

ОПТИМІЗАЦІЯ ЗБЕРІГАННЯ ДАНИХ У WEB ЗАСТОСУНКАХ З ВИКОРИСТАННЯ МОДЕЛЕЙ DWH

М. Фостяк, Л. Демків

*Львівський національний університет імені Івана Франка,
Драгоманова, 50, 79005 Львів, Україна
lidia.demkiv@gmail.com*

Створено web-застосунок для взаємодії тренера та спортсмена з метою оптимізації тренувального процесу та аналізу спортивних результатів. При розробці використано прогресивні веб технології PWA для забезпечення роботи застосунку в оффлайн режимі та фреймворк React, який з повним пакетом і бібліотеками є найкращим варіантом для створення додатків PWA.

Запропоновано модель зберігання даних для отримання ефективної аналітики і статистики тренувань тренером та дослідження власних результатів спортсменом. Реалізовано модель сховища даних на стороні сервера для отримання тренером кількісних результатів тренувань та представлення їх у візуальній формі. Розроблена схема збереження даних на стороні клієнта з використанням IndexedDB та моделі сховища даних для побудови дашборду для спортсмена.

Ключові слова: pwa, React, моделі зберігання даних DWH, IndexedDB, аналіз даних, дашборд.

Вступ

З кожним днем обсяги даних, які генеруються компаніями, організаціями та користувачами, постійно зростають, і належне використання цих даних може призвести до численних переваг і можливостей, як для великих компаній так і для окремих користувачів програмних застосунків. В останні роки технології аналізу даних на особистісному рівні стали доступними кожній людині. Це стосується відслідковування трендів щоденного життя, як от спожитих калорій, тривалості сну, фізичної активності та ін. Впровадження аналітики даних у щоденне життя, допомога в процесі створення програми тренувань та аналізі її результатів, дозволяє спортсмену та тренеру моніторити параметри і досягнення, відслідковувати показники тренувань і тим самим вирішувати певний спектр проблем, пов'язаних удосконаленням процесу тренувань [1].

В роботі представлено веб-застосунок, який слугуватиме певним середовищем комунікації між спортсменами та тренерами та дозволить створювати та планувати тренування для різних груп чи окремих спортсменів, враховуючи їх індивідуальні особливості, а також відслідковувати прогрес виконання цих тренувань. Для розробки мобільного web-застосунку використано прогресивні веб-додатки (PWA) спільно з бібліотекою React. Поєднання PWA та бібліотеки React для реалізації розробки має ряд переваг і дозволяє створювати швидкі та інтерактивні додатки. Особливістю застосунку є використання методів трансформації даних для їх оптимального зберігання на сервері і

клієнти з метою побудови аналітичних звітів. Оптимальне зберігання даних для тренера реалізовано на серверній частині застосунку, а для спортсмена на клієнтській частині. Запропонована стратегія зберігання даних враховує особливості реалізації рwa застосунків та підходи до інженерії даних, а також є зручною для отримання аналітики даних, яка допомагає приймати правильні рішення та досягати кращих результатів.

Інтерфейс та функціональні можливості прогресивного web-застосунку

Створено веб-застосунок, який реалізує можливість комунікації між спортсменами та тренерами та дозволяє створювати і планувати тренування для різних груп чи окремих спортсменів, враховувати їх індивідуальні особливості, а також відслідковувати прогрес виконання цих тренувань. Функціонально застосунок дає можливість: реєстрації у додатку нових атлетів та тренерів, власний профіль із персональними даними для кожного із атлетів та тренерів, можливість створення груп із окремими атлетами, створення завдань для тренувань, створення на основі завдань повноцінних тренувань, можливість відслідковування для тренера процесу виконання тренувань окремих атлетів та груп, можливість призначати тренування для окремих груп та атлетів, створення таблиці із розкладом та типом тренувань кожного атлета, генерування на основі таблиці календаря із відмітками усіх запланованих тренувань та інші.

Для розробки веб застосунку використано прогресивні web технології та бібліотека React. Прогресивні web технології спільно з бібліотекою React є потужною комбінацією для створення мобільних проектів, які мають ряд переваг. PWA дозволяють створити додатки, які працюють як на веб, так і на мобільних платформах. Використовуючи React, можна перевикористовувати частину коду між веб і мобільними додатками, що спрощує розробку та знижує витрати. PWA не вимагають розміщення в магазинах додатків, що дає більше контролю над розповсюдженням та оновленнями застосунку. Користувачі можуть встановлювати PWA безпосередньо з браузера. PWA можуть працювати в офлайн-режимі або при обмеженому з'єднанні з Інтернетом завдяки можливостям кешування ресурсів. React може спільно працювати з Service Workers для забезпечення такої функціональності.

