

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра автоматизованих систем обробки інформації і управління

КУРСОВА РОБОТА

з дисципліни

" ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ"

Виконали:

Група:

Варіант

Керівник:

Петров А.М. та Мезеря В.В.

ІС-32

власний

Томахіна Ольга Сергіївна

Київ 2015

Завдання

на курсову роботу по дисципліні

**"Організація баз даних та
знань"**

фахове спрямування – Комп'ютерні науки

"Інформаційні управляючі системи та технології"

напрям 6.050101 "Комп'ютерні науки"

Група ІС-32

Петров А.М. та Мезеря В.В.

Тема Курсової роботи:

Щоденник спортивних досягнень.

Дата представлення роботи до захисту до « ____ » _____ 20__ р.

	Завдання видано:	
	Дата:	« ____ » _____ 20__ р.
	Керівник:	(В.Д.Попенко)

Студент: _____ (_____)

Анотації

Курсова робота на тему «Щоденник спортивних досягнень» виконана з метою закріплення, поглиблення та узагальнення знань, які отримані під час вивчення дисципліни «Організація баз даних і знань», втілення практичних навичок в області створення сучасних систем управління базами даних з оформленням відповідного текстового, програмного та ілюстративного матеріалу. В курсовій роботі розроблена ER-діаграма предметної області «Щоденник спортивних досягнень», обрана СУБД MySQL і засоби розробки PHPMyAdmin, Google Drive, GitHub та Notepad++ для реалізації заданих функцій. Обрана платформа PHP функціонування інформаційної системи. Створена база даних sql497000, що складається з **13** таблиць і заповнена даними, близькими до реальних умов використання інформаційної системи. Розроблені форми управління довідниками з можливістю вставки, оновлення і видалення даних і відповідні запити, що реалізують задану функціональність. Розроблена форма, що реалізує виконання бізнес-функцій предметної області «Щоденник спортивних досягнень». Реалізовані **15** групуючих запитів і **12** запитів, що отримують дані з декількох таблиць, а також відповідні форми. Розроблено **10** запитів і відповідних форм, що дозволяють створити і відобразити звіти.

Аннотация

Курсовая работа на тему “Дневник спортивных достижений” выполнена с целью закрепления, углубления и обобщения знаний, полученных при изучении дисциплины «Организация баз данных и знаний», воплощения практических навыков в области создания современных систем управления базами данных с оформлением соответствующего текстового, программного и иллюстративного материала.

В курсовой работе разработана ER-диаграмма предметной области “Дневник спортивных достижений”, выбрана СУБД MySQL и средства разработки PHPMyAdmin, Google Drive, GitHub та Notepad++ для реализации заданных функций. Выбрана платформа PHP для функционирования разрабатываемой информационной системы. Создана база данных sql497000 из **13** таблиц и заполнена данными, близкими к реальным условиям использования информационной системы. Разработаны формы управления справочниками с возможностью вставки, обновления и удаления данных и соответствующие запросы, реализующие заданную функциональность. Разработана форма, которая реализует выполнение бизнес-функций предметной области “Дневник спортивных достижений”. Реализовано **15** группирующих запросов и **12** запросов, которые получают данные из нескольких таблиц, а также соответствующие формы. Разработано **10** запросов и соответствующих форм, позволяющих создать и отобразить отчеты.

Annotation

Coursework on «Diary of sporting achievements» topic performed to consolidate, deepen and generalize knowledge obtained during the study "Organization of databases and knowledge" discipline, the implementation of practical skills in the field of modern database management systems with design appropriate text, program and illustrations.

According to specified coursework task, ER-diagram of subject area «Diary of sporting achievements» is developed, the MySQL DBMS and PHPMyAdmin, Google Drive, GitHub та Notepad++ as a development tools are selected for the implementation of specified functions. The PHP platform is selected for the functioning of the developed information system. Database of sql497000 consist of **13**

tables and is filled with data close to the actual conditions of the use of information systems. The forms for data management are developed with the ability to insert, update and delete data. Queries are developed for realizing specified functionality also. To realize business functions of the «Diary of sporting achievements» subject area the corresponded forms and queries are developed. **15** grouping queries and **12** queries that join multiple tables are implemented, and the appropriate forms are developed respectively. According to specified task, **10** queries and corresponding reports are developed.

Зміст

1.	Вступ	6
1.1.	Постановка задачі.....	7
1.2.	Завдання	8
2.	Основна частина.....	8
2.1.	Створення ER-діаграми	8
2.2.	Створення бази даних та заповнення її тестовими даними	11
2.3.	Розроблення необхідних запитів	13
2.3.1.	Запити від двох чи більше таблиць	13
2.3.2.	Групуючи запити	18
2.3.3.	Запити на вставку, оновлення, видалення значень.....	23
2.3.4.	Функції.....	27
2.3.5.	Процедури.....	27
3.	Висновки.....	29
4.	Список літератури.....	30
5.	Додатки	31

1. Вступ

З розвитком науково-технічного прогресу та зокрема інформаційних технологій почали з'являтися багато професій та спеціальностей, де люди вимушені майже весь робочий день проводити, сидячи та не виконуючи практично жодної фізичної праці. Також люди більше вподобали проводити свій вільний час нерухомо за комп'ютером, що вкрай негативно впливає на стан їхнього здоров'я. Аби вирішити цю проблему, деякі люди почали активно займатися спортом, з'явилося багато нових напрямків фізичної підготовки, деякі стали результатом симбіозу деяких класичних спортивних дисциплін, інші прийшли до нас з інших культур. На ринку послуг з'явилася ціла ніша, яка надає людям можливість займатись будь-яким видом фізичної культури, використовуючи найостанніші розробки в галузі спортивного одягу та обладнання, спортивного харчування та ін. Але все це потребує немалих коштів та часу, тому в деяких людях закріпилася думка, нав'язана маркетингологами про те, що спорт – це дорого, і неможливо підтримувати гарну фізичну форму, не витрачаючи значну кількість коштів та часу. Заняття спортом чомусь перейшли з необхідного обов'язку в ранг привілеї забезпечених людей. А пересічні громадяни середнього та похилого віку чомусь часто жаліються на брак часу або ресурсів для занять фізичною підготовкою або навіть соромляться цим займатись, адже це робить їх несхожими на інших, молоді ж люди часто замінюють ці заняття віртуальними розвагами або вживання речовин, що впливають на свідомість.

Ринок інформаційних технологій зараз надає багато рішень, які роблять заняття спортом легшими та більш цікавими та інтерактивними, більшість з них орієнтована на мобільні платформи та використовує різноманітні функції сучасних мобільних пристроїв, як то GPS-навігація, акселерометр, голосові підказки та поєднує їх з різноманітними соціальними функціями, що дозволяє людям ділитись досягненнями та враженнями від занять спортом, обмінюватись досвідом тощо. Більшість з них орієнтована на специфічні види спорту, зокрема ті, що пов'язані з переміщенням у просторі: біг, велоспорт та ін.

Ідея продукту розробленого нами в рамках цієї курсової роботи трішки інша: він працює використовуючи Веб-технології, тому їм можливо користуватись як з мобільних пристроїв, так і з десктопів. Ми намагались створити універсальну класифікацію тренувань для будь-якого виду спорту, щоб можна було згодом додавати їх з мінімальними змінами у інтерфейсі та структурі. Наш продукт це перш за все – щоденник, тобто головна його задача – систематизувати ваші заняття спортом, полегшити аналіз прогресу фізичної форми та планування подальших тренувань, зберігати історію ваших досягнень, полегшити аналіз помилок та травм. Він може використовуватись як обивателями, так і спортсменами-любителями, а також професіоналами.

Звичайно, слідуючи останнім віянням моди у інформаційних технологіях, ми додали до нашого застосування соціальні функції. Але наша ціль була не стільки в спілкуванні між людьми та вихвалянні власних досягнень, скільки в обміні досвідом між спортсменами різного рівня підготовки, що полегшує та пришвидшує ваш прогрес, мотивує та робить заняття спортом більш інтерактивними, приємними та веселими.

Отже наша ціль була в розробці універсальної платформи для автоматизації та систематизування занять будь-яким видом спорту. Полегшення обміну досвідом, пошуку людей, що, як і ви, зацікавленні у заняттях певним видом спорту, стимулювання виникнення духу змагання між людьми, полегшення обміну враженнями та підтримки людьми одне одного в їх прагненні стати кращими, сильнішими фізично та морально, витривалішими, більш адаптованими до стресових ситуацій, важких навантажень та інших неприємних речей, що переслідують нас у сучасному світі.

1.1. Постановка задачі

Створити веб-застосування, яке буде виконувати функцію щоденника спортивних досягнень з можливістю ділитися ними з іншими користувачами.

У кожного користувача є власний акаунт, в якому представлені:

- Персональні дані (ім'я, фізичні дані, адреса та список видів спорту).
- Підписки (список людей, за новинами яких ви слідкуєте).
- Фоловери (список людей, які слідкують за вашими новинами).

