

**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**

**Левицький Б.Р**

<b>Балів</b>	<b>Дата</b>

**Викладач**

**Мельникова Н.І.**

**Львів - 2020**

## Зміст

Вступ.....	3
1. Логічна схема.....	4
2. Опис структури.....	5
3. Фізична модель.....	12
4. Ділова модель.....	19
5. Запити.....	20
Висновки.....	26
Список використаної літератури.....	27

## Вступ

21 століття. Інтернет набув всесвітнього поширення та надає неймовірні можливості: від перегляду розважального контенту до працевлаштування на дому. Мабуть кожен хоча б раз задумувався про те, як чудово було б працювати на дому. Але як це можливо? Відповідь проста – фріланс. Що це таке? Це вільна віддалена робота без юридичного оформлення зобов'язань сторін – найману працю, якщо говорити простіше. Усе просто, ви отримуєте певне завдання від замовника та виконуєте його до певного терміну. Після успішного завершення праці, передаєте результат цієї праці замовнику, а він оплачує вам за це.

Які ж переваги такої організації праці?

1. Свобода вибору робочого завдання.
2. Вільний графік
3. Ви працюєте вдома
4. Відповідальність лише за власні дії
5. Можливість міжнародної співпраці

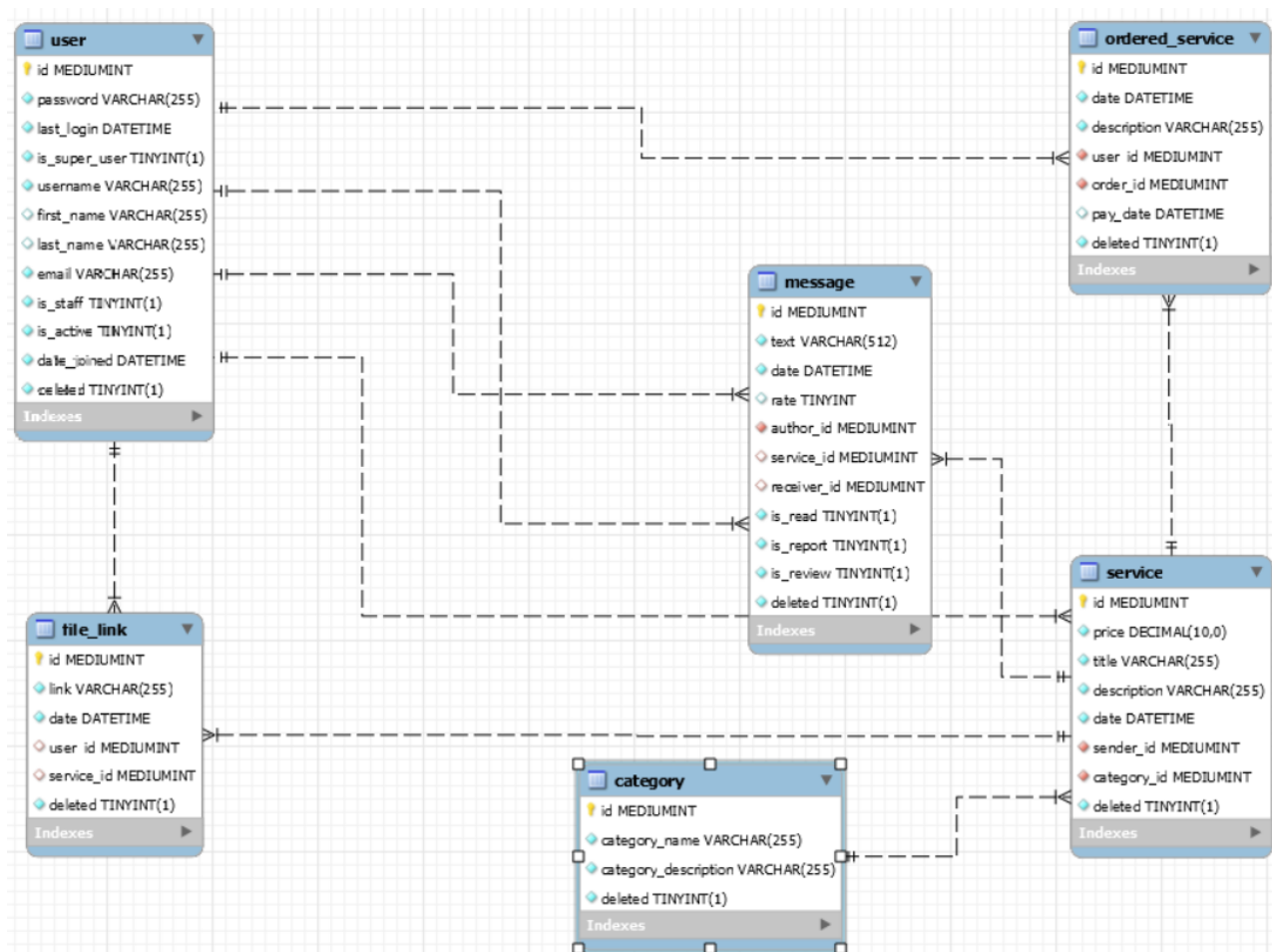
І так далі. Насправді переваг є дуже багато, і перерахувати усі ми не будемо.