Проект розроблявся у середовищі Visual Studio Code, використовуючи наступні допоміжні розширення: Code Spell Checker, NPM Intellisense, Prettier – code formatter. Для створення інтерфейсу та його оновлення використано javascript бібліотека React. Також для реалізації конкретних задач застосунку використані й інші допоміжні бібліотеки: MUI – створення власних компонент інтерфейсу, Dayjs – обробка да форматування даних типу «дата», react-beautiful-dnd – створення динамічних елементів інтерфейсу, react-hook-form – створення та валідація форм, React-router-dom – маршрутизація всередині застосунку.

На рис. 1 показано фрагмент інтерфейсу користувача з можливістю створення групи для тренувань та тренування. В застосунку також реалізовані інтерфейси для профілів тренерів та спортсменів, таблиць з розкладом тренувань, таблиць із можливістю додавати нові тренування у розклад, створення тренувань для спортсмена з можливістю відмічати виконані завдання і залишати коментарі, з результатами виконання завдань спортсменом, сторінки із списками спортсменів за групами та інші.

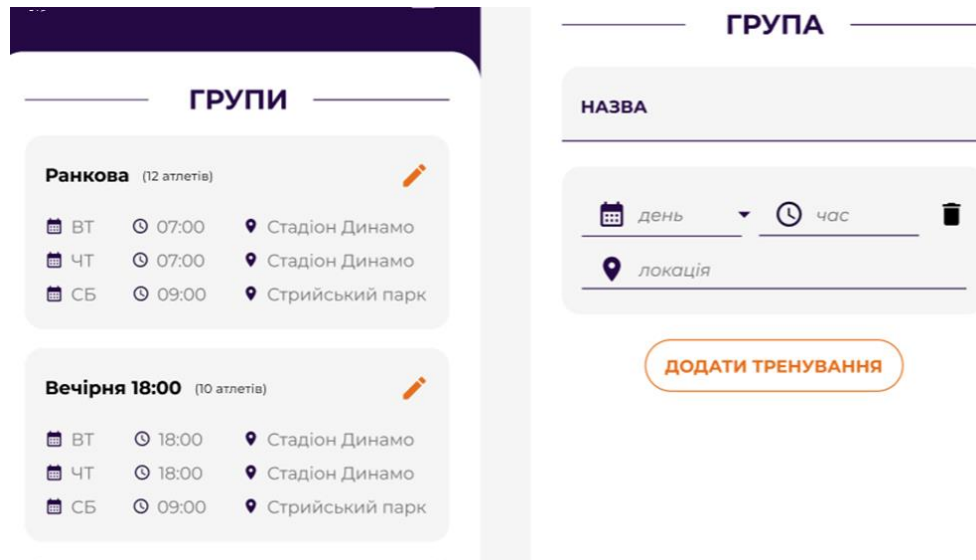


Рис. 1. Фрагмент інтерфейсів користувача з можливістю створення групи для тренувань та створення тренування.

Для створення такого інтерфейсу використана бібліотека React. Компоненти React є основними структурними блоками, які дозволяють розділяти інтерфейс на незалежні частини з метою спрощення розробки та підтримки. Кожен компонент може містити свою структуру, логіку та стиль, що допомагає створювати більш організовану та перевикористовувану кодову базу. Під час реалізації проекту створено біля сотні різноманітних компонент: функціональні компоненти, hooks, компоненти для маршрутизації. Наприклад, компонента ExerciseBadge.js відповідає за виконання вправ і може перебувати в трьох станах: виконана, не виконана, потрібно виконати. Використовуючи об'єднання різноманітних компонент побудовано модульний та легко підтримуваний інтерфейс для веб-застосунку.

База даних

Для зберігання даних використана реляційна база даних MySQL, яка забезпечує механізм зберігання та керування даними. На рис. 2 представлена частина бази даних, в якій зберігається інформація про тренування і результати спортсменів. Саме ці дані є цінними для аналізу даних, побудови дашбордів і отримання метрик, як індивідуальних для кожного атлета, так і загальних для цілої групи. Частина бази даних складається з 8 таблиць:

Members - таблиця зберігає інформацію, яка є спільною для всіх користувачів системи: тренерів і спортсменів. В ній зберігаються персональні дані: ім'я та прізвище, контактні дані та системна інформація.

Athletes - таблиця представляє атлета і має зв'язок з таблицею Members один-до-одного-або-нуль, оскільки кожен Athlete є Member, але не кожен Member є Athlete, тому що може бути тренером. В цій таблиці зберігаються дані атлета: результати, цілі, дані про здоров'я, тренувальну групу.