Користувач проводить тренування та записує результати, при бажанні він публікує тренування у себе на сторінці, інші користувачі мають змогу їх коментувати, оцінювати, підбадьорювати. Є можливість додати фотографії.

Тренування складається з вправ, для кожної вправи є такі основні параметри, як кількість підходів, кількість повторів у підході, номінал вправи(вага, дистанція, висота та ін. в залежності від виду спорту). Це більш-менш універсальна класифікація, адже для кожного виду спорту ці параметри приймають різне семантичне значення. Наприклад:

- Важка атлетика: жим лежачі від грудей, 3 підходи по 10разів номінал(вага) 50кг.
- Легка атлетика: спринт, 2 підходи по 5 разів номінал(дистанція) 100м.
- Спортивна гімнастика: підтягування, 3 підходи по 20 разів номінал(вага) ваша власна вага.

Навіть якщо тренування складається з однієї вправи, наприклад, просто ранкова пробіжка по парку, то можна записати так:

- Легка атлетика: біг підтюпцем 1 підхід по 1 разу номінал(дистанція) 5км.

Якщо на тренуванні ви виконали вправу, якої нема в стандартному списку вправ для даного виду спорту, можна самому додати її та описати, як вона виконується, тоді й інші користувачі зможуть її виконувати.

Якщо ви займаєтесь системно та у вас є чітка програма тренувань, то ви теж можете її додати та поетапно виконувати(програма буде складатись з запланованих тренувань), також вона буде доступна іншим користувачам і, якщо вона їм сподобається, то вони теж зможуть по ній займатись.

Ці функції(вправи та програми) дозволять початківцям швидко опановувати техніку виконання різних вправ та ефективно тренуватись, навчаючись у більш досвідчених спортсменів, а кращі спортсмени зможуть поділитись своїм дорогоцінним досвідом з іншими.

Користувач може підписатись на іншого спортсмена та спостерігати за його активністю, всі тренування цього спортсмена відображаються у його стрічці новин.

Користувачі можуть спілкуватись один з одним за допомогою особистих повідомлень.

Користувач має змогу не тільки додавати тренування за фактом, а і планувати їх наперед, а потім відмічати їх виконання.

За усіма статистичними даними: результатами тренувань, створеними та виконаними вправами будуть складатись рейтингові таблиці, де кожен спортсмен зможе оцінити свій рівень підготовки в порівнянні з іншими. Також можна буде подивитися перелік популярних видів спорту, або вправ за деякою ознакою. Наприклад: перелік популярних вправ з легкої атлетики, серед жінок, віком від 18 до 25 років на території України.

1.2. Завдання

На основі опису предметної області необхідно виконати наступне.

1. Розробити ER-діаграму предметної області.
2. Створити базу даних і наповнити її тестовими даними (схему бази даних відобразити в курсовій).
3. До цієї бази розробити по 2 запити наступних типів:
 - запити від двох або більше таблиць,
 - групуючі запити,
 - запити на вставку, оновлення, видалення значень .
4. Розробити необхідні форми введення інформації (1-3) у вигляді застосування ASP.NET або подібного (крім Microsoft Access).
5. Розробити ескізи форм необхідних звітів (1-3)

2. Основна частина

2.1. Створення ER-діаграми

Таблиця сутностей та їх опис:

№	Назва	Опис
1	Користувачі	Містить інформацію про користувача, його персональні данні.
2	Фоловери	Містить зв'язок між користувачем та його фоловерами.
3	Адреси	Містить адреси користувачів.
4	Повідомлення	Містить повідомлення між користувачами.
5	Пости	Містить пости, які можуть включати у себе: текст, фото, тренування.
6	Коментарі	Містить коментарі до посту від різних користувачів.
7	Фотографії	Містить фотографій до постів.
8	Лайки	Містить користувачів, які уподобали деякий пост.
9	Тренування	Містить опис тренування та загальні параметри.
10	Вправи	Містить вправи, які користувач виконував на тренуванні.
11	Види вправ	Містить види вправ, які користувач може виконати на тренуванні.
12	Види спорту	Містить види спорту, які відносяться до певного виду вправи.
13	Шаблони програми	Містить програми, які користувач має можливість виконати.
14	Програми	Містить програми, які користувачі виконує, або виконав.

Таблиця зв'язків між сутностями:

№	Перша сутність	Зв'язок	Друга сутність
1	Фоловери	Багато до одного	Користувачі
2	Пости	Багато до одного	Користувачі
3	Адреси	Багато до одного	Користувачі
4	Повідомлення	Багато до одного	Користувачі
5	Коментарі	Багато до одного	Користувачі

6	Тренування	Багато до одного	Користувачі
7	Програми	Багато до одного	Користувачі
8	Коментарі	Багато до одного	Пости
9	Фотографії	Багато до одного	Пости
10	Лайки	Багато до одного	Пости
11	Тренування	Один до одного	Пости
12	Шаблони тренувань	Багато до одного	Програми
13	Вправи	Багато до одного	Тренування
14	Вправи	Багато до одного	Види вправ
15	Види вправ	Багато до одного	Види спорту

ER-діаграма:

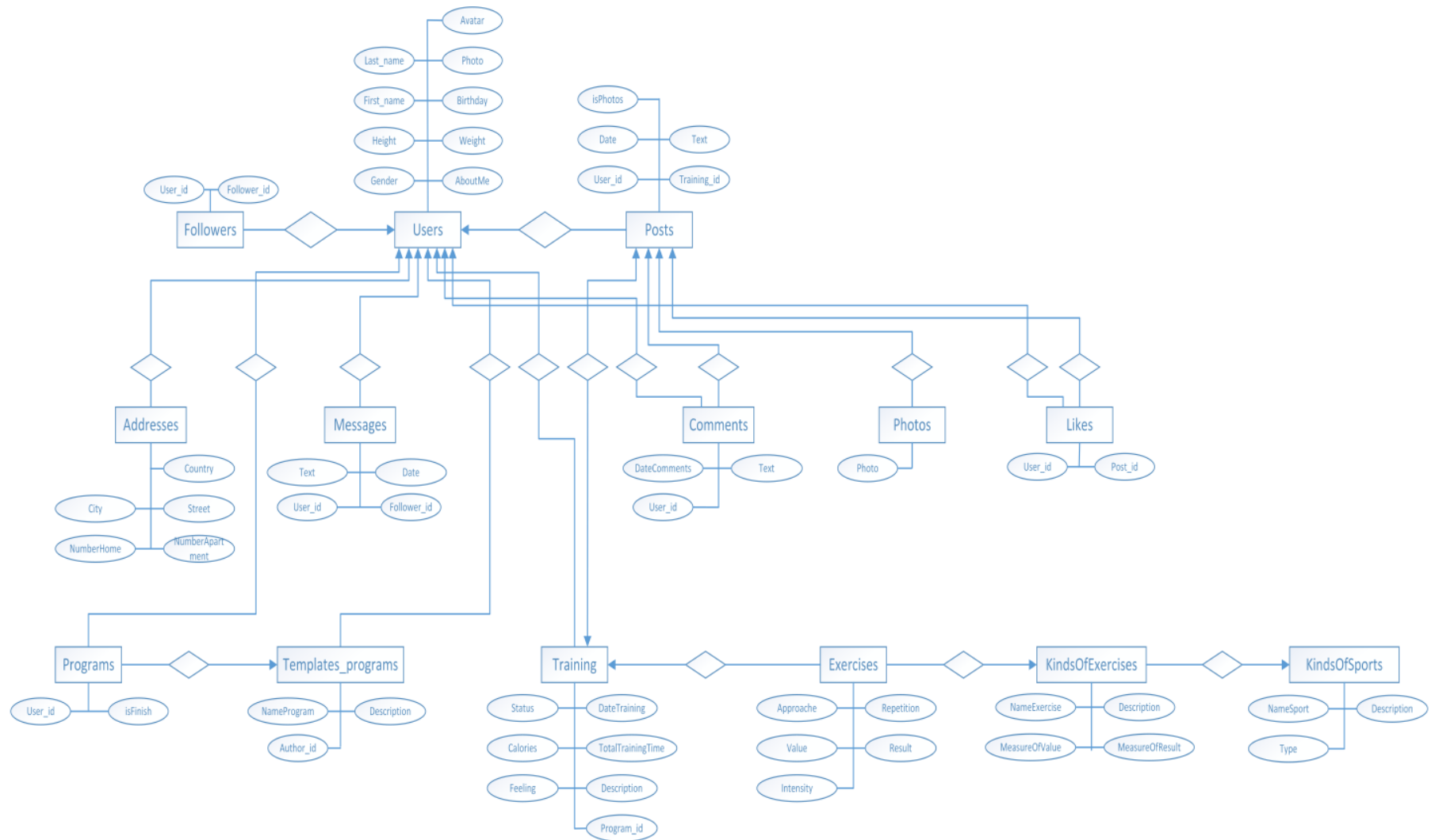


Рис. 2.1. - ER-діаграма предметної області

2.2. Створення бази даних та заповнення її тестовими даними

Схема бази даних за створеною ER- діаграмою:

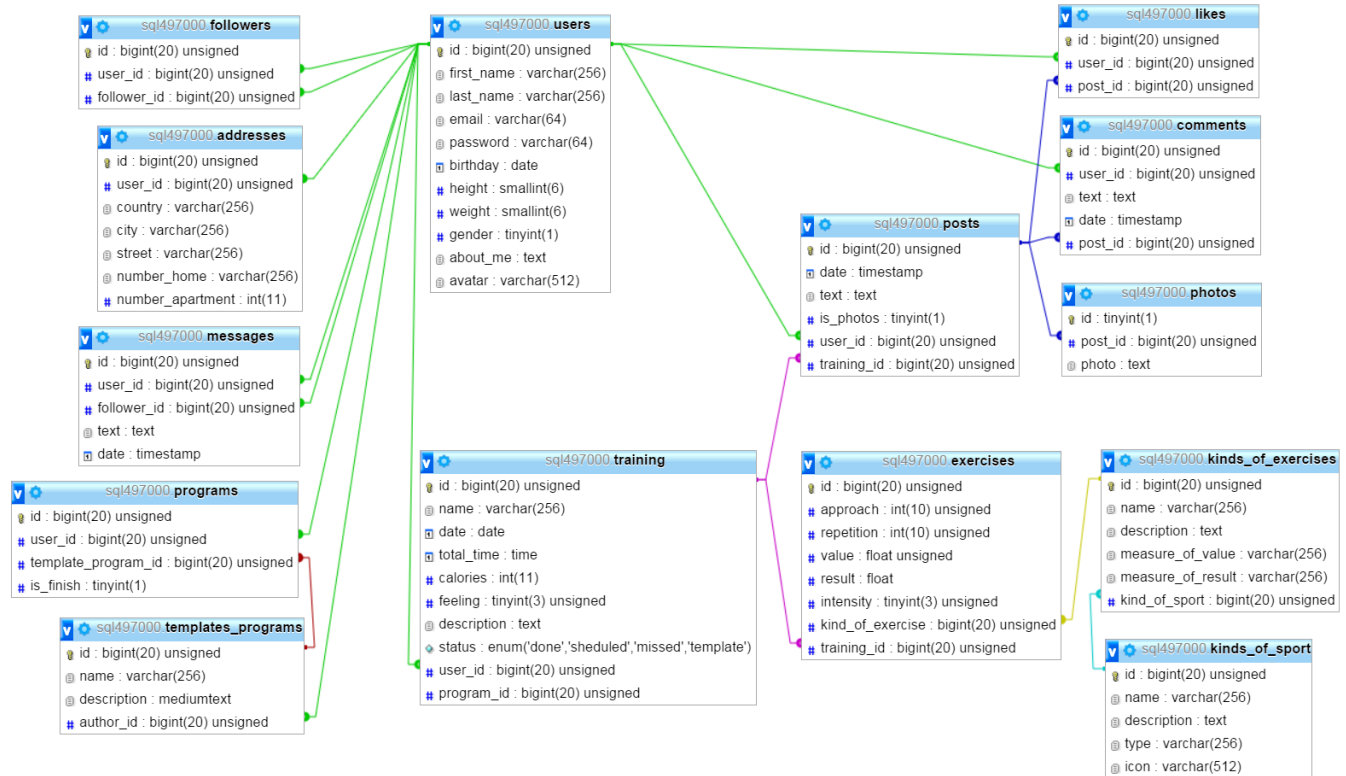


Рис. 2.2. - Схема бази даних

Нижче приведені скріншоти даних у базі даних.

	id	first_name	last_name	email	password	birthday	height	weight	gender	about_me	avatar
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Антон	Петров	anton@mail.ru	anton	1995-04-12	185	80	1	Типичный каток)	images/Anton.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Валерий	Мезеря	valera@mail.ru	valera	1996-03-09	173	60	1	Еще один каток	images/default_avatar.png
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	valera	valera	valera@mail.ru	valera	1996-12-12	180	90	1	NULL	images/default_avatar.png
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Alexander	Eryomenko	Eryomenko@gmail.com	123456789	1996-06-08	185	90	1	Hello, friends. Let me first introduce myself. My ...	images/default_avatar.png
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Petr	Scherbatyuk	Scherbatyuk@gmail.com	123456789	1996-11-03	170	77	1	Hello, friends. Let me first introduce myself. My ...	images/default_avatar.png
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	Pavel	Troyanov	Troyanov@gmail.com	123456789	1996-07-01	160	60	1	Hello, friends. Let me first introduce myself. My ...	images/default_avatar.png

Рис. 2.3. Дані з таблиці користувачув.

<input type="checkbox"/>				1	Ежедневная пробежка	2015-11-27	30:00:00	300	5	Первобытный человек вставал с первыми лучами восхо...	done	2	NULL
<input type="checkbox"/>				2	Уранный велоаезд	2015-11-27	40:00:00	400	5	Велоаезд – это ежегодное и самое массовое в Иркут...	done	1	NULL
<input type="checkbox"/>				3	Спринтерская тренировка	2015-11-12	30:00:00	500	5	Соточки	done	2	NULL
<input type="checkbox"/>				16	Легкая тренировка №1	2000-01-01	01:00:00	NULL	NULL	Легкая тренировка №1 включает в себя два упражнени...	template	NULL	1
<input type="checkbox"/>				17	Легкая тренировка №2	2000-01-03	00:30:00	NULL	NULL	Легкая тренировка №2 включает в себя одно упражнен...	template	NULL	1
<input type="checkbox"/>				19	Заплывчик №1	2015-12-05	02:00:00	444	4	NULL	done	4	NULL
<input type="checkbox"/>				20	Лесная пробежка	2015-12-06	00:30:00	300	5	NULL	done	5	NULL
<input type="checkbox"/>				22	Ride	2015-12-08	00:00:00	500	5	qwerty	done	2	NULL
<input type="checkbox"/>				25	qwert	2015-12-15	00:30:30	1234	5	qwertyuytrew	sheduled	2	NULL
<input type="checkbox"/>				27	Простая тренировка	2015-12-20	04:44:34	0	3		sheduled	1	NULL
<input type="checkbox"/>				28	Тяжелая тренья	2015-12-21	03:23:07	555	3		sheduled	1	NULL
<input type="checkbox"/>				30	Изи тренья	2015-12-14	01:20:00	0	5	Ананас, помидор и огурец	done	1	NULL
<input type="checkbox"/>				31	Лесная прогулка	2015-12-14	01:20:00	0	3		sheduled	1	NULL

Рис. 2.4. Дані з таблиці тренувань.

<input type="checkbox"/>				1	1	1	5000	1200	3	1	1
<input type="checkbox"/>				2	1	1	15	2400	3	12	2
<input type="checkbox"/>				3	1	5	200	50	5	4	3
<input type="checkbox"/>				4	1	3	100	30	5	5	3
<input type="checkbox"/>				8	3	2	30	50	3	2	16
<input type="checkbox"/>				9	2	4	60	55	3	3	16
<input type="checkbox"/>				10	1	1	100	120	2	4	17
<input type="checkbox"/>				12	2	2	50	40	3	15	19
<input type="checkbox"/>				13	5	3	5	600	5	12	19
<input type="checkbox"/>				14	1	1	7000	1800	4	1	20
<input type="checkbox"/>				17	1	1	25	3600	4	12	22
<input type="checkbox"/>				20	0	0	0	0	5	1	25
<input type="checkbox"/>				22	1	1	100	4	3	12	27

Рис. 2.5. Дані з таблиці вправ.

				id	date	text	is_photos	user_id	training_id
<input type="checkbox"/>				1	2015-11-26 13:14:41	крытое поле MAX_FILE_SIZE (значение необходимо ука...	1	1	NULL
<input type="checkbox"/>				2	2015-11-26 13:14:41	Данная возможность позволяет загружать как тексто...	NULL	2	1
<input type="checkbox"/>				3	2015-12-01 01:17:24	GHbdtN	NULL	1	2
<input type="checkbox"/>				38	2015-12-16 14:53:26	NULL	NULL	2	22
<input type="checkbox"/>				39	2015-12-16 23:27:29	NULL	NULL	2	70

Рис. 2.6. Дані з таблиці постів.

2.3. Розроблення необхідних запитів

2.3.1. Запити від двох чи більше таблиць

У новинах для перегляду доступні пости. Пост може містити такий контент: текст, текст та фото, тренування. Стрічка новин формується в залежності від того, на кого Ви підписані та власні пости. Запит на формування стрічки новин повертає усю пости.