Фріланс набирає стрімку популярність як в США так і в країнах Європи, в тому числі в Україні. Особливо поширеним є в сферах дизайну, програмування, перекладу, копірайтингу.

Як же відбувається пошук роботи? Усе просто, існують площадки для фрілансу, на яких люди можуть як запропонувати певні свої послуги, так і купити те, що їм потрібно.

Відповідно, враховуючи стрімку популярність фрілансу, було прийнято рішення розробити систему, яка б дозволила користувачам легко та зручно розміщати свої послуги та шукати потрібні. Оскільки зараз пандемія COVID-19, багато людей просто втратили свою роботу. Фріланс це чудовий вибір будь-якої людини, яка сидить на карантині. І свою актуальність цей спосіб праці буде зберігати.

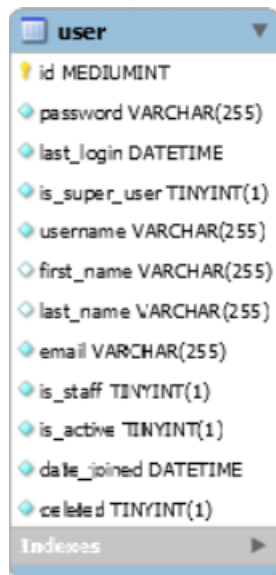
## Логічна схема БД проекту



## Опис структури БД

База даних складається з 6 таблиць: “user”, “service”, “ordered\_service”, “category”, “message” та “file\_link”.

User

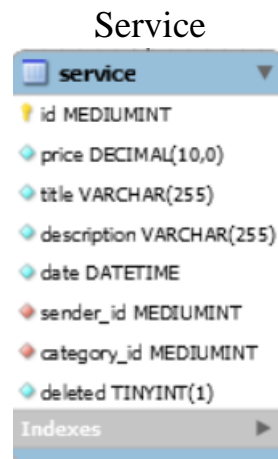


user	
id	MEDIUMINT
password	VARCHAR(255)
last_login	DATETIME
is_super_user	TINYINT(1)
username	VARCHAR(255)
first_name	VARCHAR(255)
last_name	VARCHAR(255)
email	VARCHAR(255)
is_staff	TINYINT(1)
is_active	TINYINT(1)
date_joined	DATETIME
deleted	TINYINT(1)

Таблиця, в якій будуть зберігатися дані про користувачів.

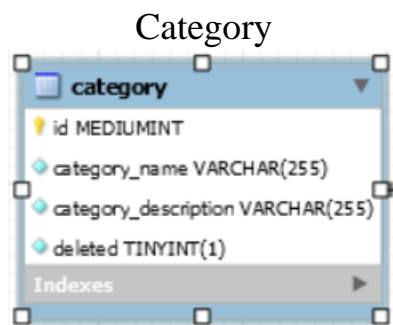
- 1) “id” – унікальний ідентифікатор користувача. Тип – unsigned medium int , обмеження – Primary key, autoincrement
- 2) “password” – пароль користувача. Тип – varchar, обмеження – not null, максимум 255 символів
- 3) “last-login” – дата останнього відвідування користувача. Тип – datetime, not null
- 4) “is\_super\_user” – службове поле для позначення адмінів – користувачів з розширеними можливостями. Тип – boolean, обмеження – not null, стандартне значення 0
- 5) “username” – унікальний псевдонім користувача. Тип varchar, обмеження – 255 символів, унікальність, not null.
- 6) “first\_name” – ім’я користувача. Тип – varchar, обмеження – 255 символів
- 7) “last\_name” – прізвище користувача. Тип – varchar, обмеження - 255
- 8) “email” – електронна пошта користувача. Тип – varchar, обмеження – 255 символів, унікальність, not null
- 9) “is\_staff” – службове поле для позначення обслуговуючого персоналу, наприклад служби підтримки. Тип boolean, обмеження – not null, стандартне значення 0
- 10) “is\_active” – службове поле для позначення чи активував користувач свій аккаунт. Тип Boolean, обмеження – not null

