

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**  
*Кафедра систем штучного інтелекту*

**РОЗРАХУНКОВА РОБОТА**

**з дисципліни "ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ"**  
**на тему:**

**"ПРОЕКТУВАННЯ БАЗИ ДАНИХ  
ІНФОРМАЦІЙНОЇ СИСТЕМИ ФРІЛАНСУ"**

**Виконав студент групи КН-208**

**Гірник Ю.В.**

| <b>Оцінка</b> | <b>Балів</b> | <b>Дата</b> |
|---------------|--------------|-------------|
|               |              |             |

**Керівник**

**к.т.н, Мельникова Н.І.**

**Львів-2020**

## **Зміст**

|                                     |    |
|-------------------------------------|----|
| Вступ.....                          | 3  |
| 1. Логічна схема.....               | 4  |
| 2. Опис структури.....              | 5  |
| 3. Фізична модель.....              | 15 |
| 4. Ділова модель.....               | 18 |
| 5. Запити.....                      | 22 |
| Висновки.....                       | 26 |
| Список використаної літератури..... | 27 |

## **Вступ**

Все частіше і частіше сучасні люди працюють віддалено. Особливо це можна помітити сьогодні – в час пандемії, коли всі змушені працювати вдома. Виникає питання: а чи зручно це чи це зникне відразу після того як людство знову повернеться до звичного ритму життя?

Моя відповідь наступна – ні, не зникне. І так, це може бути зручним. Саме для такої мети і буде проектуватись і використовуватись база даних описана тут – сервісу фрілансингу, вільних людей без жодних зобов'язань та графіків.

Безліч професій підходить для такого зручного підходу: від звичайного журналіста та художника і аж до дизайнерів та юристів.

Крім того, цей підхід є також зручним і для всіх нас, людей з потребами і бажаннями. Всього в пару кліків Ви можете замовити собі дизайн кухні або знайти онлайн-репетитора.

Завдяки всьому цьому дана тема є особливо актуальна сьогодні.