Лістинг SQL - запиту:

```
SELECT posts.id, posts.date, posts.text, posts.is_photos, users.first_name,
users.last_name, users.avatar, training.name, training.total_time,
training.calories, training.description, training.feeling
FROM (posts LEFT JOIN users on users.id = posts.user_id)
      LEFT JOIN training on posts.training_id = training.id
WHERE posts.user_id = (SELECT follower_id
                      FROM followers
                      WHERE user_id = 1 ) OR posts.user_id = 1;
```

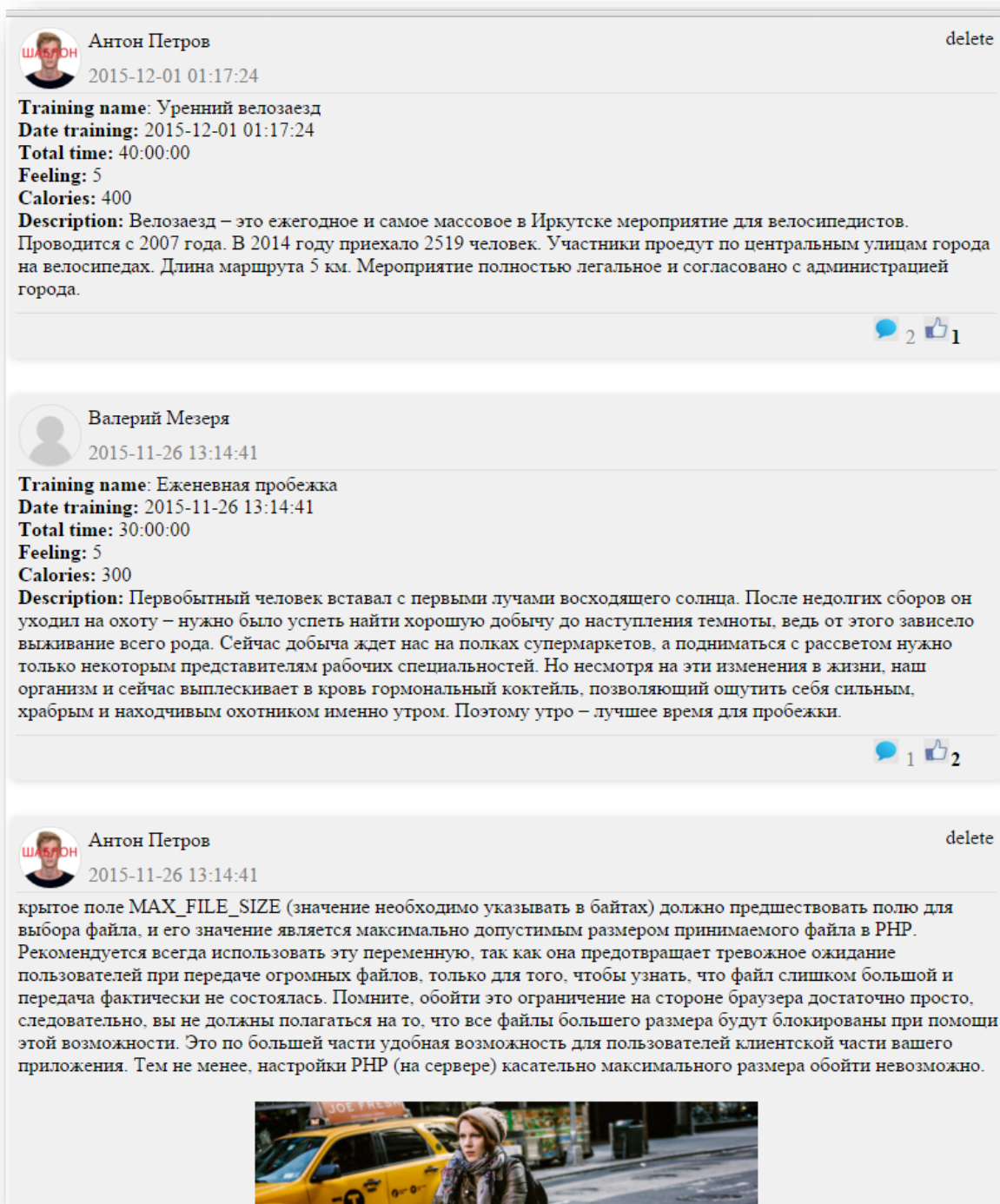


Рис.2.3. - реалізація в інтерфейсі запиту на формування стрічки новин.

Щоб вивести коментарі до посту, як це було задумано, ми використовуємо запит від двох таблиць.

Лістинг SQL - запиту:

```
SELECT text, date, first_name, last_name, avatar
FROM comments LEFT JOIN users ON comments.user_id = users.id
WHERE post_id = 1;
```

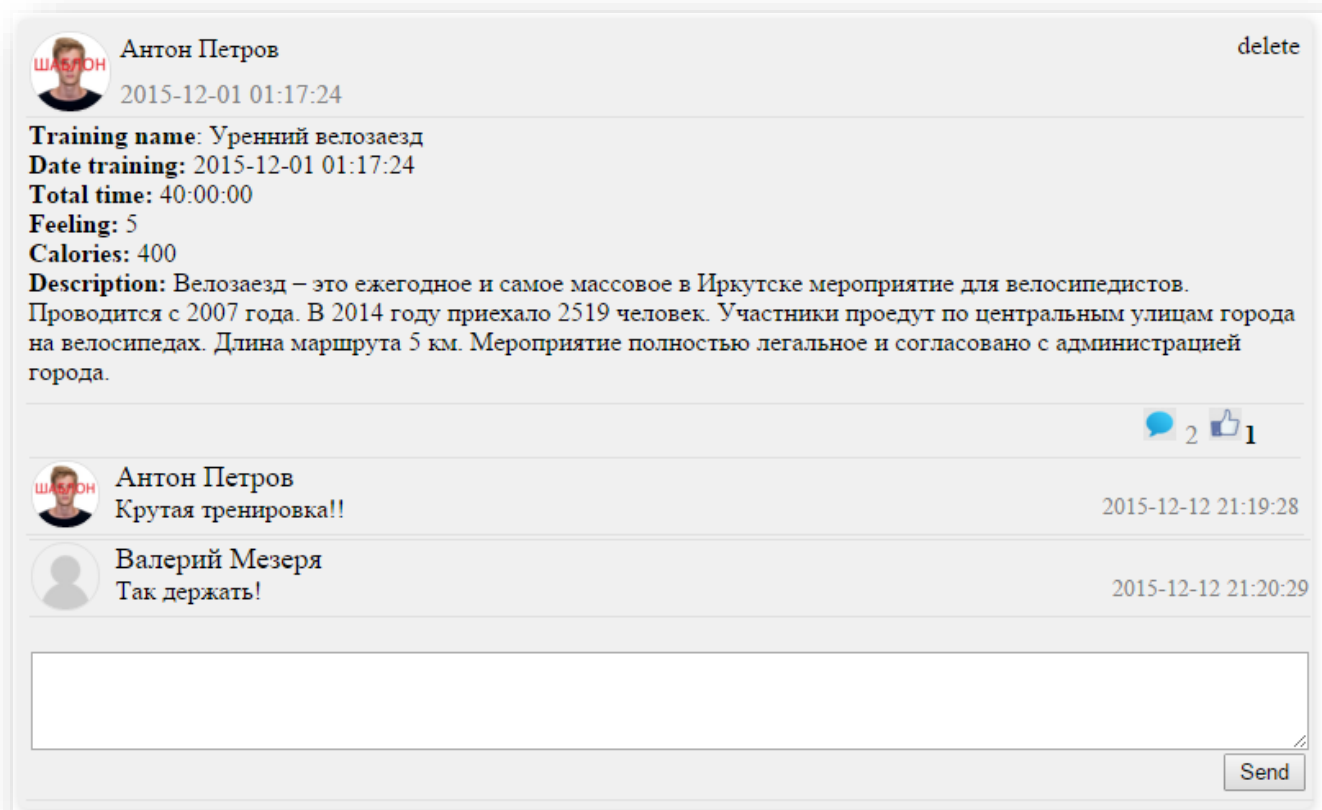


Рис.2.4. - реалізація в інтерфейсі запиту на вивід коментарів посту.

У вкладці повідомлення відображені діалоги з ким ви переписуєтесь та останнє повідомлення яким було завершено діалог. Запит, який повертає останнє повідомлення в діалозі.

Лістинг SQL - запиту:

```
SELECT avatar, text, date
FROM messages JOIN users ON messages.user_id = users.id
WHERE (user_id = 1 AND follower_id = 2) OR (user_id = 2 AND follower_id = 1)
ORDER BY messages.date DESC
LIMIT 1;
```







Dialogues	
 <div> Валерій Мезеря 2015-12-08 16:09:10 </div>	 <div> Ееее! </div>
 <div> valera valera 2015-12-12 21:32:17 </div>	 <div> я все понял. Набери меня! </div>
 <div> Petr Scherbatyuk 2015-12-12 21:31:36 </div>	 <div> Пока :) </div>

Рис. 2.5. - реалізація в інтерфейсі запиту на вивід останнього повідомлення у діалозі .