- 11) “date\_joined” – дата реєстрації користувача. Тип datetime, обмеження – not null
- 12) “deleted” – позначення чи видалений аккаунт користувача. Тип Boolean, обмеження - not null, стандартне значення - 0



Таблиця, в якій будуть зберігатися дані про створенні замовлення.

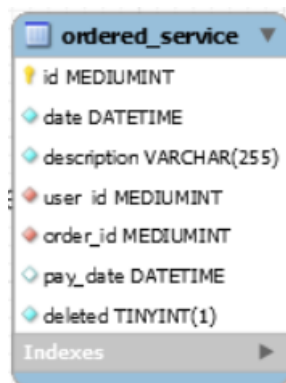
- 1) “id” – унікальний ідентифікатор замовлення. Тип – unsigned medium int , обмеження – primary key, autoincrement
- 2) “price” – вартість даного сервісу. Тип – decimal, обмеження – not null
- 3) “title” – заголовок сервісу. Тип – varchar, обмеження – максимум 255 символів, not null
- 4) “description” – опис сервісу, який заповнює його автор. Тип – varchar, обмеження – максимум 255 символів, not null
- 5) “date” – дата створення сервісу, заповнюється автоматично тригером. Тип datetime, обмеження – not null
- 6) “sender\_id” – зовнішній ключ, містить id користувача, який створив це замовлення. Тип unsigned medium int, обмеження – not null
- 7) “category\_id” – зовнішній ключ, містить id категорії, до якої належить замовлення. Тип unsigned medium int, обмеження – not null
- 8) “deleted” – позначення чи видалено замовлення. Тип Boolean, обмеження - not null, стандартне значення - 0



Таблиця, в якій будуть зберігатися дані про категорії замовлень.

- 1) “id” – унікальний ідентифікатор категорії. Тип – unsigned medium int , обмеження – primary key, autoincrement
- 2) “category\_name” – назва категорії. Тип – varchar, обмеження – максимум 255 символів, унікальність, not null
- 3) “category\_description” – опис категорії. Тип – varchar, обмеження – максимум 255 символів, not null
- 4) “deleted” - позначення чи видалено категорію. Тип Boolean, обмеження - not null, стандартне значення - 0

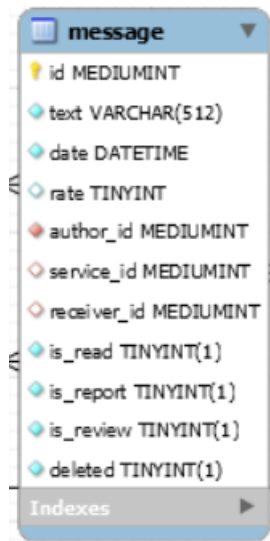
### Ordered\_service



Таблиця, в якій зберігаються дані про куплені замовлення

- 1) ”id” – унікальний ідентифікатор купленого замовлення. Тип – unsigned medium int , обмеження – primary key, autoincrement
- 2) “date” – дата купівлі замовлення. Тип – date, обмеження not null
- 3) “user\_id” – зовнішній ключ, містить id користувача, який купив замовлення. Тип – unsigned medium int, обмеження – not null
- 4) “order\_id” - зовнішній ключ, містить id сервісу, який купив користувач. Тип – unsigned medium int, обмеження – not null
- 5) “pay\_date” – дата оплати замовлення. Тип – date
- 6) “deleted” - позначення чи видалено куплене замовлення. Тип Boolean, обмеження - not null, стандартне значення - 0