## Логічна схема БД проекту

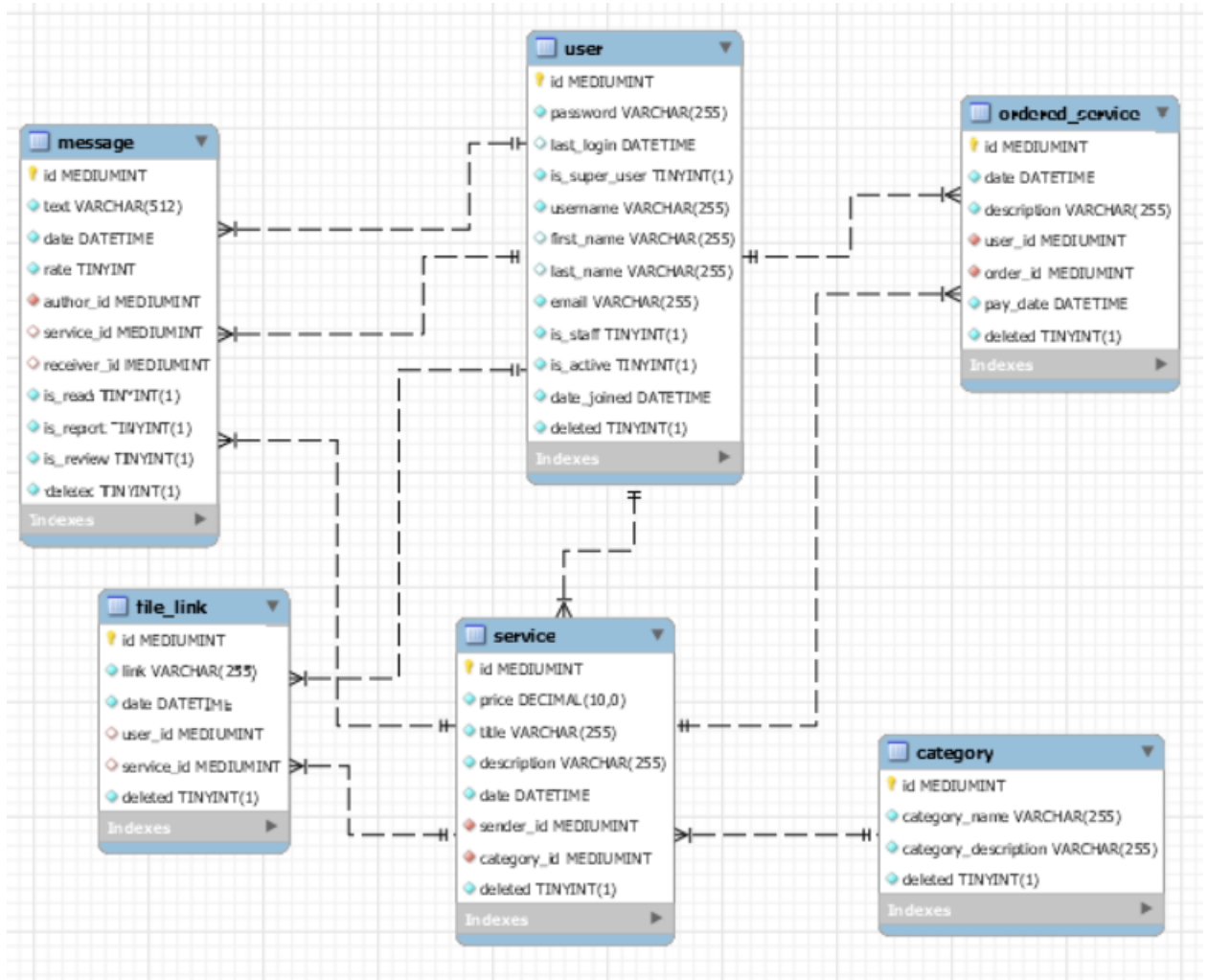


Рис.1 Логічна схема БД

## Опис структури БД

База даних складається з 6 таблиць:

- 1) user,
- 2) category,
- 3) service,
- 4) ordered service,
- 5) message
- 6) file link

Давайте в такому ж порядку розглянемо кожну з них:

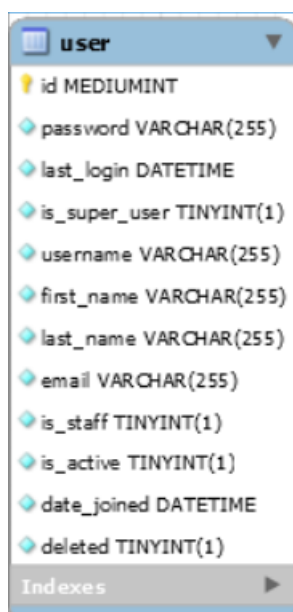


Рис.2 таблиця User

Таблиця user – для зберігання даних про користувачів.

1. “id”(unsigned medium int) – унікальний ідентифікатор користувача, первинний ключ. Властивості: Primary key, autoincrement
2. “password”(varchar(255)) – пароль користувача для входу в систему. Властивості: not null.
3. “last-login”( datetime) – дата останнього відвідування користувача.
4. “is\_super\_user”(boolean) – поле для позначення адміністраторів – користувачів з розширеними можливостями. Властивості: not null
5. “username”(varchar(255)) – унікальний псевдонім користувача. Властивості: unique, not null.

6. "first\_name"(varchar(255)) – ім'я користувача, при початковій реєстрації є необов'язковим полем.
7. "last\_name"(varchar(255)) – прізвище користувача, при початковій реєстрації також є необов'язковим полем.
8. "email"(varchar(255)) – електронна пошта користувача. Властивості: unique, not null.
9. "is\_staff"(Boolean) – поле для позначення працівників служби підтримки сервісу. Властивості: not null.
10. "is\_active"(Boolean) – поле для позначення активації користувачем свого аккаунту через пошту. Властивості: not null.
11. "date\_joined"(datetime) – дата реєстрації користувача. Властивості: not null.
12. "deleted"(Boolean) – чи видалений аккаунт користувача. Властивості: not null, default false.

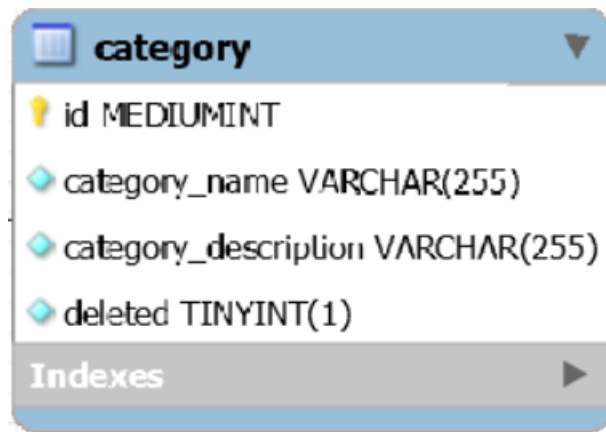


Рис.3 таблиця Category

Таблиця category – для зберігання даних про категорії.

1. “id”(unsigned medium int) – унікальний ідентифікатор категорії, первинний ключ. Властивості: Primary key, autoincrement
2. “category\_name”(varchar(255)) – назва категорії. Властивості: unique, not null.
3. “category\_description”(varchar(255)) – опис категорії. Властивості: not null.
4. “deleted”(Boolean) – чи видалений ця категорія. Властивості: not null, default false.

