# PostgreSQL Regular Exam - 15 June 2024

# Social Media Profiles

## *You've been chosen to assist renowned social media personalities in managing their online presence. Your expertise in database management will enable you to create and populate a cutting-edge database, ensuring seamless access to information based on specified criteria. Understanding the database structure and populating it with relevant data is paramount for effective information retrieval.*

## Section 0: Database Overview

You are given an Entity/Relationship Diagram of the **Social Media Profiles**. This diagram illustrates the connections among different entities within **Social Media Profiles**, offering a visual depiction of the database structure:

A computer screen shot of a computer

Description automatically generated

The **Social Media Profiles** must hold information about **accounts**, **addresses**, **photos**, **comments**, **accounts\_photos**, and **likes**.

Your task is to set up a database named sm\_profiles (**Social Media Profiles**) with the following **tables**:

* accounts – contains information about the **accounts**.
* addresses – contains information about the **addresses**.
* photos – contains information about the **photos**.
* comments – contains information about the **comments**.
* accounts\_photos – a **many-to-many** **mapping** table between the **accounts** and the **photos**.
* likes – contains information about the **likes**.

## Section 1: Data Definition Language (DDL) - 30 Pts

Make sure you implement the **database** **tables** correctly.

**Important Note:** When working with dates, please adhere strictly to the specified data types in the model tables. For example, if a column is defined as type '**DATE**,' ensure that you utilize the '**DATE**' data type. Similarly, if a column is designated as '**TIMESTAMP**,' use the '**TIMESTAMP**' data type. Failure to use the correct data type may result in your submission being rejected by the Judge system.

### Table Design

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

Submit only your **CREATE** statements for all tables to the Judge System.

#### accounts

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**,from **0** to **2,147,483,647** | **Primary Key**, Unique table identification, Auto-incremented |
| username | **String** containing a maximum of **30 characters** | **NULL** is **NOT** permitted**. UNIQUE** values. |
| password | **String** containing a maximum of **30 characters** | **NULL** is **NOT** permitted**.** |
| email | **String** containing a maximum of **50 characters** | **NULL** is **NOT** permitted**.** |
| gender | **Exactly 1 character** (**'M' or 'F'** char) | **NULL** is **NOT** permitted**.** The column must contain an **'M' or 'F'** char. |
| age | **Integer**,from **0** to **2,147,483,647** | **NULL** is **NOT** permitted**.** |
| job\_title | A **string** containing a maximum of **40 characters** | **NULL** is **NOT** permitted**.** |
| ip | A **string** containing a maximum of **30 characters** | **NULL** is **NOT** permitted**.** |

#### addresses

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**,from **0** to **2,147,483,647** | **Primary Key**, Unique table identification, Auto-incremented |
| street | **String** containing a maximum of **30 characters** | **NULL** is **NOT** permitted. |
| town | **String** containing a maximum of **30 characters** | **NULL** is **NOT** permitted**.** |
| country | **String** containing a maximum of **30 characters** | **NULL** is **NOT** permitted**.** |
| account\_id | **Integer**,from **0** to **2,147,483,647** | Relationship with table accounts. Cascade Operations. **NULL** is **NOT** permitted**.** |

#### photos

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**,from **0** to **2,147,483,647** | **Primary Key**, Unique table identification, Auto-incremented |
| description | A variable **unlimited length** of **text** | **NULL** is permitted. |
| capture\_date | **TIMESTAMP**, indicatesthe **date** and **time** | **NULL** is **NOT** permitted**.** |
| views | **Integer**,from **0** to **2,147,483,647** | **The DEFAULT value is 0.** The column must always have a value **greater than or equal to zero. NULL** is **NOT** permitted. |

#### comments

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**,from **0** to **2,147,483,647** | **Primary Key**, Unique table identification, Auto-incremented |
| content | **String** containing a maximum of **255 characters**. | **NULL** is **NOT** permitted**.** |
| published\_on | **TIMESTAMP**, indicatesthe **date** and **time** | **NULL** is **NOT** permitted**.** |
| photo\_id | **Integer**,from **0** to **2,147,483,647** | Relationship with table photos. Cascade Operations. **NULL** is **NOT** permitted**.** |

#### accounts\_photos

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| account\_id | **Integer**,from **0** to **2,147,483,647** | Relationship with table accounts.Cascade Operations. **NULL** is **NOT** permitted**.** |
| photo\_id | **Integer**,from **0** to **2,147,483,647** | Relationship with table photos. Cascade Operations. **NULL** is **NOT** permitted**.** |
| - | **-** | **Composite Primary Key** on both columns (**account\_id** and **photo\_id**) |