## Message



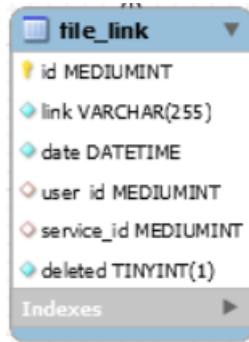
Field	Type
id	MEDIUMINT
text	VARCHAR(512)
date	DATETIME
rate	TINYINT
author_id	MEDIUMINT
service_id	MEDIUMINT
receiver_id	MEDIUMINT
is_read	TINYINT(1)
is_report	TINYINT(1)
is_review	TINYINT(1)
deleted	TINYINT(1)

Таблиця, що зберігає інформацію про повідомлення між користувачами, відгуки на замовлення а також скарги на користувачів.

- 1) "id" – унікальний ідентифікатор купленого замовлення. Тип – unsigned medium int , обмеження – primary key, autoincrement
- 2) "text" – зміст повідомлення, відгука чи скарги. Тип – varchar, обмеження – максимум 512 символів
- 3) "rate" – оцінка виконаного замовлення, якщо це є відгук. Тип tinyint.
- 4) "author\_id" – зовнішній ключ, містить id користувача – автора. Тип unsigned medium int, обмеження – not null
- 5) "service\_id" - зовнішній ключ, містить id замовлення, на яке написано відгук. Тип unsigned medium int
- 6) "receiver\_id" – зовнішній ключ, містить id користувача – отримувача повідомлення або скарги. Тип unsigned medium int
- 7) "is\_read" – поле для ідентифікації чи було прочитано повідомлення чи скарга. Тип boolean, стандартне значення 0
- 8) "is\_report" – службове поле для визначення чи це скарга. Тип boolean, стандартне значення 0
- 9) "is\_review" – службове поле для визначення чи це відгук. Тип Boolean, стандартне значення 0
- 10) "deleted" - позначення чи видалено повідомлення, скарга чи відгук. Тип Boolean, обмеження - not null, стандартне значення – 0



## File\_link



The screenshot shows a database table named 'file\_link' with the following columns and data types:

Column Name	Data Type
id	MEDIUMINT
link	VARCHAR(255)
date	DATETIME
user_id	MEDIUMINT
service_id	MEDIUMINT
deleted	TINYINT(1)

Below the columns, there is a section labeled 'Indexes' with a right-pointing arrow.

Таблиця, що зберігає інформацію про медіа-контент сервісу а також користувача.

- 1) "id" – унікальний ідентифікатор медіа-контенту. Тип – unsigned medium int , обмеження – primary key, autoincrement
- 2) "link" – поле, яке містить шлях до файлу медіа-контенту. Тип – varchar, обмеження – максимум 255 символів, not null
- 3) "date" – дата створення медіа-контенту. Тип – datetime, обмеження - not null
- 4) "user\_id" - зовнішній ключ, містить id користувача, ким медіа-контент був загрузений.
- 5) "service\_id" – зовнішній ключ, містить id замовлення, для якого медіа-контент був загрузений.
- 6) "deleted" - позначення чи видалено замовлення. Тип Boolean, обмеження - not null, стандартне значення – 0

## Нормалізація бази даних

Після побудови логічної структури, варто перевірити, чи не порушені у заданому проекті принципи нормалізації, а саме:

- Виключені повторювані групи в окремих таблицях.
- Створені окремі таблиці для кожного набору пов'язаних даних.
- Визначений кожен набір пов'язаних даних за допомогою первинного ключа.

Перевіряємо нашу базу даних по формах нормалізації:

Перша нормальна форма:

- Кожна таблиця повинна мати мінімальний набір колонок, які ідентифікують запис
- Уникати повторень груп правильно визначаючи неключові атрибути
- Кожен атрибут повинен мати одне значення.

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

Тепер перевіримо нашу базу даних на другу нормальну форму.

Друга нормальна форма:

- База даних відповідає першій нормальній формі
- Дані, що повторно з'являються в декількох рядках виносяться в окремі таблиці

Можна побачити, що ми винесли категорії замовлень в окрему таблицю для уникнення повторів.

Отже усі відношення, що подані базовими таблицями відповідають другій нормальній формі.