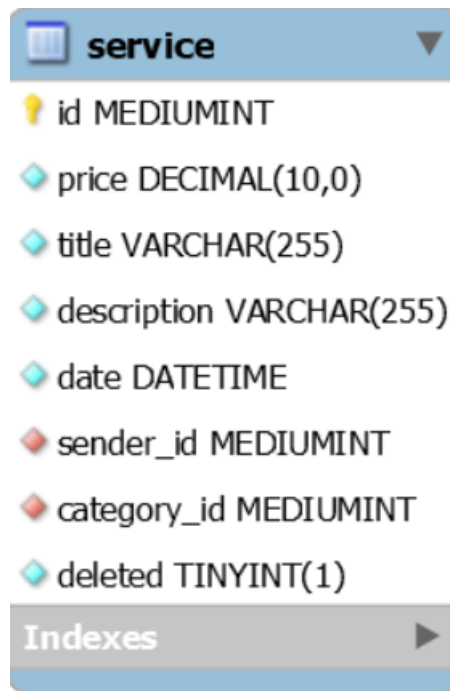


Рис.4 таблиця Service

Таблиця service – для зберігання даних про створенні сервіси.

1. “id”(unsigned medium int) – унікальний ідентифікатор сервісу, первинний ключ. Властивості: Primary key, autoincrement
2. “price”(decimal) – ціна сервісу. Властивості: not null.
3. “title”(varchar(255)) – заголовок сервісу. Властивості: not null
4. “description”(varchar(255)) –детальний опис сервісу. Властивості: not null
5. “date”(datetime) – дата додання сервісу. Властивості: not null
6. “sender\_id”(unsigned medium int) – зовнішній ключ, що містить посилання на користувача, який створив цей сервіс. Властивості: not null
7. “category\_id”(unsigned medium int) – зовнішній ключ, що містить посилання на категорію, до якої належить сервіс. Властивості: not null
8. “deleted”(Boolean) – чи видалений цей сервіс. Властивості: not null, default false.



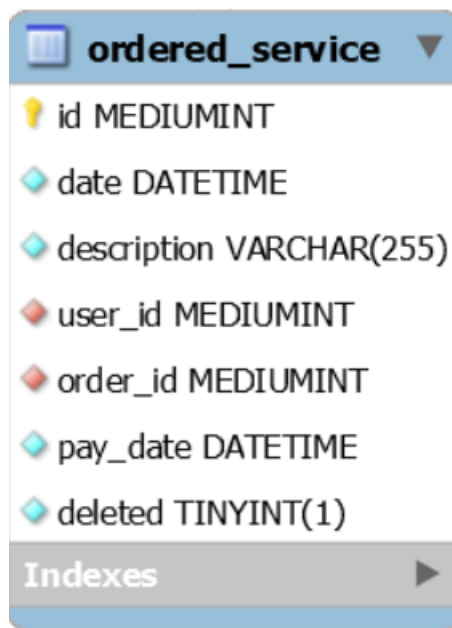


Рис.5 таблиця Ordered service

Таблиця ordered service – для зберігання даних про оформленні сервіси.

1. “id”(unsigned medium int) – унікальний ідентифікатор оформленого сервісу, первинний ключ. Властивості: Primary key, autoincrement.
2. “date”(datetime) – дата оформлення сервісу. Властивості: not null.
3. “user\_id”(unsigned medium int) – зовнішній ключ, що містить посилання на користувача, який оформив сервіс. Властивості: not null.
4. “order\_id”(unsigned medium int) - зовнішній ключ, що містить посилання на сервіс, який оформив користувач. Властивості: not null.
5. “pay\_date”(datetime) – дата оплати даного оформленого сервісу.
6. “deleted”(Boolean) – чи видалений цей оформлений сервіс. Властивості: not null, default false.

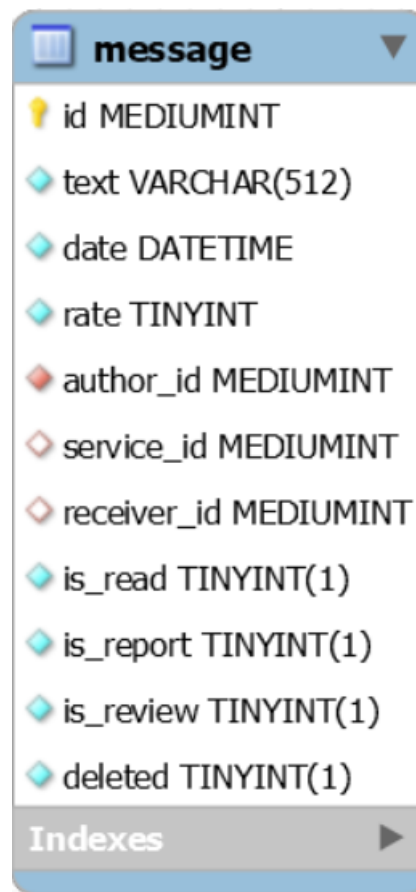


Рис.6 таблиця Message

Таблиця message – для зберігання даних про повідомлення між користувачами, відгуки на сервіси або скарги на користувачів.