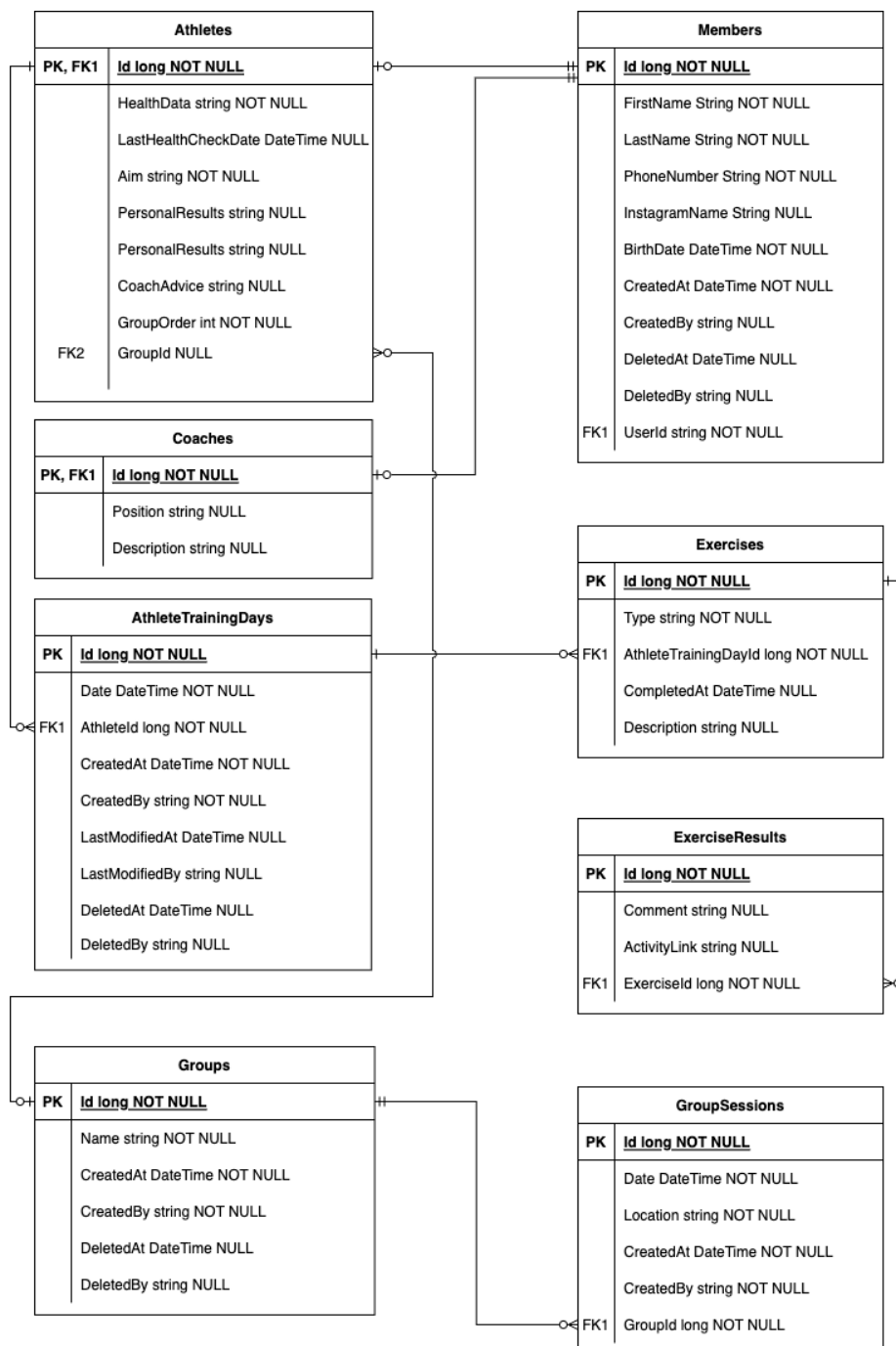


Рис.2. ER схема MySQL бази даних.

Coaches – таблиця зберігає дані про тренера. Так само, як і Athletes, таблиця має зв'язок один-до-одного-або-нуль з таблицею Members.

Groups – таблиця за якою можна визначити в якій групі тренується конкретний атлет. Ця таблиця містить назву групи і системні дані. Має зв'язок з Athletes один або нуль до багато або нуль, оскільки група може бути порожньою, або ж атлет може не бути в жодній з груп. Наприклад, коли він взяв паузу в тренуваннях.

