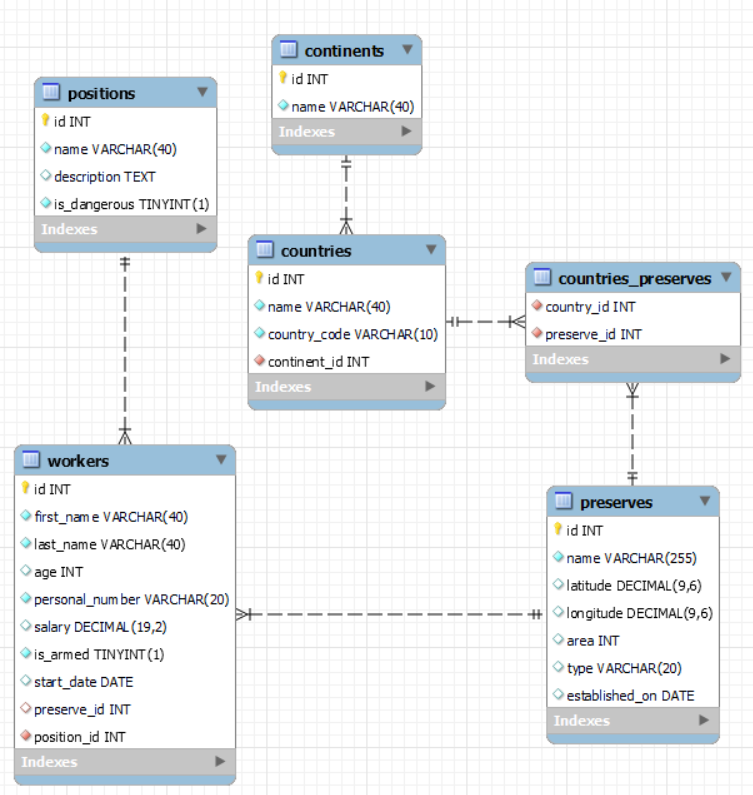
# MySQL Exam Preparation – 03 Oct 2024 Wildlife preserves around the world

[Link: Database Basics MySQL Exam](https://judge.softuni.org/Contests/4531/Database-Basics-MySQL-Regular-Exam-10-February-2024)

*Years ago, animals and plants were the masters of nature. Our ancestors lived in harmony and understanding with them. Today, unfortunately, man prevails over this idyll and enters, steals and destroys an increasingly large territory - the home of the wild world. Its inhabitants are forced to live in smaller and smaller spaces, sized by the same person with whom they once shared. These are the wildlife preserves.*

## Section 0: Database Overview

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



The **preserve’s** **Database** needs to hold information about **continents**, **countries**, **workers**, **preserves**, **and positions**.

Your task is to create a database called **preserves\_db**. Then you will have to create several **tables**.

* continents – contains information about the **continents**.
* countries – contains information about the **countries**.
  + Each country has a continent.
* preserves – contains information about the **preserves**.
  + Each **preserve** has a country.
* workers – contains information about the **workers**.
  + Each worker has a position.
* positions – contains information about the **positions**.
* countries\_preserves – a **many** to **many** **mapping** table between the **countries** and the **preserves**.

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

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

The instructions you'll be given will be the minimum needed to implement the database.

### Table Design

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

#### continents

|  |  |  |
| --- | --- | --- |
| **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 **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |

#### 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 **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| country\_code | A **string** containing a maximum of 1**0 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| continent\_id | **Integer,** from **1** to **2,147,483,647.** | Relationship with table continents.  **NULL** is **NOT** permitted**.** |

#### preserves

|  |  |  |
| --- | --- | --- |
| **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 **255 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| latitude | **DECIMAL**, up to **9 digits**, **6** of which are after the **decimal point**. |  |
| longitude | **DECIMAL**, up to **9 digits**, **6** of which are after the **decimal point**. |  |
| area | **Integer,** from **1** to **2,147,483,647.** |  |
| type | A **string** containing a maximum of **20 characters**. Unicode is **NOT** needed. |  |
| established\_on | The **DATE** of the establishment of the preserve. | **NULL** ispermitted**.** |

#### positions

|  |  |  |
| --- | --- | --- |
| **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 **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| description | A very **long** string field |  |
| is\_dangerous | It can be **true** or **false**. | **NULL** is **NOT** permitted**.** |

#### workers

|  |  |  |
| --- | --- | --- |
| **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 **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**.** |
| last\_name | A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**.** |
| age | **Integer,** from **1** to **2,147,483,647.** |  |
| personal\_number | A **string** containing a maximum of **20 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| salary | **DECIMAL**, up to **19 digits**, **2** of which are after the **decimal point**. |  |
| is\_armed | It can be **true** or **false**. | **NULL** is **NOT** permitted**.** |
| start\_date | The **DATE** the worker started his work |  |
| preserve\_id | **Integer,** from **1** to **2,147,483,647.** | Relationship with table preserves. |
| position\_id | **Integer,** from **1** to **2,147,483,647.** | Relationship with table positions. |

#### countries\_preserves

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| country\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table countries. |
| preserve\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table preserves. |

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 will not have to think of data and lose essential 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 **preserves** table, based on the **preserves** table.