#### likes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer**,from **0** to **2,147,483,647** | **Primary Key**, Unique table identification, Auto-incremented |
| photo\_id | **Integer**,from **0** to **2,147,483,647** | Relationship with table **photos**. Cascade Operations. **NULL** is **NOT** permitted**.** |
| account\_id | **Integer**,from **0** to **2,147,483,647** | Relationship with table **accounts**. Cascade Operations. **NULL** is **NOT** permitted**.** |

## Section 2: Data Manipulation Language (DML) - 10 Pts

Before starting, ensure to import the **'dataset.sql'** file. Successful insertion of data is contingent upon the proper creation of the database structure.

This section entails executing various data manipulation tasks:

### Insert

You need to insert data records into the '**addresses**' table, derived from the '**accounts**' table.

For **accounts** identified as **female gender**, insert corresponding data into the '**addresses**' table with the following values:

* street -set it tothe **username** of the **account**.
* town - set it to the **password** of the **account**.
* country - set it to the **ip** of the **account**.
* account\_id - set it to the **age** of the **account**.

#### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **street** | **town** | **country** | **account\_id** |
| … | … | … | … | … |
| 99 | 43 Sommers Street | Komsomolskiy | Russia | 96 |
| 100 | 34 Nobel Point | Voloka | Ukraine | 87 |
| 101 | hgrigoryov4 | U9aH8s4wFns | 50.162.216.179 | 4 |
| 102 | wsline6 | oeu5Gbdg0 | 148.176.123.206 | 21 |
| 103 | bvickress7 | uof8US | 60.134.160.216 | 47 |
| … | … | … | … | … |

### Update

**Rename** those **countries** from table'**addresses**', which meet the following conditions:

* If the country name starts with 'B' - **change** it to **'Blocked'**.
* If the country name starts with 'T' - **change** it to **'Test'**.
* If the country name starts with 'P' - **change** it to **'In Progress'**.

#### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **street** | **town** | **country** | **account\_id** |
| … | … | … | … | … |
| 52 | 98 North Way | Boise | United States | 63 |
| 53 | 139 Canary Crossing | Pak Phanang | Test | 84 |
| 54 | 1737 Cody Plaza | Qijing | China | 70 |
| 55 | 6 Ramsey Place | Rawa Mazowiecka | In Progress | 48 |
| 56 | 27400 Di Loreto Terrace | Marechal Deodoro | Blocked | 79 |
| … | … | … | … | … |

### Delete

As you may recall, during our initial work, data was inserted and updated. Now, there is a need to **remove** certain **addresses** from the database.

**Delete** all **addresses** from table '**addresses**', where the **id** is divisible by **2 and** the **street** name **contains** the **letter 'r'** (case-insensitive).

#### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **street** | **town** | **country** | **account\_id** |
| … | … | … | … | … |
| 51 | 16416 Pierstorff Parkway | Mulyorejo | Indonesia | 45 |
| 53 | 139 Canary Crossing | Pak Phanang | Test | 84 |
| 54 | 1737 Cody Plaza | Qijing | China | 70 |
| 55 | 6 Ramsey Place | Rawa Mazowiecka | In Progress | 48 |
| 57 | 7991 Esch Trail | Bartniczka | In Progress | 56 |
| … | … | … | … | … |

## Section 3: Querying - 40 Pts

**Important Note**: Now, we'll conduct some data extraction tasks. Please ensure that the database is cleared of any manipulations from the previous operations in the Data Manipulation Language (DML) section. **Insert** the provided **dataset** **again** to maintain consistency with the examples in this section.

### Accounts

Extract information about **accounts**. Select only those whose **age** is **greater** than or **equal** to **18** years and the **length** of their **username** is **greater** than **9** symbols.

**Order** the results by **age descending** then by **username ascending**.

#### Required Columns

* username
* gender
* age

#### Example

|  |  |  |
| --- | --- | --- |
| **username** | **gender** | **age** |
| chartfordz | M | 100 |
| mcaygill1d | F | 100 |
| gjoannidi3 | M | 98 |
| prestorickx | M | 98 |
| cbaythorp8 | M | 96 |
| … | … | … |

### Top 3 Most Commented Photos

Extract from the database, the top **3** **most commented** **photos** with their **count** of **comments**. **Exclude** those photos that have **no description**.