GroupSessions – таблиця в якій зберігаються дані про групові тренування: час, місце та посилання на групу через foreign key GroupId. Зв'язок один до багатьох з Groups, тому що група є обов'язковою для тренування.

AthleteTrainingDays – таблиця, яка представляє тренувальний день спортсмена. Вправи для тренувального дня зберігаються в наступній таблиці Exercises і пов'язані між собою за допомогою foreign key AthleteTrainingDayId. Крім того, в таблиці AthleteTrainingDays є дата тренувального дня, та foreign key на атлета AthleteId. Зв'язок один до багатьох, оскільки один атлет має багато тренувальних днів.

Exercises – таблиця, в якій кожен рядок представляє вправу чи завдання, яке спортсмен має виконати в межах тренувального дня. В таблиці зберігаються завдання, їх опис з вказівками до виконання і дата, коли завдання було виконане - CompletedAt. Якщо завдання не виконане, то в CompletedAt значення NULL. Зв'язок з таблицею AthleteTrainingDays один до багатьох. Один тренувальний день має кілька вправ.

ExcerciseResults – таблиця з результати виконання вправи кожним атлетом, коментар та посилання на запис тренування в трекінг-додатку. Таблиця має foreign key на Exercises ExerciseId. Зв'язок один до одного або нуль, тому що якщо вправа не виконана, то не має запису в цій таблиці.

Таким чином в базі даних кожний спортсмен пов'язаний з групою, яка має розклад тренувань. Спортсмен має багато тренувальних днів у які може виконати різні завдання, зробити відмітку про те, що завдання виконано та залишити коментар. Для зберігання та доступу до спільної інформації спортсменом і тренером у базі даних використано наслідування. Інші таблиці бази даних не вказані на рис.2 з метою спрощення схеми для представлення на рис.2. Вони містять інформацію потрібну для авторизації, обліку абонементів, тощо.

Моделі сховищ даних

Щоденне швидке збільшення кількості даних у структурованих реляційних базах даних проектів утруднює побудову аналітичних звітів і тим самим сприяє появі нових архітектурних рішень для зберігання даних. Перехід від OLTP (Online Transaction Processing) до OLAP (Online Analytical Processing) має численні переваги для проектів, які хочуть зробити комплексний аналіз своїх даних та використовувати їх для прийняття обґрунтованих рішень та розробки ефективні стратегій.

Транзакційні бази даних мають високий рівень нормалізації, що сповільнює обробку запитів, які використовують декілька таблиць або їх з'єднань одночасно. OLAP-системи побудовані для оптимізованої роботи з аналітичними запитамі та великими обсягами даних. Вони забезпечують швидкий час відгуку на запити, незважаючи на обсяг та складність даних, дозволяють виконувати складні аналітичні запити, групування, агрегації, фільтрацію та розрахунки [2,3].

В рамках архітектури даних і аналітики, існують різні види сховищ даних, такі як DWH (Data Warehouse), Data Lake та Data Mart, які можуть включати OLAP (Online Analytical Processing) компоненти для здійснення аналізу даних. DWH є централізованим

сховищем даних, яке об'єднує дані з різних джерел і джерел зовнішніх систем. Дані у DWH зазвичай проходять процес ETL (Extract, Transform, Load), перетворюючись для аналітичного аналізу та звітування. Data Lake представляє собою сховище, де дані зберігаються у неструктурованому або напівструктурованому вигляді. В DataLake можуть включатися дані різних типів: структуровані, напівструктуровані та неструктуровані, включаючи тексти, логи, зображення, відео тощо. Дані в Data Lake можуть піддаватися ETL-перетворенням та очищенню, перш ніж використовуватися для аналізу. Data Mart є підмножиною DWH та представляє сховище даних, спеціалізоване для певної вибраної групи користувачів. В Data Mart включаються дані, необхідні для конкретних аналітичних потреб [4,5].

В роботі реалізована логіка переміщення даних з реляційної бази даних до двох сховищ даних побудованих за DWH моделлю. Одне сховище даних зберігається на сервері проекту і зберігає дані для тренера, інше сховище реалізовано на клієнтській стороні проекту для спортсмена. Сховища створені з метою оптимального, тривалого зберігання і використання даних процесу тренувань для побудови відповідних аналітичних звітів.

