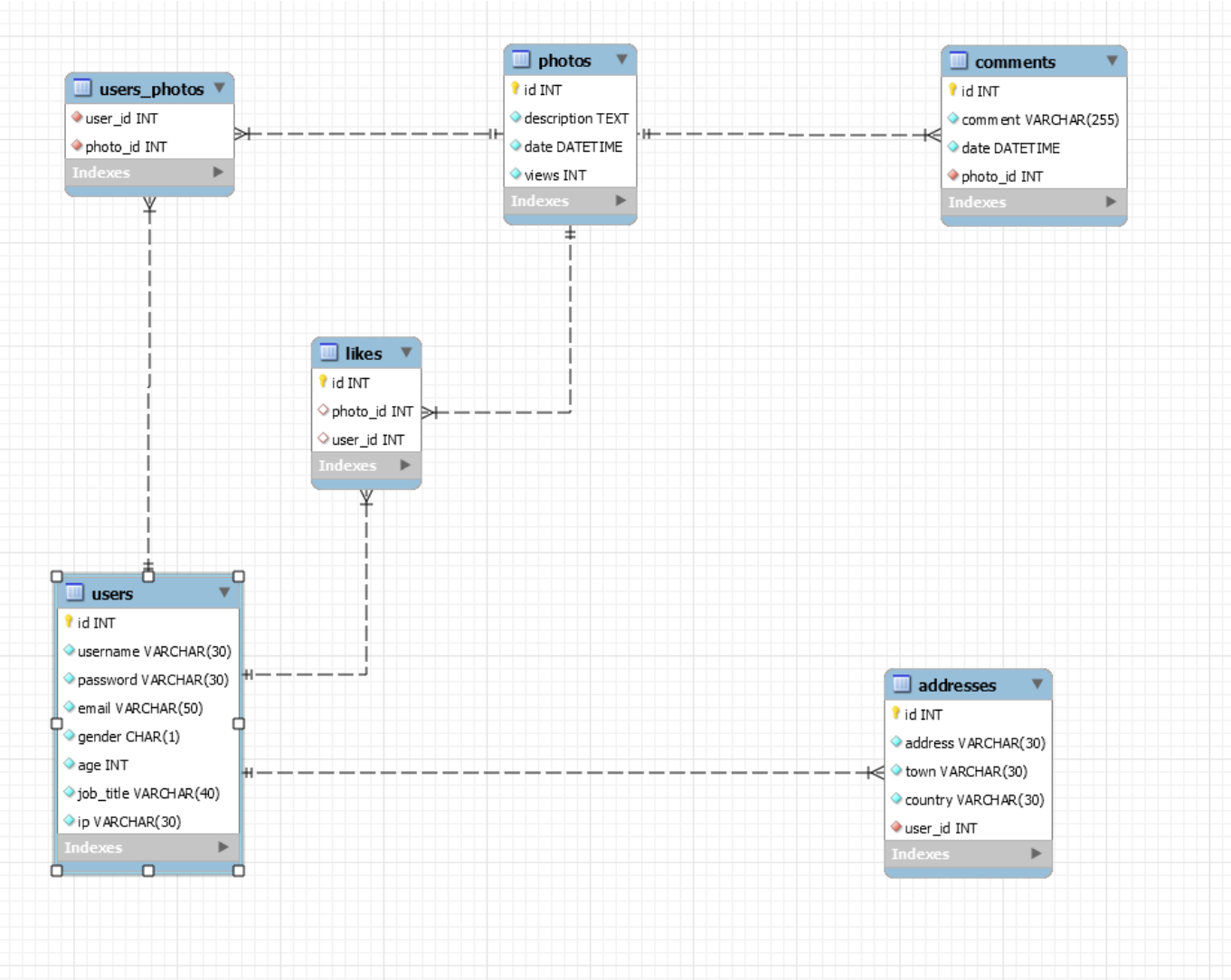
MySQL Exam

Insta Influencers

You have been selected to help the most famous Insta influencers. Thanks to your knowledge of databases, you have been selected to create the structure of a brand-new database and to fill it. Once the base is ready, you will be able to respond without any problems to any information request from the influencers based on certain criteria. As with other databases, it is most important first to become familiar with the structure you need to build, and then fill it with given data.

Section 0: Database Overview

You have been given an Entity / Relationship Diagram of the **Insta Influencers**:



The **Insta Influencers** needs to hold information about **users**, **addresses**, **photos**, **comments**, **users_photos, likes**.

Your task is to create a database called **`instd` (Insta Database)**. Then you will have to create several **tables**.

- **`users`** – contains information about the **users**.

    - Each user has an **id**, **username**, **password**, **email**, **gender**, **age**, **job_title** and **ip**.
- **addresses** – contains information about the **addresses**.
    - Each address has an id, address, town, country and user_id.
- **photos** – contains information about the **photos**.
    - Each photo has id, description, date and views.
- **comments** – contains information about the **comments**.
    - Each comment has id, comment, date and photo_id.
- **users_photos** – a **many** to **many mapping** table between the **users** and the **photos**.
    - Have composite primary key from **user_id** and **photo_id**
- **likes** – contains information about the **likes**.
    - Each like has **id, photo_id and user_id.**

Section 1: Data Definition Language (DDL) – 40 Pts

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

The instructions you'll be given will be the minimal required for you to implement the database.