1. “id”(unsigned medium int) – унікальний ідентифікатор повідомлення, первинний ключ. Властивості: Primary key, autoincrement.
2. “text”(varchar(512)) – зміст повідомлення, відгука або скарги. Властивості: not null.
3. “rate”(tinyint) – оцінка виконаного сервісу, якщо це є відгук.
4. “author\_id”(unsigned medium int) – зовнішній ключ, що містить посилання на користувача – автора. Властивості: not null
5. “service\_id”(unsigned medium int) - зовнішній ключ, що містить посилання на сервіс, на яке написано відгук.
6. “receiver\_id”(unsigned medium int) – зовнішній ключ, що містить посилання на користувача – отримувача повідомлення або скарги.

7. "is\_read"(Boolean) – чи прочитане повідомлення отримувачем/розглянута скарга адміністратором. Властивості: default false.
8. "is\_report"(Boolean) –поле для визначення чи це є скарга. Властивості: default false.
9. "is\_review"(Boolean) –поле для визначення чи це є відгук. Властивості: default false.
10. "deleted"(Boolean) – чи видалене це повідомлення/відгук/скаргу. Властивості: not null, default false.

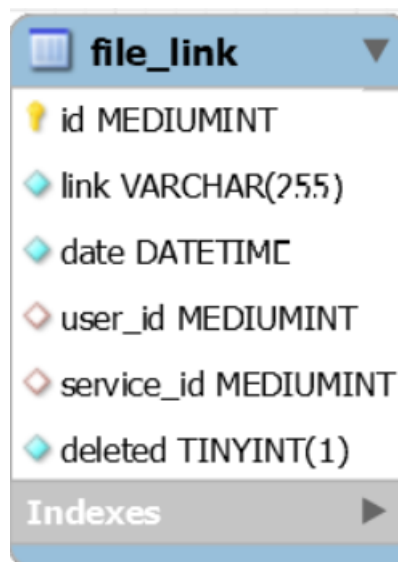


Рис.7 таблиця File link

Таблиця file link – для зберігання медіа-файлів сервісу або окремого користувача.

1. “id”(unsigned medium int) – унікальний ідентифікатор медіа-файлів, первинний ключ. Властивості: Primary key, autoincrement.
2. “link”(varchar(255)) – поле, яке містить шлях до файлу. Властивості: not null.
3. “date”(datetime) – дата додання медіа-файлу. Властивості: not null.
4. “user\_id”(unsigned medium int) - зовнішній ключ, що містить посилання на користувача, який завантажив файл.
5. “service\_id”(unsigned medium int) – зовнішній ключ, що містить посилання на сервіс, для якого завантажений файл.
6. “deleted”(Boolean) – чи видалений файл. Властивості: not null, default false.

## Нормалізація бази даних

Перевіримо нашу базу даних на нормалізацію для уникнення надмірності даних та уникнення аномалій

Перша нормальна форма:

- 1) Будь-яке поле повинно бути неподільним (властивість атомарності полів).
- 2) Не повинно бути полів, в яких позначені різні види одного і того ж, наприклад, товарів.

Отже, усі відношення, що подані базовими таблицями, відповідають вимогам першої нормальної форми.

Друга нормальна форма:

- 1) База даних відповідає першій нормальній формі
- 2) Будь-яке поле повинно залежати тільки від ключа (ключ – це поле або комбінація полів, однозначно визначає запис).
- 3) Не повинно бути полів, які можуть бути знайдені з допомогою інших

Можна побачити, що категорії оформлених сервісів в окрему таблицю для уникнення повторів. Отже усі відношення, що подані базовими таблицями відповідають другій нормальній формі.

Третя нормальна форма:

- 1) База даних відповідає другій нормальній формі
- 2) Відсутні транзитивні залежності між неключовими атрибутами

Оскільки у нас відсутні поля, які транзитивно залежать від інших, неключових полів, то можна зробити висновок, що база даних знаходиться в третій нормальній формі.

Нормальна форма Бойса-Кодда:

3НФ не співпадає з НФБК лише тоді, коли одночасно виконуються наступні умови:

- 1) Відношення має два або більшу кількість потенційних ключів
- 2) Ці потенційні ключі складені, тобто містять більше ніж один атрибут
- 3) Ці потенційні ключі перекриваються, тобто мають по крайній мірі один спільний атрибут

У таблиці категорії присутні два потенційні ключі: `id`, `category_name`. Проте усі ці значення унікальні, тобто 3 пункт не виконується. Отже НФБК виконується.

Таким чином, база даних відповідає нормальній формі Бойса-Кодда.