Третя нормальна форма:

- Схема бази даних повинна відповідати другій нормальній формі
- Будь-яке поле, що залежить від основного ключа та будь-якого іншого поля, має виноситися в окрему таблицю

Оскільки у нас відсутні поля, які залежать від основного ключа та будь-якого іншого поля, можна зробити висновок, що база даних знаходиться в третій нормальній формі.

Нормальна форма Бойса-Кодда:

3НФ не співпадає з НФБК лише тоді, коли:

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

У нас присутні 2 або більше потенційні ключі. В таблиці користувач присутні: id, username, email. Також в таблиці категорії присутні: id, category\_name. Проте усі ці значення унікальні, тобто 2 пункт не виконується. Отже НФБК виконується.

Таким чином, база даних відповідає НФБК.

## Фізична модель БД

```
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) 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 NOT NULL,  
author_id MEDIUMINT UNSIGNED NOT NULL,  
service_id MEDIUMINT UNSIGNED,  
receiver_id MEDIUMINT UNSIGNED,  
is_read BOOLEAN NOT NULL DEFAULT 0,  
is_report BOOLEAN NOT NULL DEFAULT 0,  
is_review BOOLEAN NOT NULL DEFAULT 0,  
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;**

## Тригери

Тригер забезпечує встановлення дати реєстрації користувача:

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

Тригер забезпечує створення запису в таблиці file\_link для новоствореного юзера з стандартним фото профайлу

```
CREATE TRIGGER after_insert_user  
AFTER INSERT  
ON user  
FOR EACH ROW  
INSERT INTO file_link(link, user_id) VALUES("default_avatar.jpg", NEW.id);
```

Тригер забезпечує встановлення дати завантаження медіа-контенту в таблицю file\_link:

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

Тригер забезпечує встановлення дати створення замовлення в таблицю service

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



Тригер забезпечує встановлення дати покупки замовлення в таблицю ordered\_service

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

Тригер забезпечує встановлення дати відправлення повідомлення, відгуку чи скарги

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

Тригер забезпечує “м’ягке” каскадне видалення усіх замовлень користувача, який видалив свій аккаунт.

```
DELIMITER $$  
CREATE TRIGGER after_update_user  
AFTER UPDATE  
ON user  
FOR EACH ROW  
BEGIN  
    IF NEW.deleted = 1 THEN  
        UPDATE service  
        SET service.deleted = 1  
        WHERE service.sender_id = OLD.id;  
    END IF; END $$
```

Тригер забезпечує “м’яке” каскадне видалення контенту сервісу, який було видалено

```
DELIMITER $$  
CREATE TRIGGER after_update_service  
AFTER UPDATE  
ON service  
FOR EACH ROW  
BEGIN  
IF NEW.deleted = 1 THEN  
    UPDATE file_link  
    SET file_link.deleted = 1  
    WHERE file_link.service_id = OLD.id;  
END IF; END $$
```

Тригер забезпечує заміну видаленої категорії в замовленнях на “Other” після видалення певної категорії

```
DELIMITER $$  
CREATE TRIGGER after_update_category  
AFTER UPDATE  
ON category  
FOR EACH ROW  
BEGIN  
IF NEW.deleted = 1 THEN  
    UPDATE service  
    SET service.category_id= (SELECT c.id FROM category c WHERE  
c.name="Other")  
    WHERE service.category_id= OLD.id;  
END IF;  
END $$
```

## Ділова модель

### Ділова модель організації «Сайт фрілансу»

Класи Функції	Користу вач	Сервіс	Замовле ння	Повідом лення	Категор ія	Файл
Створення сервісу	*	*			*	
Обмін повідомленням и	*			*		
Обмін файлами	*					*
Сортування за категоріями		*			*	
Замовлення сервісу	*	*	*			
Виконання замовлення	*	*	*	*		*

Отже, для забезпечення функціоналу, нам потрібні такі основні операції:

- CRUD користувача
- Додавання, видалення та вивід інформації про замовлення
- Додавання, видалення та оновлення медіа-контенту
- Додавання, видалення купленого замовлення
- Додавання, видалення та вивід інформації про відгуки на замовлення
- Додавання, видалення та вивід інформації про скарги на користувача
- Додавання, видалення та вивід повідомлень

Усі ці операції можуть комбінуватися, накладатися певні умови та доповнювати одна одну.

## Запити

Створює запис про користувача

```
INSERT INTO user(username, email, password) VALUES(username,email,password);
```

Виводить інформацію про користувача та фото його профайлу.