Запит, який повертає історію переписки з користувачем.

Лістинг SQL - запиту:

```

SELECT avatar, first_name, last_name, text, date
FROM messages LEFT JOIN users ON messages.user_id = users.id
WHERE (user_id = 1 AND follower_id = 1) OR (user_id = 1 AND follower_id = 1)
ORDER BY messages.date;

```


Dialog with Валерий

	Антон Петров Привет1	2015-12-01 01:19:51
	Валерий Мезеря привет2	2015-12-01 01:20:12
	Антон Петров как твои дела?	2015-12-01 01:20:40
	Валерий Мезеря хорошо. А твои как?	2015-12-01 01:21:00
	Антон Петров отлично	2015-12-01 01:21:42
	Антон Петров спасибо	2015-12-01 01:21:56
	Валерий Мезеря Ееее!	2015-12-08 16:09:10

type your message...

Рис. 2.6. - реалізація в інтерфейсі запиту на вивід історії переписки.

Пошук користувачів. Запит формується в залежності від фільтру та повертає користувачів, які відповідають таким параметрам як:

- Ім'я, прізвище.
- Стать (чоловік, жінка або обидва).
- Вік.
- Зріст та вага.
- Регіон проживання (країна та місто).

Лістинг SQL - запиту:

```
SELECT users.id as id, first_name, last_name, country, city, avatar
FROM users LEFT JOIN addresses ON users.id = addresses.user_id
WHERE (first_name LIKE ("%") OR last_name LIKE ("%"))
      AND country LIKE ("%")
      AND city LIKE ("%")
      AND gender = 1
      AND age(users.id) >= 18 AND age(users.id) <= 21
```

```
AND height >= 100 AND height <= 200
AND weight >= 50 AND weight <= 100;
```

The interface shows a search bar with the placeholder text "enter user name you want to search" and a "SEARCH" button. Below the search bar is a list of user profiles, each with a circular profile picture, a name, and a location. The profiles are:

- Антон Петров (Ukraine, Kiev) - Profile picture shows a person with a red "ШАБЛОН" watermark.
- Валерий Мезеря (Ukraine, Kiev)
- valera valera (Ukraine, Kiev)
- Alexander Eryomenko (Ukraine, Chernihiv)
- Petr Scherbatyuk (Ukraine, Odessa) - Includes a "Subscribe" button.
- Pavel Troyanov (Canada, Ottawa) - Includes a "Subscribe" button.

On the right side, there are filters:

- Country: Ukraine
- City: Kyiv
- Age from 16 to 40
- Gender: ☒ Man ☐ Woman
- [Other](#)
- Height from 120 to 190
- Weight from 60 to 90

Рис.2.7. - реалізація в інтерфейсі запиту пошуку людей.

2.3.2. Групуєчи запити

Групуєчий запит, який повертає назву спорту та кількість людей, які ним займалися в залежності від таких параметрів як:

- За деякий проміжок часу (за весь час, за останній місяць, за пів року та за рік).
- Тільки друзі.
- Стать (чоловік, жінка або обидва).
- Вік.
- Регіон проживання (країна та місто).

Лістинг SQL - запиту:

```
SELECT count(user_with_sport_table.name) AS count_users, user_with_sport_table.name
FROM (SELECT training.user_id, kinds_of_sport.name
      FROM ((training LEFT JOIN exercises ON training.id = exercises.training_id)
            LEFT JOIN kinds_of_exercises ON exercises.kind_of_exercise =
kinds_of_exercises.id)
      LEFT JOIN kinds_of_sport ON kinds_of_sport.id =
kinds_of_exercises.kind_of_sport)
```

```

RIGHT JOIN addresses ON training.user_id = addresses.user_id
WHERE training.status = 'done'
      AND training.user_id IN (SELECT users.id
                                FROM users
                                WHERE users.gender = 1
                                      AND age(users.id) >= 10
                                      AND age(users.id) <= 21
                                      AND users.id IN (SELECT user_id
                                                       FROM followers
                                                       WHERE follower_id = 1))

      AND training.date > back_date(1)
      AND training.date <= back_date(0)
      AND addresses.country LIKE ("%Украина%")
      AND addresses.city LIKE ("%Киев%")
      AND kinds_of_sport.name LIKE ("%Бег%")
GROUP BY training.user_id, kinds_of_sport.id) user_with_sport_table
GROUP BY user_with_sport_table.name
ORDER BY count_users DESC;

```

NEWS
TRAINING
PEOPLE
MESSAGES
STATISTICA
PROFILE
LOG OUT

Sport | Exercise | People

1	Бег	x2	Top exercise for: current month ▾ <input type="checkbox"/> Only of my friends Gender: <input checked="" type="radio"/> Both <input type="radio"/> Man <input type="radio"/> Woman Age from <input type="text" value="16"/> to <input type="text" value="40"/> Country: <input type="text" value="Ukraine"/> City: <input type="text" value="Kyiv"/>
2	Велоспорт	x2	
3	Плавание	x1	

Рис. 2.8. - реалізація в інтерфейсі запиту статистики видів спорту.

Групуючий запит, який повертає назву вправи, опис виконання вправи та кількість людей, які займалися даною вправою в залежності від таких параметрів як:

- Вид спорту (Легка атлетика, плавання, гребля і т.д.)
- За деякий проміжок часу (за весь час, за останній місяць, за пів року та за рік).
- Тільки друзі.
- Стать (чоловік, жінка або обидва).
- Вік.
- Регіон проживання (країна та місто).

Лістинг SQL - запиту:

```
SELECT table_with_exercise.name_exercise,
table_with_exercise.exercises_description,
count(table_with_exercise.count_exercise) AS count_people
FROM (SELECT count(training.user_id) AS count_exercise, kinds_of_exercises.name AS
name_exercise, kinds_of_exercises.description AS exercises_description
FROM (((training LEFT JOIN exercises ON training.id = exercises.training_id)
LEFT JOIN kinds_of_exercises ON exercises.kind_of_exercise =
kinds_of_exercises.id)
LEFT JOIN kinds_of_sport ON kinds_of_sport.id =
kinds_of_exercises.kind_of_sport)
RIGHT JOIN addresses ON training.user_id = addresses.user_id
WHERE training.status = 'done'
AND kinds_of_sport.name = 'Бег'
AND training.user_id IN (SELECT users.id
FROM users
WHERE users.gender = 1
AND age(users.id) >= 10
AND age(users.id) <= 19
AND users.id IN (SELECT user_id
FROM followers
WHERE follower_id = 1))
AND training.date > back_date(1)
AND training.date <= back_date(0)
AND addresses.country LIKE ("%Украина%")
AND addresses.city LIKE ("%Киев%")
AND kinds_of_exercises.name LIKE ("%Пробежка%")
GROUP BY kinds_of_exercises.name, training.user_id) table_with_exercise
GROUP BY table_with_exercise.name_exercise;
```

NEWS
TRAINING
PEOPLE
MESSAGES
STATISTICA
PROFILE
LOG OUT

Sport | Exercise | People

enter exercise name you want to search

1 "Пробежка"
Description: Преодоление бегом произвольной дистанции.

SEARCH

Kind of sport:
Ber

Top exercise for:
current month

☐ Only of my friends

Gender:
☒ Both
☐ Man
☐ Woman

Age from 16 to 40

Country:
Ukraine

City:
Kyiv

Рис. 2.9. - реалізація в інтерфейсі запиту статистики вправ.

Групуючий запит, який повертає детальну інформацію про вправу ім'я користувача, кількість підходів у вправі, кількість повторів в відході, величину вправи та одиницю виміру величини, результат вправи та одиницю виміру результату та інтенсивність виконання. Запит автоматично сортирується від обраного користувачем параметру (ім'я, к-ті підходів, к-ті повторів, величини вправи, результату вправи, або інтенсивності).

Всі ці вправи фільтруються за такими:

- Назва вправи.
- За деякий проміжок часу (за весь час, за останній місяць, за пів року та за рік).
- Тільки друзі.
- Стать (чоловік, жінка або обидва).
- Вік.
- Зріст та вага.
- Регіон проживання (країна та місто).