## Фізична модель БД

**CREATE DATABASE** wanderlance;

**USE** wanderlance;

**CREATE TABLE** user (

id **MEDIUMINT UNSIGNED NOT NULL AUTO\_INCREMENT,**

password **VARCHAR(255) NOT NULL,**

last\_login **DATETIME,**

is\_super\_user **BOOLEAN NOT NULL,**

username **VARCHAR(255) UNIQUE NOT NULL,**

first\_name **VARCHAR(255),**

last\_name **VARCHAR(255),**

email **VARCHAR(255) NOT NULL,**

is\_staff **BOOLEAN NOT NULL,**

is\_active **BOOLEAN NOT NULL,**

date\_joined **DATETIME NOT NULL,**

deleted **BOOLEAN NOT NULL DEFAULT FALSE,**

**PRIMARY KEY** (id)

);

**CREATE TABLE** service (

id **MEDIUMINT UNSIGNED NOT NULL AUTO\_INCREMENT,**

price **DECIMAL NOT NULL,**

title **VARCHAR(255) NOT NULL,**

description **VARCHAR(255) NOT NULL,**

date **DATETIME NOT NULL,**

sender\_id **MEDIUMINT UNSIGNED NOT NULL,**

category\_id **MEDIUMINT UNSIGNED NOT NULL,**

deleted **BOOLEAN NOT NULL DEFAULT FALSE,**

**PRIMARY KEY** (id)

);

**CREATE TABLE** ordered\_service (

id **MEDIUMINT UNSIGNED NOT NULL AUTO\_INCREMENT,**  
date **DATETIME NOT NULL,**  
description **VARCHAR(255) NOT NULL,**  
user\_id **MEDIUMINT UNSIGNED NOT NULL,**  
order\_id **MEDIUMINT UNSIGNED NOT NULL,**  
pay\_date **DATETIME,**  
deleted **BOOLEAN NOT NULL DEFAULT FALSE,**  
**PRIMARY KEY (id)**  
);

**CREATE TABLE** message (  
id **MEDIUMINT UNSIGNED NOT NULL AUTO\_INCREMENT,**  
text **VARCHAR(512) NOT NULL,**  
date **DATETIME NOT NULL,**  
rate **TINYINT UNSIGNED,**  
author\_id **MEDIUMINT UNSIGNED NOT NULL,**  
service\_id **MEDIUMINT UNSIGNED,**  
receiver\_id **MEDIUMINT UNSIGNED,**  
is\_read **BOOLEAN DEFAULT FALSE,**  
is\_report **BOOLEAN NOT NULL DEFAULT FALSE,**  
is\_review **BOOLEAN NOT NULL DEFAULT FALSE,**  
deleted **BOOLEAN NOT NULL DEFAULT FALSE,**  
**PRIMARY KEY (id)**  
);

**CREATE TABLE** category (  
id **MEDIUMINT UNSIGNED NOT NULL AUTO\_INCREMENT,**  
category\_name **VARCHAR(255) NOT NULL UNIQUE,**  
category\_description **VARCHAR(255) NOT NULL,**  
deleted **BOOLEAN NOT NULL DEFAULT FALSE,**  
**PRIMARY KEY (id)**  
);



```
CREATE TABLE file_link (  
id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
link VARCHAR(255) NOT NULL,  
date DATETIME NOT NULL,  
user_id MEDIUMINT UNSIGNED,  
service_id MEDIUMINT UNSIGNED,  
deleted BOOLEAN NOT NULL DEFAULT FALSE,  
PRIMARY KEY (id)  
);
```

```
ALTER TABLE service ADD CONSTRAINT FOREIGN KEY (sender_id) REFERENCES  
user (id) ON DELETE CASCADE;
```

```
ALTER TABLE service ADD CONSTRAINT FOREIGN KEY (category_id)  
REFERENCES category (id) ON DELETE CASCADE;
```

```
ALTER TABLE ordered_service ADD CONSTRAINT FOREIGN KEY (user_id)  
REFERENCES user (id) ON DELETE CASCADE;
```

```
ALTER TABLE ordered_service ADD CONSTRAINT FOREIGN KEY (order_id)  
REFERENCES service (id) ON DELETE CASCADE;
```

```
ALTER TABLE message ADD CONSTRAINT FOREIGN KEY (author_id)  
REFERENCES user (id) ON DELETE CASCADE;
```

```
ALTER TABLE message ADD CONSTRAINT FOREIGN KEY (service_id)  
REFERENCES service (id) ON DELETE CASCADE;
```

```
ALTER TABLE message ADD CONSTRAINT FOREIGN KEY (receiver_id)  
REFERENCES user (id) ON DELETE CASCADE;
```

```
ALTER TABLE file_link ADD CONSTRAINT FOREIGN KEY (user_id) REFERENCES  
user (id) ON DELETE CASCADE;
```

```
ALTER TABLE `file_link` ADD CONSTRAINT FOREIGN KEY (`service_id`)  
REFERENCES `service`(`id`) ON DELETE CASCADE;
```