```
SELECT user.id, username, first_name, last_name, date_joined, file_link.link as photo
FROM user
INNER JOIN file_link ON file_link.user_id = user.id
WHERE user.id = id;
```

	id	username	first_name	last_name	date_joined	photo
▶	2	Bodual	Богдан	Левицький	2020-04-09 13:01:52	default_avatar.jpg

Оновлює інформацію про користувача, а саме прізвище та ім'я

```
UPDATE user
SET first_name = first_name, last_name = last_name
WHERE user.id = id;
```

	id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined	deleted
▶	3	pkmnxaer2l	2020-05-22 10:50:31	1	Votzik	first_name	last_name	votzik01@gmail.com	1	1	2020-04-12 19:08:43	0
*		NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

“Видаляє” користувача

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

	id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined	deleted
▶	6	fgdssd1gf	2020-06-23 09:31:53	0	Berdnka	Данило	Бердник	Berdnik_D@gmail.com	0	1	2020-05-30 09:53:41	1
*		NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Створює запис про замовлення

```
INSERT INTO service(price, title, description, sender_id, category_id)
VALUES(price,title, description, sender_id, (SELECT id FROM category WHERE
category.category_name = cat_name));
```

	id	price	title	description	date	sender_id	category_id	deleted
▶	18	10	Тест	Тестовий опис	2020-05-24 13:59:36	5	1	0
*		NULL	NULL	NULL	NULL	NULL	NULL	NULL

Виводить інформацію про замовлення та коротку інформацію про власника сервісу та категорію

```
SELECT service.id as service_id, service.title, service.price, service.description,
service.date, user.id as user_id, user.username, category.category_name as category
```

```
FROM service
```

```
INNER JOIN user ON service.sender_id = user.id
```

```
INNER JOIN category ON service.category_id = category.id;
```

	service_id	title	price	description	date	user_id	username	category
►	1	консультатор	100	консультації	2020-05-22 15:55:30	1	Reitar	Business
	3	піар менеджер	300	піар та менеджмент	2020-05-22 15:55:30	3	Votzik	Digital marketing
	2	монтажер	200	монтую відео	2020-05-22 15:55:30	2	Bodual	Graphics and design
	8	кінорежисер	800	роблю кіно	2020-05-22 15:55:30	8	Bronets	Graphics and design
	4	музикант	400	роблю музику	2020-05-22 15:55:30	4	Edw1x	Music and audio
	5	програміст	500	роблю програму	2020-05-22 15:55:30	5	Medio9ine	Programming and tech
	6	аніматор	600	роблю анімації	2020-05-22 15:55:30	6	Berdnka	Video and animation
	7	мастер рефератів	700	пишу реферати	2020-05-22 15:55:30	7	Volchok	Writing and transation
	9	копірайтер	900	підписую документи	2020-05-22 15:55:30	9	Andronalin	Writing and transation
	10	письменник	1000	пишу мемуари	2020-05-22 15:55:30	10	Seaplayer	Writing and transation

“Видаляє” замовлення

```
UPDATE service
```

```
SET deleted = 1
```

```
WHERE id = id;
```

	id	price	title	description	date	sender_id	category_id	deleted
►	1	100	консультатор	консультації	2020-05-22 15:55:30	1	1	1

Змінити медіа-контент(фото) певного користувача

```
UPDATE file_link
```

```
SET link = link
```

```
WHERE user_id = id;
```

	id	link	date	user_id	service_id	deleted
	3	link	2020-05-02 21:12:13	1	NULL	0

Завантажити медіа-контент до певного замовлення:

```
INSERT INTO file_link(link, service_id) VALUES(link, s_id);
```

	id	link	date	user_id	service_id	deleted
►	23	link	2020-05-24 14:03:32	NULL	5	0
*	NULL	NULL	NULL	NULL	NULL	NULL

Видалити медіа-контент до певного замовлення:

```
UPDATE file_link
```

```
SET deleted = 1
```

```
WHERE file_link.id = id;
```

	id	link	date	user_id	service_id	deleted
▶	3	kjbdvjkjndkvj.jpg	2020-05-02 21:12:13	1	NULL	1
★	NULL	NULL	NULL	NULL	NULL	NULL

Виводить медіа-контент сервісу, посортований по даті

```
SELECT id, link FROM file_link
```

```
WHERE service_id = id AND deleted = 0
```

```
ORDER BY date DESC;
```

	id	link
▶	16	kdndkjnfsvkfsnki.png
	15	superpuper.png
★	NULL	NULL

Виводить замовлення конкретного користувача

```
SELECT service.id as service_id, service.title, service.price, service.description,  
service.date, user.id as user_id, user.username, category.category_name as category
```

```
FROM service
```

```
INNER JOIN user ON service.sender_id = user.id
```

```
INNER JOIN category ON service.category_id = category.id
```

```
WHERE service.sender_id = id;
```

	service_id	title	price	description	date	user_id	username	category
▶	3	піар менеджер	300	піар та менеджмент	2020-05-22 15:55:30	3	Votzik	Digital marketing

Виводить замовлення по певній категорії

```
SELECT service.id as service_id, service.title, service.price, service.description,  
service.date, user.id as user_id, user.username, category.category_name as category  
FROM service  
INNER JOIN user ON service.sender_id = user.id  
INNER JOIN category ON service.category_id = category.id  
WHERE category.category_name = cat_name;
```

	service_id	title	price	description	date	user_id	username	category
▶	1	консультатор	100	консультації	2020-05-22 15:55:30	1	Reitar	Business

Купити певне замовлення певним користувачем

```
INSERT INTO ordered_service(description, user_id, order_id, pay_date)  
VALUES(description, user_id, order_id, pay_date);
```

	id	date	user_id	order_id	pay_date	deleted	description
▶	6	2020-05-24 15:55:30	5	6	2020-05-25 15:55:30	0	зробить анімацію
	11	2020-05-24 14:05:57	5	5	2020-05-02 21:12:13	0	description
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Виводить інформацію про замовлення фрілансера, які в нього купили

```
SELECT ordered_service.id as ordered_service_id, ordered_service.date,  
ordered_service.description, pay_date,  
service.id as service_id, service.title, service.price, service.description, service.date, user.id  
as user_id, user.username, category.category_name as category  
FROM ordered_service  
INNER JOIN service ON service.id = ordered_service.order_id  
INNER JOIN user ON service.sender_id = user.id  
INNER JOIN category ON service.category_id = category.id  
WHERE service.sender_id = s_id AND ordered_service.deleted = 0;
```

	ordered_service_id	date	description	pay_date	service_id	title	price	description	date	user_id	username	category
▶	3	2020-05-24 15:55:30	зробить піар	2020-05-25 15:55:30	3	піар менеджера	300	піар та менеджмент	2020-05-22 15:55:30	3	Votzik	Digital marketing

Виводить інформацію про замовлення які купив певний користувач

```
SELECT ordered_service.id as ordered_service_id, ordered_service.date,  
ordered_service.description, pay_date,  
  
service.id as service_id, service.title, service.price, service.description, service.date, user.id  
as user_id, user.username, category.category_name as category  
  
FROM ordered_service  
  
INNER JOIN service ON service.id = ordered_service.order_id  
  
INNER JOIN user ON service.sender_id = user.id  
  
INNER JOIN category ON service.category_id = category.id  
  
WHERE service.user_id= u_id AND ordered_service.deleted = 0;
```

	date	description	pay_date	service_id	title	price	description	date	user_id	username	category
►	2020-05-24 15:55:30	зробіть кіно	2020-05-25 15:55:30	8	кінорежисер	800	роблю кіно	2020-05-22 15:55:30	8	Bronets	Graphics and design

Виводить скарги подані на певного користувача,