Лістинг SQL - запиту:

```

SELECT users.first_name, users.last_name, exercises.approach, exercises.repetition,
exercises.value, kinds_of_exercises.measure_of_value, exercises.result,
kinds_of_exercises.measure_of_result, exercises.intensity
FROM (((training LEFT JOIN exercises ON training.id = exercises.training_id)
LEFT JOIN kinds_of_exercises ON exercises.kind_of_exercise = kinds_of_exercises.id)
LEFT JOIN kinds_of_sport ON kinds_of_sport.id = kinds_of_exercises.kind_of_sport)
RIGHT JOIN addresses ON training.user_id = addresses.user_id)
LEFT JOIN users ON training.user_id = users.id
WHERE training.status = 'done'
      AND training.user_id IN (SELECT users.id
                                FROM users
                                WHERE users.gender = 1
                                      AND age(users.id) >= 10
                                      AND age(users.id) <= 19
                                      AND users.id IN (SELECT user_id
                                                        FROM followers
                                                        WHERE follower_id = 1))

      AND training.date > back_date(1)
      AND training.date <= back_date(0)
      AND addresses.country LIKE ("%Украина%")
      AND addresses.city LIKE ("%Киев%")
      AND kinds_of_exercises.name LIKE ("%Пробежка%")
      AND kinds_of_sport.name LIKE ("%Бег%")
      AND users.height >= 100
      AND users.height <= 200
      AND users.weight >= 30
      AND users.weight <= 100
ORDER BY users.first_name, users.last_name;

```

NEWS
TRAINING
PEOPLE
MESSAGES
STATISTICA
PROFILE
LOG OUT

Sport | Exercise | People

enter person name you want to search

SEARCH

#	Name	Approache	Repetition	Value	Result	Intensity
0	Антон Петров	1	1	1	1	5
1	Антон Петров	1	1	1	1	5
2	Валерий Мезеря	1	1	1	1	5
3	Валерий Мезеря	1	1	10000	30	3

Exercise:

Пробежка

▼

Top exercise for:

current month

▼

☐ Only of my friends

Gender:

☒ Both
☐ Man
☐ Woman

Age from

16

to

40

Heigth from

120

to

190

Weight from

60

to

90

Country:

Ukraine

City:

Kyiv

Рис. 2.10. - реалізація в інтерфейсі запиту статистики вправ по особам та їх результатам.

2.3.3. Запити на вставку, оновлення, видалення значень.

Запит на додавання нового повідомлення.

Лістинг SQL - запиту:

```
INSERT INTO messages (user_id, follower_id, text) VALUES(1, 2, "Hello!");
```

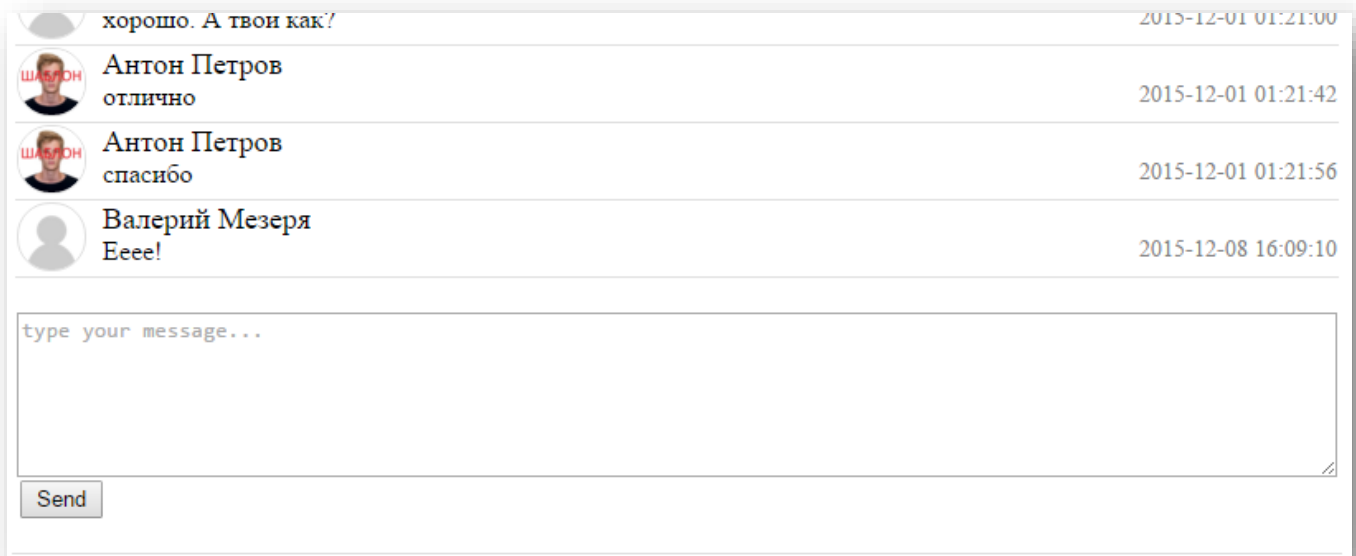


Рис. 2.11 - реалізація в інтерфейсі запиту на додавання повідомлення.

Два запита на додавання та видалення лайків.

Лістинг SQL - запиту:

```
INSERT INTO likes (post_id, user_id) VALUES(1,3);
```

```
DELETE FROM likes WHERE post_id = 1 AND user_id = 1;
```

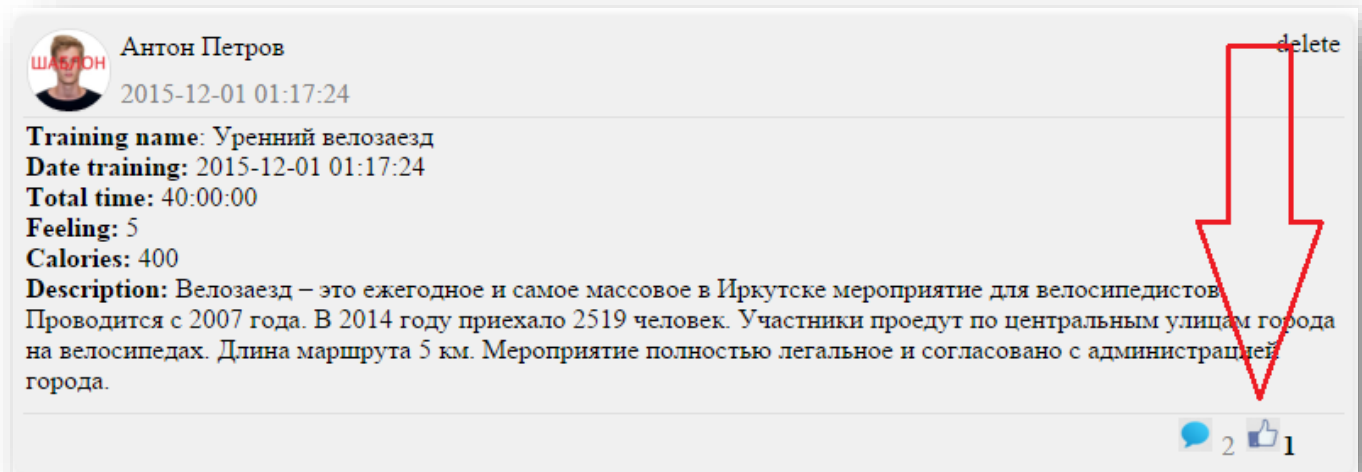


Рис. 2.12 - реалізація в інтерфейсі запиту видалення та додавання лайка.

Зміна статусу тренування з «запланована» на «виконана». Запит стає доступний тільки в день запланованого тренування.

Лістинг SQL - запиту:

```
UPDATE training SET status = 1 WHERE id = 1;
```

Лесная прогулка
2015-12-14

delete

Total time: 41:50:30
Feeling: 3
Calories: 0

perform

Sport	Exercise	Approache	Repetition	Value	Result	Intensity
Велоспорт	Велозаезд	1	1	50 км	30000 сек	4

Description:

share

Рис.2.13. - реалізація в інтерфейсі запиту зміни статусу тренування.

Реєстрація користувача мережі. Запит на створення нового користувача, додавання персональних даних.

Лістинг SQL - запиту:

```
INSERT INTO users (first_name, last_name, email, password, birthday, height,  
weight, gender, country, city, street, house, apartment)  
VALUES('Anton','Petrov', 'anton@mail.ru', '123456789', 2015-04-12, 186, 80, 1,  
'Ukraine', 'Kiev', 'Salutna', 12, 228);
```

First name	Birhtday	Country
<input type="text"/>	<input type="text" value="ДД. ММ. ГГГГ"/>	<input type="text" value="Ukraine"/>
Last name	Height	City
<input type="text"/>	<input type="text" value="185 (cm)"/>	<input type="text" value="Kyiv"/>
E-mail	Weight	Street
<input type="text"/>	<input type="text" value="77 (rg)"/>	<input type="text" value="пр-т Победы"/>
Password	Gender	House
<input type="password"/>	<input type="text" value="Man"/>	<input type="text" value="10"/>
Repeat password		Apartment
<input type="password"/>		<input type="text" value="13"/>
<input type="button" value="REGISTRATION"/>		

Рис. 2.14. - реалізація в інтерфейсі запиту реєстрації.

Видалення програми у користувача.