## Ділова модель

### Ділова модель організації «Сайт фрілансу»

| Класи<br>Функції          | Користувач | Сервіс | Оф. сервіс | Повідомлення | Категорія | Файл |
|---------------------------|------------|--------|------------|--------------|-----------|------|
| Створення сервісу         | *          | *      |            |              | *         |      |
| Обмін повідомленнями      | *          |        |            | *            |           |      |
| Обмін файлами             | *          |        |            |              |           | *    |
| Сортування за категоріями |            | *      |            |              | *         |      |
| Замовлення сервісу        | *          | *      | *          |              |           |      |
| Виконання замовлення      | *          | *      | *          | *            |           | *    |

Для забезпечення роботоздатності нам потрібні наступні функції:

- 1) Основа – створення сервісу.
- 2) Реалізація чату через обмін повідомленнями.
- 3) Можливість прикріпити до певного юзера чи сервісу медіа-файл.
- 4) Вибірку потрібних сервісів для користувачами за різними параметрами: датою, ціною тощо.
- 5) Можливість оформити сервіс.

**Тригери**, створені для реалізація функціоналу в базі даних:

1-5) У всіх таблицях, окрім категорії, є поле date – яке позначає дату додання запису, автоматичне додання дати додання. Створимо тригери для цієї функції для цих 5-ти таблиць:

```
CREATE TRIGGER date_file BEFORE INSERT  
ON file_link FOR EACH ROW  
SET NEW.date = NOW();
```

```
CREATE TRIGGER date_user BEFORE INSERT  
ON user FOR EACH ROW  
SET NEW.date_joined = NOW();
```

```
CREATE TRIGGER date_service BEFORE INSERT  
ON service FOR EACH ROW  
SET NEW.date = NOW();
```

```
CREATE TRIGGER date_ordered_service BEFORE INSERT  
ON ordered_service FOR EACH ROW  
SET NEW.date = NOW();
```

```
CREATE TRIGGER date_message BEFORE INSERT  
ON message FOR EACH ROW  
SET NEW.date = NOW();
```

6-7) Тригер для “м’якого видалення” сервісу з усіма відгуками та повідомленнями, користувача з сервісами, медіа-файлами та оформленими сервісами:

```
DELIMITER %%;  
CREATE TRIGGER delete_service AFTER UPDATE  
ON service FOR EACH ROW  
BEGIN
```

```

IF (OLD.deleted!=NEW.deleted and NEW.deleted=1) THEN
BEGIN
    UPDATE message SET message.deleted = 1 WHERE
message.service_id=OLD.id;
    UPDATE ordered_service SET ordered_service.deleted=1 WHERE
ordered_service.order_id=OLD.id;
END;
END IF;
END %%;

DELIMITER %%;

CREATE TRIGGER delete_user AFTER UPDATE
ON user FOR EACH ROW
BEGIN
    IF (OLD.deleted!=NEW.deleted and NEW.deleted=1) THEN
BEGIN
        UPDATE file_link SET file_link.deleted=1 WHERE file_link.user_id=OLD.id;
        UPDATE service SET service.deleted=1 WHERE service.sender_id=OLD.id;
        UPDATE ordered_service SET ordered_service.deleted=1 WHERE
ordered_service.suser_id=OLD.id;
    END; END IF;
END %%;

```

8) Встановлення стандартного фото при відсутності іншого:

```

CREATE TRIGGER new_image BEFORE INSERT
ON file_link FOR EACH ROW
IF (ISNULL(new.link)) THEN
    SET new.link = 'default_avatar.jpg';
END IF;

CREATE TRIGGER new_image AFTER UPDATE
ON file_link FOR EACH ROW
IF (ISNULL(new.link)) THEN
    SET new.link = 'default_avatar.jpg';

```

**END IF;**

9) Створення запису медіа-файлу при створенні сервісу:

**CREATE TRIGGER new\_service AFTER INSERT**

**ON service FOR EACH ROW**

**INSERT INTO file\_link(service\_id) VALUE**

**(new.id);**

10) Створення запису медіа-файлу при створенні користувача:

**CREATE TRIGGER new\_user AFTER INSERT**

**ON user FOR EACH ROW**

**INSERT INTO file\_link(user\_id) VALUE**

**(new.id);**

## Запити

1) Додання нового користувача:

```
INSERT INTO user(password, username, first_name, last_name, email, is_staff, is_active, is_super_user) VALUE
```

```
("asdasf123", "Kokos", "Irop", "Гора", "gora22@gmail.com", 0, 1, 0);
```

```
SHOW * FROM user;
```

| id | password     | last_login          | is_super_user | username     | first_name | last_name | email               | is_staff | is_active | date_joined         | deleted |
|----|--------------|---------------------|---------------|--------------|------------|-----------|---------------------|----------|-----------|---------------------|---------|
| 1  | asdasf123    | 2020-05-21 08:10:38 | 0             | Kokos        | Irop       | Гора      | gora22@gmail.com    | 0        | 1         | 2020-03-12 16:08:11 | 0       |
| 2  | fczvxx12     | 2020-05-22 09:11:45 | 0             | Kek          | Василь     | Ткаченко  | vtkach@gmail.com    | 0        | 1         | 2020-02-09 03:01:12 | 0       |
| 3  | 2dsASDxzcv   | 2020-05-23 10:12:21 | 0             | korol        | Іван       | Кофейний  | kkokos@gmail.com    | 0        | 1         | 2020-03-12 19:08:53 | 0       |
| 4  | asdqw12334   | 2020-06-24 11:13:11 | 0             | wanderlancer | Микола     | Хоровий   | asdr@gmail.com      | 0        | 1         | 2020-02-12 05:31:52 | 0       |
| 5  | iungdrey     | 2020-06-25 12:14:52 | 0             | lolik        | Макс       | Шуков     | Max@gmail.com       | 0        | 1         | 2020-05-05 10:33:51 | 0       |
| 6  | dasdzxcassAA | 2020-06-23 13:15:13 | 0             | Dai5         | Іванка     | Хома      | Ivanka322@gmail.com | 1        | 1         | 2020-05-12 19:13:31 | 0       |
| 11 | asdasf123    | 2020-05-24 13:58:33 | 0             | Kokos        | Irop       | Гора      | gora22@gmail.com    | 0        | 1         | 2020-05-24 13:58:33 | 0       |

| id | link               | date                | user_id | service_id | deleted |
|----|--------------------|---------------------|---------|------------|---------|
| 2  | default_avatar.jpg | 2020-05-20 10:00:00 | NULL    | 2          | 0       |
| 3  | sdsad.jpg          | 2020-05-20 10:00:00 | NULL    | 3          | 0       |
| 4  | hqqq.jpg           | 2020-05-20 10:00:00 | NULL    | 4          | 0       |
| 5  | vwwwk.jpg          | 2020-05-20 10:00:00 | NULL    | 5          | 0       |
| 6  | sdd.jfif           | 2020-05-20 10:00:00 | NULL    | 6          | 0       |
| 7  | osssss.webp        | 2020-05-20 10:00:00 | 1       | NULL       | 0       |
| 8  | default_avatar.jpg | 2020-05-20 10:00:00 | 2       | NULL       | 0       |
| 9  | qwed.jpg           | 2020-05-24 13:04:04 | NULL    | 1          | 0       |
| 12 | default_avatar.jpg | 2020-05-24 13:58:33 | 11      | NULL       | 0       |

2) Видалення користувача:

```
UPDATE user SET user.deleted=1 WHERE id=11;
```

```
SELECT * FROM file_link;
```

| id | link               | date                | user_id | service_id | deleted |
|----|--------------------|---------------------|---------|------------|---------|
| 2  | default_avatar.jpg | 2020-05-20 10:00:00 | NULL    | 2          | 0       |
| 3  | sdsad.jpg          | 2020-05-20 10:00:00 | NULL    | 3          | 0       |
| 4  | hqqq.jpg           | 2020-05-20 10:00:00 | NULL    | 4          | 0       |
| 5  | vwwwk.jpg          | 2020-05-20 10:00:00 | NULL    | 5          | 0       |
| 6  | sdd.jfif           | 2020-05-20 10:00:00 | NULL    | 6          | 0       |
| 7  | osssss.webp        | 2020-05-20 10:00:00 | 1       | NULL       | 0       |
| 8  | default_avatar.jpg | 2020-05-20 10:00:00 | 2       | NULL       | 0       |
| 9  | qwed.jpg           | 2020-05-24 13:04:04 | NULL    | 1          | 0       |
| 12 | default_avatar.jpg | 2020-05-24 13:58:33 | 11      | NULL       | 1       |

Як бачимо, тригери успішно спрацювали.