For all **preserves** which are located in the **southern** hemisphere (**latitude <** **0**), **insert data** in the **preserves** tablewith the **following values**:

• **name** – **set it** to the **preserve name** followed by white space and then **"is in South Hemisphere"** text

*(name + " " + "is in South Hemisphere")*

• **latitude** – keep the same

• **longitude** – keep the same

• **area** – set it to **area** multiplied by **preserve** **id**

• **type** – set it to the **preserve** type but **in lowercase**

• **established\_on** – keep the same

### Update

Due to the dangerous nature of their work, increase the **salary** of the workers with **position\_id** - **5, 8, 11 and 13** by **500**.

### Delete

**Delete** all **preserves**, without information about their establishment.

## 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.

### Most experienced workers

Extract from the **preserves\_db** database, info about the **workers** with more than **5 years** of experience.

(experience **must be** calculated from the day the workers **started working** until **01-01-2024**)

**Order** the results by **days\_of\_experience** in **descending** order and show only the first **10** results.

#### Required Columns

* full\_name (first\_name + " " + last\_name)
* days\_of\_experience (duration of experience in days)

#### Example

|  |  |
| --- | --- |
| **full\_name** | **days\_of\_experience** |
| Jin Lee | 18937 |
| Farida Hassan | 17552 |
| ... | ... |
| Abebe Tesfaye | 12925 |
| Olivia Thomas | 12751 |

### Worker's salary

Write a query that returns: **worker\_id, first\_name**, **last\_name** , **preserve name** and **country\_code** from **table** **workers**. **Filter** only the workers whose **salary** is higher than **5000** and their **age** is lower than **50**.

**Order** the results **ascending** by **country\_code**.

#### Required Columns

* id (worker)
* first\_name
* last\_name
* preserve\_name
* country\_code

#### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **preserve\_name** | **country\_code** |
| 30 | William | Rodriguez | Los Glaciares National Park | AR |
| 4 | Amina | Al-Mansoori | Pantanal | BR |
| 53 | Sophia | Lopez | Colca Valley National Park | PE |
| 16 | Akira | Sato | Kruger National Park | ZA |

### Armed workers count

Write a query that returns: the **name** of the **preserve** and the **total** **count** of **armed** workers who worked there.

**Order** by **armed\_workers** count in **descending** order, then by **preserve** name **ascending**.

#### Required Columns

* name (preserve)
* armed\_workers

#### Example

|  |  |
| --- | --- |
| **name** | **armed\_workers** |
| Pantanal | 4 |
| Serengeti National Park | 3 |
| … | … |
| Tierra del Fuego | 1 |
| Vatnajokull National Park | 1 |

### Oldest preserves

There are many preserves around the world, but we need to find the oldest ones.

Extract from the **preserves\_db** database, the **name, country\_code** and **year** of establishment for the **five** oldest preserves, which are founded in **May.**

**Order** the results **ascending** by **established\_on.**

#### Required Columns

* name (preserve)
* country\_code
* founded\_in

#### Examples

|  |  |  |
| --- | --- | --- |
| **name** | **country\_code** | **founded\_in** |
| Kruger National Park | ZA | 1898 |
| Los Glaciares National Park | AR | 1937 |
| Torres del Paine | CL | 1959 |
| Northeast Greenland National Park | GL | 1974 |
| Sundarbans | IN | 1984 |

### Preserve categories

Let's make the size of the preserve a bit abstract by categorising it. From the database extract the **id**, **name** and **category**. If the **area** is equal to or less than **100** the user must see **"very small",** above **100** and equal to or less than **1000** it should display "**small**", above **1000** and equal to or less than **10000** it should display "**medium**", above **10000** and equal to or less than **50000** should display "**large**" and above **50000** it should display **"very large"**

**Order** the results **descending** by **area**.

#### Required Columns

* id (preserve)
* name (preserve)
* category

#### Example

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **category** |
| 2 | Northeast Greenland National Park | very large |
| 13 | Great Barrier Reef | very large |
| 7 | Tierra del Fuego | very large |
| … | … | … |
| 33 | Dolomiti Bellunesi National Park | very small |
| 27 | Hundred Islands National Park | very small |

## 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.

### Extract average salary

Create a **user defined function** with the name **udf\_average\_salary\_by\_position\_name (name VARCHAR(40))** that receives a **position name** and returns the average amount of salary for this position for all workers who practice it

#### Required Columns

* name (position)
* position\_average\_salary (average\_salary\_amount)

#### Example

|  |  |
| --- | --- |
| **Query** | |
| SELECT p.name, udf\_average\_salary\_by\_position\_name('Forester') as position\_average\_salary FROM positions p  WHERE p.name = 'Forester' | |
| name | position\_average\_salary |
| Forester | 5620.55 |

### Improving the standard of living

Create a stored procedure udp\_increase\_salaries\_by\_country which accepts the following parameters:

* country\_name (VARCHAR(40))

Extract data on all workers who work in all preserves of the territory of the given **country** and increase their salaries by **5%**

#### Result

|  |
| --- |
| **Query** |
| CALL increase\_salaries\_by\_country (Germany); |
| This execution will update the salaries of 3 workers who are working in preserves in Germany |

#### Result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **first\_name** | **last\_name** | **->** | **salary** before | **salary** after |
| Lucas | Anderson | -> | 3800.25 | 3990.26 |
| Kwesi | Asante | -> | 3800.25 | 3990.26 |
| Evelyn | Lea | -> | 4100.25 | 4305.26 |