Лістинг SQL - запиту:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION;
```

```
DELETE FROM exercises WHERE exercises.training_id IN (SELECT training.id
                                                         FROM training
                                                         WHERE training.program_id =
1 AND training.user_id = 2 AND training.status = 'sheduled');
```

```
DELETE FROM training WHERE training.program_id = 1 AND training.user_id = 2 AND
training.status = 'sheduled';
```

```
DELETE FROM programs WHERE programs.user_id = 1 AND programs.template_program_id =
1 AND programs.is_finish = 0;
```

```
COMMIT;
```

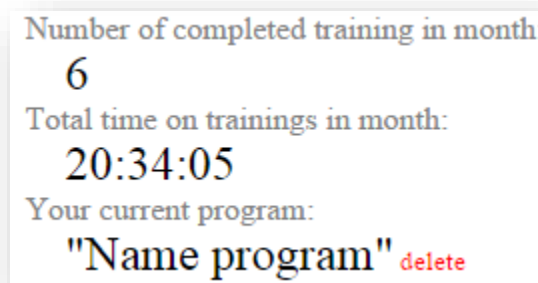


Рис. 2.15. - реалізація в інтерфейсі запиту реєстрації.

2.3.4. Функції

При розробці бази даних було створено дві допоміжні функції `age()` та `back_date()`. Функція `age()` приймає на вхід номер користувача та повертає його повну кількість років.

Лістинг SQL - запиту:

```
BEGIN DECLARE age_num INT;  
SELECT YEAR(CURDATE()) - YEAR(birthday) INTO age_num  
FROM users  
WHERE id = user_id;  
RETURN age_num;  
END
```

Функція `back_date()` приймає на вхід кількість місяців, а повертає дату початку місяця мінус кількість прийнятих місяців.

Лістинг SQL - запиту:

```
BEGIN  
DECLARE cur_month date;  
SELECT DATE_ADD(DATE_ADD(CURDATE(),Interval -DAYOFMONTH(CURDATE()) DAY),  
Interval -month MONTH) INTO cur_month;  
RETURN cur_month;  
END
```

2.3.5. Процедури

Було створено дві процедури, які в зв'язці підписують користувача на програму. Одна додає процедура підписує користувача та додає ряд тренувань, а друга додає вправи до тренувань.

Лістинг SQL – запиту першої процедури:

```
delimiter //  
CREATE PROCEDURE apply_program(IN apply_user_id INT, IN apply_template_prog_id INT)  
BEGIN  
DECLARE new_prog_id INT;  
  
INSERT INTO programs (user_id, template_program_id, is_finish)
```

```

VALUES( apply_user_id, apply_template_prog_id, 0);

SELECT programs.id INTO new_prog_id FROM programs
WHERE programs.user_id = apply_user_id AND
programs.template_program_id = apply_template_prog_id AND
programs.is_finish = 0;

INSERT INTO training ( name, date ,total_time, calories, feeling, description,
status, user_id, program_id)
SELECT training.name, DATE_ADD(CURDATE(),Interval DAYOFYEAR(date) DAY),
training.total_time, training.calories, training.feeling, training.description,
'sheduled', apply_user_id, new_prog_id
FROM training
WHERE status = 'template' AND program_id = apply_template_prog_id AND user_id IS
NULL;

CALL apply_program_exercise(apply_user_id, new_prog_id);
END//
delimiter;

```

Лістинг SQL – запиту другої процедури:

```

delimiter //
CREATE PROCEDURE apply_program_exercise(IN apply_user_id INT, IN new_prog_id INT)
BEGIN
DECLARE new_training_id INT;
DECLARE new_training_name VARCHAR(64);
DECLARE training_count INT;
DECLARE cur CURSOR FOR SELECT training.id, training.name FROM training WHERE
user_id = apply_user_id AND program_id = new_prog_id AND status = 'sheduled';

SELECT count(*) INTO training_count FROM training WHERE user_id = apply_user_id AND
program_id = new_prog_id AND status = 'sheduled';

OPEN cur;

WHILE training_count > 0 DO
FETCH cur INTO new_training_id, new_training_name;

INSERT INTO exercises (approach, repetition, value, result, intensity,
kind_of_exercise, training_id)
SELECT approach, repetition, value, result, intensity, kind_of_exercise,
new_training_id
FROM exercises LEFT JOIN training ON exercises.training_id = training.id
WHERE training.name = new_training_name AND training.status = 'template' AND
training.user_id IS NULL;
SET training_count := training_count - 1;
END WHILE;
CLOSE cur;
END//
delimiter;

```

3. Висновки

В результаті виконання курсової роботи було створено продукт, який дозволяє людям, котрі займаються спортом або просто фізичною активністю зберігати результати своїх тренувань, аналізувати їх, поліпшувати програму тренувань, слідкувати за успіхами своїх друзів та підбадьорювати їх, переймати досвід в більш досвідчених колег.

Основний інструмент керування своїми заняттями – це розділ «Тренування», тут користувач має змогу побачити детальну інформацію про всі свої тренування, додати нове, отримати наглядну інформацію про розклад своїх тренувань з календарю, оцінити деякі статистичні дані, як то загальний час та кількість тренувань за місяць, може відфільтрувати тренування за статусом «Виконані», «Пропущені» або «Заплановані». Також є змога побачити свою поточну програму тренувань.

На початку розробки перед нами стояла досить складна задача: побудувати універсальну класифікацію занять для різних видів спорту, що дозволила б легко порівнювати результати один з одним, вести статистику. Ми вирішили цю задачу так: основна одиниця тренування в нас – це Вправа. Кожна вправа відноситься до певного виду спорту та являє собою певний вид активності в рамках цього спорту, наприклад спринт на 100м в бігу, заплив кролем на 50 м в плаванні, жим лежачі від грудей у важкій атлетиці.

Кожний вид спорту має звичайно не фіксований набір вправ, користувачі мають змогу їх додавати та вибрати вже існуючі, можливо ті, про котрі не знали, але хочуть спробувати, для цього кожна створена вправа забезпечується докладним описом виконання та різних технічних нюансів.

Дуже гарний інструмент для обміну досвідом між спортсменами – це програми тренувань.

Програма – це набір тренувань з певних видів спорту. Вона переслідує певну ціль, наприклад програма схуднення, програма підготовки до марафону або скажімо програма підвищення силових якостей. Програми можуть додавати як адміни сайту, так і користувачі, акцент звичайно зроблений на користувальницькі програми. Модель використання така: наприклад ви впродовж місяця готувались до марафону, ви записували усі свої тренування, фіксували стан здоров'я та настроїв, після чого виступили на змаганнях та показали гарний результат, отже ваша програма тренувань виявилася ефективною, і тепер ви зможете створити її, та вона стане доступна усім користувачам, кожен з них може підписатись на неї та, використовуючи ваш досвід, якісно підготуватись до змагань. Якість програми буде визначатись її популярністю, тобто чим більше користувачів її виконали, тим кращою вона є.

Отже, ми реалізували власне бачення того, як має виглядати програмний продукт, такий собі «щоденник-помічник» з соціальними функціями для людей, які займаються спортом або просто тримають себе у формі та бажають зробити цей процес простішим, приємнішим та веселішим.

Ссилка на репозиторій GitHub:

<https://github.com/ValeriyMezeria/sportdiary.git>

4. Список літератури

1. Ульман Дж. Основы систем баз данных. Пер.с англ. – М.: Финансы и статистика, 1983. – 334 стр.
2. Гарсиа-Молина, Гектор, Ульман, Джеффри, Д., Уидом,Дженнифер Системы баз данных. Полный курс.: Пер.с англ. – М. : Издательский дом “Вильямс”, 2003. – 1088 с. : ил. – Парал. тит. англ.
3. Організація баз даних та знань. Курсова робота. Навчальний посібник. для студентів з дисципліни «Організація баз даних та знань», що навчаються за напрямом 0501 «Комп’ютерні науки» (кредитно-модульна система для денної форми навчання). /Укл.: В.В. Гавриленко, Е.Г. Логачов. – К.: НТУ, 2008. – 146 с.
4. Руководство по программированию на С#. [Електронний ресурс].Режим доступу: <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>.
5. Форматування SQL-запитів онлайн. [Електронний ресурс].Режим доступу: <http://sqlformat.org/>
6. Система питань і відповідей з програмування StackOweflow.[Електронний ресурс]. Режим доступу: <http://stackoverflow.com/>
7. Керівництво MySQL.[Електронний ресурс]. Режим доступу: <http://dev.mysql.com/doc/refman/5.7/en/>

5. Додатки

controller_feed.php

```
<?php
session_start();
include 'application/core/authentication.php';

class Controller_Feed extends Controller
{

    function action_feed()
    {
        $authentication = new Authentication();
        $auth_opt = $authentication->get_auth_opt();

        $model = new Model_Feed();

        if(isset($_GET['delete']))
        {
            $model->delete_post($_GET['delete']);
        }

        if(isset($_GET['like_post_id']))
        {
            $model->like_post($_GET['like_post_id'], $_SESSION['user_id']);
        }

        if(isset($_GET['recieve_comment']))
        {
            $model->add_comment($_GET['recieve_comment']);
        }

        if($_GET['recieve_post'] == 1)
        {
            if(isset($_FILES['userfile']['tmp_name']))
            {
                $destination = 'images/'.$_FILES['userfile']['name'];
                move_uploaded_file($_FILES['userfile']['tmp_name'], $destination);
            }
            $model->add_post($destination);
        }