- 3) Показ кількості замовлень для всіх сервісів конкретного користувача посортованих по даті:

```
SELECT A.id, A.price, A.title, COUNT(F.id) AS 'Замовлень'  
FROM service AS A  
    INNER JOIN ordered_service AS F ON F.order_id=A.id  
WHERE A.sender_id=3  
GROUP BY A.id  
ORDER BY A.date;
```

| id | price | title       | Замовлень |
|----|-------|-------------|-----------|
| 3  | 300   | Лого мейкер | 2         |
| 4  | 400   | Флейтист    | 1         |

- 4) Повернення чату між користувача посортованих по даті:

```
SELECT A.text, B.username AS "Відправник", C.username AS "Отримувач " FROM  
message AS A  
    INNER JOIN user AS B ON A.author_id=B.id  
    INNER JOIN user AS C ON A.receiver_id=C.id  
WHERE ((AUTHOR_ID=1 AND RECEIVER_ID=3) OR (AUTHOR_ID=3 AND  
RECEIVER_ID=1)) AND IS_REPORT=0  
ORDER BY A.date;
```

| text              | Відправник | Отримувач |
|-------------------|------------|-----------|
| hello             | korol      | Kokos     |
| i want some photo | korol      | Kokos     |
| ok                | Kokos      | korol     |

- 5) Повернення даних про користувача, дату оформлення та ціну всіх активних замовлень конкретного фрілансера, тобто таких сервісів, які ще неоплачені(поле pay\_date is Null)

```

SELECT A.first_name, A.last_name, B.title, B.price, C.date
FROM service AS B
INNER JOIN ordered_service AS C ON C.order_id=B.id
INNER JOIN user AS A ON A.id=C.user_id
WHERE ISNULL(C.pay_date)=True AND B.sender_id=3;

```

| first_name | last_name | title       | price | date                |
|------------|-----------|-------------|-------|---------------------|
| Irop       | Гора      | Лого мейкер | 300   | 2020-05-14 10:00:01 |
| Irop       | Гора      | Флейтист    | 400   | 2020-05-14 10:00:01 |

- б) Повернення даних про скарги на конкретного користувача: адресата, адресанта скарги та самого тексту скарги

```

SELECT B.first_name AS 'Receiver first name',
C.last_name AS 'Receiver last name',
C.first_name AS 'Author first name',
C.last_name AS 'Author last name',
A.text AS 'Report text', A.date
FROM message AS A
INNER JOIN user AS B ON A.receiver_id = B.id
INNER JOIN user AS C ON A.author_id = C.id
WHERE B.id=2 AND A.is_report=1
ORDER BY A.date;

```

| Receiver first name | Receiver last name | Author first name | Author last name | Report text                   | date                |
|---------------------|--------------------|-------------------|------------------|-------------------------------|---------------------|
| Василь              | Хоровий            | Микола            | Хоровий          | help me                       | 2020-05-23 10:10:00 |
| Василь              | Кофейний           | Іван              | Кофейний         | Awfull service, huge deceiver | 2020-05-23 10:10:00 |



7) Повернення сервісів з власниками, посортованих по даті:

**SELECT** title, price, a.username, a.first\_name, a.last\_name

**FROM** service

**INNER JOIN** user **AS** a **ON** service.sender\_id=a.id

**ORDER BY** date;

| title       | price | username     | first_name | last_name |
|-------------|-------|--------------|------------|-----------|
| Письменник  | 100   | Kokos        | Igor       | Гора      |
| Художник    | 200   | Kek          | Василь     | Ткаченко  |
| Лого мейкер | 300   | korol        | Іван       | Кофейний  |
| Флейтист    | 400   | korol        | Іван       | Кофейний  |
| Геймер      | 500   | korol        | Іван       | Кофейний  |
| Вчитель     | 600   | wanderlancer | Микола     | Хоровий   |

## **Висновки**

При виконанні цієї розрахункової роботи була спроектована та реалізована реляційна база даних в середовищі MySQL Workbench. Це дозволило закріпити теоретичні знання, набуті при вивченні цього курсу.

Вибрана область “Фрілансу” була детально вивчена та досліджена. Наприклад, було пророблена робота по побудові обмежень, нормалізації бази даних.

Були реалізовані всі основні тригери, які потрібні для забезпечення цілісності бази даних та підтримкою стабільності системи. Також були виконані запити, що потрібні для використання бази даних у реальних умовах.

У результаті я отримав БД, яка може використовуватись в реальному житті, наприклад для веб-сервісу з надання послуг для фрілансерів.

## Список літератури

- 1) <https://www.coursera.org/learn/sql-for-data-science>
- 2) <http://vns.lpnu.ua/course/view.php?id=7897>
- 3) <https://www.w3schools.com/sql/>
- 4) <https://dev.mysql.com/doc/refman/8.0/en/>
- 5) <https://dev.mysql.com/doc/workbench/en/>
- 6) <https://www.guru99.com/database-normalization.html>
- 7) [http://www.mstu.edu.ru/study/materials/zelenkov/ch\\_2\\_4.html](http://www.mstu.edu.ru/study/materials/zelenkov/ch_2_4.html)
- 8) Пасічник В.В., Резніченко В.А. Організація баз даних та знань - К.: Видавнича група BHV, 2006. — 384 с.
- 9) Петкович Душан. Microsoft SQL Server 2012. Руководство для начинающих. СПб.: БХВ-Петербург, 2013. — 816 с.
- 10) Alan Beaulieu Learning SQL: Sebastopol, 2009 - 337