01. Table Design

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

users

Column Name	Data Type	Constraints
id	**Integer,** from **1** to 2,147,483,647.	**Primary Key**
username	A **string** containing a maximum of **30 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.** **UNIQUE** values.
password	A **string** containing a maximum of **30 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.**
email	A **string** containing a maximum of **50 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.**
gender	**Exactly 1 character – M or F**	**NULL** is **NOT** permitted**.**
age	**Integer,** from **1** to 2,147,483,647.	**NULL** is **NOT** permitted**.**
job_title	A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.**
ip	A **string** containing a maximum of **30 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.**

addresses

Column Name	Data Type	Constraints
id	**Integer,** from **1** to 2,147,483,647.	**Primary Key** **AUTO_INCREMENT**

address	A **string** containing a maximum of **30 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted.
town	A **string** containing a maximum of **30 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.**
country	A **string** containing a maximum of **30 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.**
user_id	**Integer,** from **1** to 2,147,483,647.	Relationship with table **users**. **NULL** is **NOT** permitted**.**

photos

Column Name	Data Type	Constraints
id	**Integer,** from **1** to 2,147,483,647.	**Primary Key** **AUTO_INCREMENT**
description	**Very big String.**	**NULL** is **NOT** permitted.
date	The exact date and time.	**NULL** is **NOT** permitted**.**
views	**Integer,** from **1** to 2,147,483,647.	**DEFAULT value is 0.** **NULL** is **NOT** permitted**.**

comments

Column Name	Data Type	Constraints
id	**Integer,** from **1** to 2,147,483,647.	**Primary Key** **AUTO_INCREMENT**
comment	**A String** containing a maximum of **255 characters**. Unicode is **NOT** needed.	**NULL** is **NOT** permitted**.**
date	The exact date and time.	**NULL** is **NOT** permitted**.**
photo_id	**Integer,** from **1** to 2,147,483,647.	Relationship with table **photos**. **NULL** is **NOT** permitted**.**

users_photos

Column Name	Data Type	Constraints

`user_id`	**Integer**, from **1** to 2,147,483,647.	Relationship with table **`users`**. **NULL** is **NOT** permitted**.**
`photo_id`	**Integer**, from **1** to 2,147,483,647.	Relationship with table **`photos`**. **NULL** is **NOT** permitted**.**

`likes`

Column Name	Data Type	Constraints
`id`	**Integer,** from **1** to 2,147,483,647.	**Primary Key AUTO_INCREMENT**
`photo_id`	**Integer,** from **1** to 2,147,483,647.	Relationship with table **photos**.
`user_id`	**Integer,** from **1** to 2,147,483,647.	Relationship with table **users**.

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 don't have to imagine it and lose precious 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.

02. Insert

You will have to **insert** records of data into the **addresses** table, based on the **users** table.

For **`users`** with **`male gender`**, **insert data** in the **`addresses`** table with the **following values**:

- **`address`** – set it to **username** of the **user.**
- **`town`** – set it to **password** of the **user**.
- **`country`** – set it to **ip** of the **user**.
- **`user_id`** – set it to **age** of the **user**.

03. Update

Rename those **countries**, 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'**.

04. Delete

As you remember at the beginning of our work, we **inserted** and **updated** some data. Now you need to **remove** some **addresses**.

Delete all **addresses** from table **addresses,** which **id** is divisible by **3**.

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.

05. Users

Extract from the Insta Database (instd), info about all the **users**.

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
mgethingq	M	99
…	…	…

06. Extract 5 Most Commented Photos

Extract from the database, 5 most commented **photos** with their count of **comments.** Sort the results by **commentsCount, descending**, then by **id** in **ascending** order.

Required Columns

- `id`
- `date_and_time`

- `description`
- `commentsCount`

Example

id	date_and_time	description	commentsCount
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
…	…	…	…

07. Lucky Users

When the user has the **same** id as its photo, it is considered Lucky User. Extract from the database **all lucky users**.

Extract **id_username** (concat id + " " + username) and email of **all lucky users.** Order the results ascending by user **id.**

Required Columns

- `id_username`
- `email`

Example

id_username	email
12 aroccob	dpendrichb@hhs.gov
…	…

08. Count Likes and Comments

Extract from the database, **photos id** with their **likes and comments. Order** them by count of **likes descending,** then by **comments count descending** and lastly by **photo id ascending.**

Required Columns

- `photo_id`
- `likes_count`
- `comments_count`

Example

photo_id	likes_count	comments_count
1	4	2
58	4	1

69	4	0
...	...	...

09. The Photo on the Tenth Day of the Month

Extract from the database those **photos** that their upload day is **10** and **summarize** their description. **The summary must be 30 symbols long plus "..." at the end.** Order the results by **date descending order.**

Required Columns

- `summary`
- `date`

Example

summary	date
Suspendisse potenti. In eleife...	2019-12-10 15:20:14
Quisque id justo sit amet sapi...	2019-10-10 08:58:52
Mauris enim leo, rhoncus sed, ...	2019-05-10 14:40:22
...	...

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.

10. Get User's Photos Count

Create a **user defined function** with the name **`udf_users_photos_count(username VARCHAR(30))`** that receives a **username** and returns the number of photos this user has upload.

Example

Query
`SELECT udf_users_photos_count('ssantryd') AS photosCount;`

photosCount
2

11. Increase User Age

Create a stored procedure **udp_modify_user** which accepts the following parameters:

- **address**
- **town**

udp_modify_user (address VARCHAR(30), town VARCHAR(30)) that receives an **address** and **town** and increase the age of the user by **10** years **only if** the given user **exists**.

Show all needed info for this user: **username**, **email**, **gender**, **age** and **job_title**.

CALL **udp_modify_user ('97 Valley Edge Parkway', 'Divinópolis');**

Query

```
CALL udp_modify_user ('97 Valley Edge Parkway', 'Divinópolis');
SELECT u.username, u.email,u.gender,u.age,u.job_title FROM users AS u
WHERE u.username = 'eblagden21';
```

Result

username	email	gender	age	Job_title
eblagden21	eishak21@skyrock.com	M	91	Associate Professor