        $data = $model->get_posts($_SESSION['user_id']);

        $this->view->generate('view_feed.php', 'view_skeleton.php', $data,
        $auth_opt);
    }

    function action_404()
    {
        $this->view->generate('view_404.php', 'view_skeleton.php', $data,
        $auth_opt);
    }
}
```

controller_training.php

```
<?php
session_start();
include 'application/core/authentication.php';

class Controller_Training extends Controller
{

    function action_training()
    {
        $host = $ _SERVER['HTTP_HOST'];
        $authentication = new Authentication();
        $auth_opt = $authentication->get_auth_opt();

        $model = new Model_Training();

        if(isset($_GET['add_training_post']))
        {
            $model->add_training_post($_GET['add_training_post']);
            echo '<script type="text/javascript">location.href =
"http://'. $host. '/Feed/feed";</script>';
        }

        if(isset($_GET['delete_training']))
        {
            $model->delete_training($_GET['delete_training']);
        }

        if(isset($_GET['delete_program']))
        {
            $model->delete_program($_GET['delete_program']);
        }

        if(isset($_GET['perform']))
            $model->done_training($_GET['perform']);

        $options['status'] = $_GET['status'];

        if($_GET['status'] == 'missed')
            $data = $model->get_training($auth_opt['user_id'], 'missed');
        else if ($_GET['status'] == 'sheduled')
            $data = $model->get_training($auth_opt['user_id'], 'sheduled');
        else
        {
            $data = $model->get_training($auth_opt['user_id'], 'done');
            $options['status'] = 'done';
        }

        $options['cur_program'] = $model->get_cur_program();

        $this->view->generate('view_training.php', 'view_skeleton.php', $data,
$auth_opt, $options);
    }

    function action_add_training()
    {
```



```

        $authentication = new Authentication();
        $auth_opt = $authentication->get_auth_opt();
        $model = new Model_Training();
        $data = $model->get_all_exercises();
        $options['kos'] = $model->get_all_sports();

        if($_GET['recieve'] == 1)
        {

            $model->add_training();

            $options['result'] = 'Training sucessfully added!';
            $this->view->generate('view_add_training.php', 'view_skeleton.php',
            $data, $auth_opt, $options);
        }
        else if($_GET['recieve'] == 2)
        {
            $model->add_exercise();

            $options['result'] = 'Exercise sucessfully added!';
            $this->view->generate('view_add_training.php', 'view_skeleton.php',
            $data, $auth_opt, $options);
        }
        else
        {
            $this->view->generate('view_add_training.php', 'view_skeleton.php',
            $data, $auth_opt, $options);
        }

    }

    function action_add_exercise()
    {
        $authentication = new Authentication();
        $model = new Model_Training();

        $data = $model->get_all_sports();

        if($_GET['recieve'] == 1)
        {
            $model->add_exercise();
            $options['result'] = 'Exercise added!'; }

        $this->view->generate('view_add_exercise.php', 'view_skeleton.php', $data,
        $auth_opt, $options);
    }
}

```

controller_main.php

```

<?php
session_start();
require_once 'application/core/authentication.php';

class Controller_Main extends Controller
{

```

```

function action_index()
{
    $model = new Model_Main();
    $host = $_SERVER['HTTP_HOST'];

    if($_GET['logout'] == 1)
    {
        $authentication = new Authentication();
        $authentication->logout();
    }

    if($_GET['recieve'] == 1)
    {
        if($model->try_login($_POST['email'], $_POST['password']))
        {
            $authentication = new Authentication();
            $auth_opt = $authentication->get_auth_opt();

            echo '<script type="text/javascript">location.href =
"http://'.$host.'/Feed/feed";</script>';
        }
        else
        {
            $options['def_email'] = $_POST['email'];
            $options['result'] = 'Login error!';
            $this->view->generate('view_main_login.php',
'view_skeleton_main.php', $data, $auth_opt, $options);
        }
    }
    else
    {
        $this->view->generate('view_main_login.php', 'view_skeleton_main.php');
    }
}

function action_registration()
{
    $model = new Model_Main();
    $host = $_SERVER['HTTP_HOST'];

    if($_GET['recieve'] == 1)
    {
        $options['result'] = $model->try_registration();

        if($options['result'] == '')
        {
            $options['result'] = 'Вы успешно зарегистрированы, теперь вы можете
войти.';
            $this->view->generate('view_main_login.php',
'view_skeleton_main.php', $data, $auth_opt, $options);
        }
        else
        {
            $this->view->generate('view_main_registration.php',
'view_skeleton_main.php', $data, $auth_opt, $options);
        }
    }
}

```

```

    }
}
else
{
    $this->view->generate('view_main_registration.php',
'view_skeleton_main.php', $data, $auth_opt);
}
}
}

```

controller_people.php

```

<?php
session_start();
include 'application/core/authentication.php';

class Controller_People extends Controller
{

    function action_people()
    {
        $authentication = new Authentication();
        $auth_opt = $authentication->get_auth_opt();

        $model = new Model_People();

        if($_GET['recieve'] == 1)
        {

            $data = $model->search_people();
            $options['user_id'] = $_SESSION['user_id'];
            $options['subscribes'] = $model->get_subscribes();

            $this->view->generate('view_people.php', 'view_skeleton.php', $data,
$auth_opt, $options);
        }
        else
        {
            $this->view->generate('view_people.php', 'view_skeleton.php', $data,
$auth_opt, $options);
        }

    }
}

```

controller_message.php

```

<?php
session_start();
include 'application/core/authentication.php';

class Controller_Message extends Controller
{

    function action_message_list()
    {

```

```

        $authentication = new Authentication();
        $auth_opt = $authentication->get_auth_opt();

        $model = new Model_Message();
        $data = $model->get_message_list();

        $this->view->generate('view_message.php', 'view_skeleton.php', $data,
$auth_opt);
    }

    function action_message_story()
    {
        $authentication = new Authentication();
        $model = new Model_Message();

        if(isset($_GET['companion']))
        {
            if($_GET['recieve'] == 1)
            {
                $model->add_message($_GET['companion']);
            }

            $data = $model->get_message_story($_GET['companion']);
            $options['companion'] = $_GET['companion'];
            $this->view->generate('view_message_story.php', 'view_skeleton.php',
$data, $auth_opt, $options);
        }
        else
        {
        }
    }
}

```

controller_statistica.php

```

<?php
session_start();
include 'application/core/authentication.php';

class Controller_Statistica extends Controller
{
    function action_sport()
    {
        $authentication = new Authentication();
        $model = new Model_Statistica();

        if($_GET['recieve'] == 1)
            $data = $model->get_sport_statistic();

        $this->view->generate('view_statistica_sport.php', 'view_skeleton.php',
$data, $auth_opt, $options);
    }

    function action_exercise()
    {
        $authentication = new Authentication();
    }
}

```

```

        $model = new Model_Statistica();

        if($_GET['recieve'] == 1)
            $data = $model->get_exercise_statistic();
        else
            $data = $model->get_all_kos();

        $this->view->generate('view_statistica_exercise.php', 'view_skeleton.php',
        $data, $auth_opt, $options);
    }

    function action_people()
    {
        $authentication = new Authentication();
        $model = new Model_Statistica();

        $sort_dir = 'DESC';
        if(!isset($_GET['sort_by']))
            $sort_by = 'repetition';
        else
        {
            $sort_by = $_GET['sort_by'];
            if($_GET['sort_dir'] == 'ASC')
                $sort_dir = 'DESC';
            else
                $sort_dir = 'ASC';
        }

        $options['sort_dir'] = $sort_dir;
        $options['sort_by'] = $sort_by;

        if($_GET['recieve'] == 1)
            $data = $model->get_people_statistic($sort_by, $sort_dir);
        else
            $data = $model->get_all_koe(0);

        $this->view->generate('view_statistica_people.php', 'view_skeleton.php',
        $data, $auth_opt, $options);
    }

    function action_program()
    {
        $authentication = new Authentication();
        $model = new Model_Statistica();

        if($_GET['recieve'] == 1)
        {
            $data = $model->get_program_statistic();
            $options['has_program'] = $model->has_program();
        }

        $this->view->generate('view_statistica_program.php', 'view_skeleton.php',
        $data, $auth_opt, $options);
    }

```

```
function action_apply_program()  
{  
    $authentication = new Authentication();  
    $model = new Model_Statistica();  
  
    if(isset($_GET['program_id']))  
    {  
        $model->apply_program($_GET['program_id']);  
    }  
  
    echo '<script type="text/javascript">location.href =  
"http://'.$_SERVER['HTTP_HOST'].'/Training/training"</script>';  
}  
}
```