** these animals are. Select **owner's** **name** and **animal's** **name** (in format **'owner-animal'**), owner's phone **number** and the **id** of the **cage**. **Order** the result by the **name** of the **owner** (**ascending**) and then by the **name** of the **animal** (**descending**).

### Example

|  |  |  |
| --- | --- | --- |
| **OwnersAnimals** | **PhoneNumber** | **CageId** |
| Anelia Mihova-Koala | 0897856147 | 16 |
| Borislava Kamenova-Fennec Fox | 0877477112 | 21 |
| Gergana Mancheva-Brown bear | 0897412123 | 26 |
| Kaloqn Stoqnov-Leopard | 0878325642 | 32 |
| Kaloqn Stoqnov-Elephant | 0878325642 | 37 |
| Kamelia Yancheva-Lion | 0876213799 | 7 |
| … | … | … |

## Volunteers in Sofia

Extract information about the **volunteers**, involved in **'Education program assistant'** department, who live in **Sofia**. Select their **name**, **phone number** and their **address** in Sofia (skip city's name). **Order** the result by the **name** of the volunteers (**ascending)**.

### Example

|  |  |  |
| --- | --- | --- |
| **Name** | **PhoneNumber** | **Address** |
| Dilyana Stoeva | 0889412025 | 15 Lyulyak str. |
| Kiril Kostadinov | 0896541233 | 213 Tsarigradsko shose str. |
| Yanko Totev | 0896369258 | 54 Hristo Botev str. |
| Zdravko Asenov | 0889652365 | 6 Neven str. |

## Animals for Adoption

Extract all **animals**, who does **not** have an **owner** and are **younger** than **5 years** (5 years from '01/01/2022'), **except** for the **Birds**. Select their **name**, **year of birth** and **animal type. Order** the result by **animal's name.**

### Example

|  |  |  |
| --- | --- | --- |
| **Name** | **BirthYear** | **AnimalType** |
| Banded Archer Fish | 2022 | Fish |
| Chameleon | 2018 | Reptiles |
| Desert Hairy Scorpion | 2020 | Invertebrates |
| Goliath Frog | 2020 | Amphibians |
| Koi | 2021 | Fish |
| Poison Frog | 2020 | Amphibians |

# Section 4. Programmability (20 pts)

## All Volunteers in a Department

Create a **user-defined function** named **udf\_GetVolunteersCountFromADepartment** (**@VolunteersDepartment)** that receives a **department** and returns the count of **volunteers,** who are involved in this department.

### Examples

|  |
| --- |
| **Query** |
| SELECT dbo.udf\_GetVolunteersCountFromADepartment ('Education program assistant') |
| **Output** |
| 6 |

|  |
| --- |
| **Query** |
| SELECT dbo.udf\_GetVolunteersCountFromADepartment ('Guest engagement') |
| **Output** |
| 4 |

|  |
| --- |
| **Query** |
| SELECT dbo.udf\_GetVolunteersCountFromADepartment ('Zoo events') |
| **Output** |
| 5 |

## Animals with Owner or Not

Create a **stored procedure**, named **usp\_AnimalsWithOwnersOrNot(@AnimalName).**

Extract the name of the **owner** of the given animal. If there is no owner, put **'For adoption'.**

### Example

|  |
| --- |
| **Query** |
| EXEC usp\_AnimalsWithOwnersOrNot 'Pumpkinseed Sunfish' |

### Result

|  |  |
| --- | --- |
| **Name** | **OwnersName** |
| Pumpkinseed Sunfish | Kamelia Yancheva |

### Example

|  |
| --- |
| **Query** |
| EXEC usp\_AnimalsWithOwnersOrNot 'Hippo' |

### Result

|  |  |
| --- | --- |
| **Name** | **OwnersName** |
| Hippo | For adoption |

### Example

|  |
| --- |
| **Query** |
| EXEC usp\_AnimalsWithOwnersOrNot 'Brown bear' |

### Result

|  |  |
| --- | --- |
| **Name** | **OwnersName** |
| Brown bear | Gergana Mancheva |