```
SELECT ua.id as id_author, ua.username as report_author_username , ur.id as id_receiver,  
ur.username AS report_receiver_username,  
  
message.text as report  
  
FROM message  
  
INNER JOIN user ua ON message.author_id = ua.id  
  
INNER JOIN user ur ON message.receiver_id = ur.id  
  
WHERE is_report = 1 AND message.receiver_id = id;
```

	ordered_service_id	date	description	pay_date	service_id	title	price	description	date	user_id	username	category
►	3	2020-05-24 15:55:30	зробіть піар	2020-05-25 15:55:30	3	піар менеджер	300	піар та менеджмент	2020-05-22 15:55:30	3	Votzik	Digital marketing



Виводить відгуки про певний сервіс, а також інформацію про користувача, який відправив відгук та фото його профайлу

```
SELECT message.id, message.text, message.date, message.rate, user.id, user.username,
user.first_name, user.last_name, file_link.link as photo
```

```
FROM message
```

```
INNER JOIN service ON service.id = message.service_id
```

```
INNER JOIN user ON user.id = message.author_id
```

```
INNER JOIN file_link ON file_link.user_id = user.id
```

```
WHERE service.id = 1 AND is_review = 1 AND message.deleted = 0;
```

	id	text	date	rate	id	username	first_name	last_name	photo
►	7	Дуже хороший продукт купив би ще	2020-05-25 15:00:00	4	1	Reitar	Богдан	Горностай	kjbdvjkndkvj.jpg
	8	Дуже поганий продукт не купив би ще	2020-05-25 16:00:00	1	3	Votzik	Irop	Чайковский	hzz.jpg

Виводить діалог між двома користувачами, посортовано по даті:

```
SELECT u1.id, u1.username as author, u2.id, u2.username as receiver, text, date, is_read,
message.deleted
```

```
FROM message
```

```
INNER JOIN user u1 ON u1.id = message.author_id
```

```
INNER JOIN user u2 ON u2.id = message.receiver_id
```

```
WHERE ((author_id = a_id AND receiver_id = r_id) OR (author_id = r_id AND
receiver_id = a_id)) AND is_report = 0
```

```
ORDER BY message.date;
```

id	author	id	receiver	text	date	is_read	deleted
2	Kek	4	wanderlancer	Hi, i want something to buy	2020-05-23 10:10:10	1	0
4	wanderlancer	2	Kek	What do u want?	2020-05-23 10:10:20	1	0
2	Kek	4	wanderlancer	Logo to my site	2020-05-23 10:10:30	1	0

## **Висновок**

Під час виконання даної розрахункової роботи ми спроектували модель бази даних для веб-сервісу “Фріланс”. Бази даних, після нормалізації, складається з 6 таблиць, модель якої, виконану в нотації Баркера, можна побачити в розділі “Логічна схема”. Поля кожної таблиці та накладені на них обмеження було описано в розділі “Опис структури”.

Як систему керування базою даних . було обрано MySQL, оскільки вона швидка, стійка та проста у використанні. Відповідний код реалізації бази даних в MySQL представлено в розділі “Фізична модель” Для забезпечення цілісності бази даних та підтримки бізнес логіки, було написано 8 тригерів реалізація та функціонал яких також було описано в цьому розділі.

Для перевірки роботоздатності бази, тригерів та запитів, базу даних було заповнено тестовими даними. Ми переконалися в правильності написання запитів та тригерів, результати усіх запитів було показано в розділі “Запити”

В результаті виконання даної розрахункової роботи, ми отримали повнофункціональну базу даних для веб-сервісу “Фріланс”, нормалізовану до НФБК

Таким чином ми закріпили теоретичні та практичні навички, які набули під час вивчення предмету “Організація баз даних та знань”

## Список використаних літературних джерел

1. <https://dev.mysql.com/doc/>
2. <https://dev.mysql.com/doc/workbench/en/>
3. <https://www.coursera.org/learn/sql-for-data-science>
4. <http://vns.lpnu.ua/course/view.php?id=7897>
5. <https://www.w3schools.com/sql/>
6. <https://www.w3schools.in/dbms/database-normalization/>
7. <https://www.vertabelo.com/blog/barkers-erd-notation/>
8. Пасічник В.В., Резніченко В.А. Організація баз даних та знань - К.: Видавни-ча група BHV, 2006. — 384 с.: іл. — ISBN 966-552-156-X.
9. Петкович Душан. Microsoft SQL Server 2012. Руководство для начинающих. СПб.: БХВ-Петербург, 2013. — 816 с.
10. «Організація баз даних та знань» Романюк О. Н. Савчук Т. О.