Sort the results by **comments\_count descending**, then by **photo** **id** in **ascending** order.

#### Required Columns

* id as photo\_id
* capture\_date
* description
* comments\_count

#### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **photo\_id** | **capture\_date** | **description** | **comments\_count** |
| 23 | 2019-10-13 14:13:42 | Duis bibendum, felis sed interdum venenatis, turpis enim blandit … | 4 |
| 25 | 2019-07-20 13:08:03 | In congue. Etiam justo. Etiam pretium… | 4 |
| 14 | 2020-02-16 13:49:08 | Praesent blandit. Nam nulla. Integer pede justo… | 3 |

\*The **description** has been shortened for illustrative purposes. You are not supposed to reformat it for this task.

### Lucky Accounts

Identify **Lucky Accounts** where the **account id** matches their **photo id**.

Extract the **id** **concatenated** with the **username** (as **id\_username**) and the **email** for all such **lucky accounts**.

**Sort** the results in **ascending** order by **account id**.

#### Required Columns

* id\_username (id + " " + username)
* email

#### Example

|  |  |
| --- | --- |
| **id\_username** | email |
| 12 aroccob | dpendrichb@hhs.gov |
| … | … |

### Count Likes and Comments

Retrieve **photo IDs** along with associated **likes count** and **comments count** from the database.

**Sort** the results by the **number of likes** in **descending** order, then by the **number of comments** in **descending order**, and finally by **photo id** in **ascending** order.

#### Required Columns

* id as photo\_id
* likes\_count
* comments\_count

#### Example

|  |  |  |
| --- | --- | --- |
| photo\_id | likes\_count | comments\_count |
| 1 | 4 | 2 |
| 58 | 4 | 1 |
| 69 | 4 | 0 |
| … | … | … |

### Photos Captured on the Tenth Day of the Month

Extract **photos** from the database only if their **capture day** is the **10th day** of the **month**.

**Summarize** their **description**:

* The **summary** must beup to **10 symbols long**. **Append** an **ellipsis** ( **"..."** )at the **end**.

Return the **capture date** in the format '**DD.MM** **HH24:MI**' as **date**.

Order the results by **capture\_date in descending order**.

#### Required Columns

* summary
* date

#### Example

|  |  |
| --- | --- |
| **summary** | date |
| ... | 10.12 15:20 |
| Quisque id... | 10.10 08:58 |
| Mauris eni... | 10.05 14:40 |
| … | … |

## Section 4: Programmability - 20 Pts

Now it's time to showcase your database skills with some dynamic scripting. Get ready to write a series of functions and procedures to demonstrate your versatility.

### Get Accounts Photos Count

Create a **user-defined function** named **udf\_accounts\_photos\_count(account\_username VARCHAR(30))** that receives an account's **username** and **returns** the **number of photos** the account has uploaded.

Submit **only** your **user-defined function** to the Judge system.

#### Example

|  |
| --- |
| **Test Query** |
| SELECT udf\_accounts\_photos\_count('ssantryd') AS photos\_count; |
| Result |
| photos\_count |
| 2 |

### Modify Accounts Job Title

Create a stored procedure **udp\_modify\_account** with the following parameters:

* address\_street
* address\_town

The procedure **udp\_modify\_account(address\_street VARCHAR(30), address\_town VARCHAR(30))** receives a **street** and a **town** as inputsand **modifies** the associated account's **job title** by appending the string **"(Remote) "** at the **beginning** of the current job title if the account **exists**.

Submit **only** your **stored procedure** to the Judge system.

#### Example

|  |
| --- |
| **Test Query** |
| CALL udp\_modify\_account('97 Valley Edge Parkway', 'Nonexistent');  SELECT a.username, a.gender, a.job\_title FROM accounts AS a  WHERE a.job\_title ILIKE '(Remote)%'; |

|  |  |  |
| --- | --- | --- |
| **Result** | | |
| **username** | **gender** | **job\_title** |
|  |  |  |

|  |
| --- |
| **Test Query** |
| CALL udp\_modify\_account('97 Valley Edge Parkway', 'Divinópolis');  SELECT a.username, a.gender, a.job\_title FROM accounts AS a  WHERE a.job\_title ILIKE '(Remote)%'; |

|  |  |  |
| --- | --- | --- |
| **Result** | | |
| **username** | **gender** | **job\_title** |
| eblagden21 | M | (Remote) Associate Professor |