Властивості сховищ даних проекту, які побудовані за моделлю DWH: дані, що надходять у сховище, стають доступні тільки для читання, інформацію організовано відповідно до основних аспектів діяльності, дані в сховище надходять з бази даних (рис.2) і відповідно агрегуються, записи в DW ніколи не змінюються, являючи собою відбитки даних, зроблені у певний час, перед завантаженням у сховища дані фільтруються, зберігаються у певній послідовності, а також формується підсумкова інформація. В сховищах даних надмірність даних є мінімальною оскільки: при завантаженні у сховище дані сортуються і фільтруються; інформація у сховищах зберігається в хронологічному порядку, що майже повністю виключає перекриття даних; при завантаженні у сховище ані зводяться до єдиного формату, включаючи обчислення підсумкових (агрегованих) показників [6].

Побудови сховищ даних починаються з визначення питань на які сховище має відповісти. Серверне сховище даних побудовано для таких питань:

- Скільки учасників ходить в кожную групу?
- Як змінюється кількість учасників груп з часом?
- Кількість спортсменів, які були на тренуваннях в групах протягом певного часу?
- В яких групах більший відсоток виконаних вправ?
- Відсоток виконаних(не виконаних) вправ групою(спортсменом)?
- Відсоток виконаних(не виконаних) типів вправ групою спортсменів?
- Виконання яких вправ дає кращі спортивні результати?
- та інші.

Для локального сховища даних спортсмена сформульовано такі питання:

- Якою є періодичність відвідування занять?
- Який відсоток виконаних (не виконаних) вправ?
- Який відсоток виконаних (не виконаних) вправ по групах вправ?
- Як впливає виконання вправ на параметр (наприклад, дистанцію для бігу)?
- Як залежать результати від відвідування занять?
- Як залежать результати від відсотка виконаних вправ?
- та інші.

Трансформація даних для аналізу

ETL - це процес, який використано для переміщення та трансформації даних з бази даних web-застосунку до моделі Data Warehouse. Метою ETL є зробити дані придатними для аналізу, звітності та аналітики. На рис.3 показано загальний опис процесу ETL для трансформації даних з бази даних web-застосунку до моделі Data Warehouse:

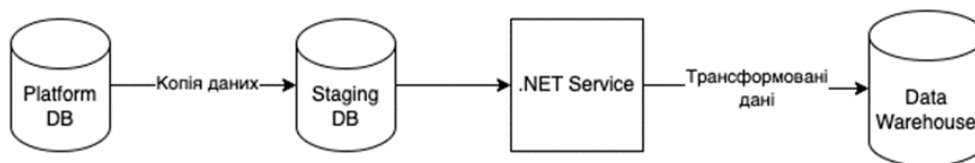


Рис. 3.Схема трансформації даних з бази даних у модель DWH.

На етапі вилучення дані отримуються з бази даних. Цей процес включає визначення відповідних таблиць, написання запитів, які формують нові дані для моделі. Отримані дані завантажуються в Staging environment, яка може бути окремою базою даних для полегшення трансформації даних.

Трансформація даних є важливим кроком, на якому необроблені вилучені дані очищаються та структуруються для того, щоб зробити їх придатними для аналізу. Трансформації включають: очищення, інтеграцію, об'єднання даних, агрегацію, конвертацію, валідацію тощо. На етапі трансформації вилучаються дані про неважливі типи завдань: розминка, заминка, розтягування, масаж. Відбувається агрегація даних за типами вправ та обчислюються відсотки виконаних вправ. Коментарі і параметри про досягнуті результати зчитуються і зберігаються в структурованому форматі: відстань, темп, частота серцевих скорочень.

Після того як дані трансформовані, вони завантажуються в модель сховища даних. Оскільки об'єми даних не є великими, то використовується повне завантаження. Окремий .NET сервіс, який виконується за розкладом, вичитує дані зі Staging database та завантажує в модель DWH.

Для даної моделі використовується схема зірки. Дана модель швидка у налаштуванні і побудові. Схема зірки є інтуїтивно зрозумілою та зручною для створення звітів. Більшість інструментів BI (бізнес аналізу) націлені на роботу з схемою зірки. За допомогою схеми зірки ми можемо розділити усі дані на факти, що містять кількісні дані про процес тренувань, і розмірності, які описують дані про факти та без яких факти не мають жодного значення. Серверна таблиця фактів проекту містить інформацію про тренера, спортсмена, дату, групу, відвідування групи, поля з назвами груп вправ, поля з відмітками про виконання вправ, поля з параметрами про результати спортсмена, а також допоміжні ETL поля про дату і особливості завантаження даних у сховище. Для ідентифікації кожного рядка таблиці фактів призначено первинний ключ для розпізнавання записів. У таблиці розмірностей міститься відносно невелика кількість записів у порівнянні з таблицею фактів, проте будь який з цих записів може описувати велике число даних з таблиці фактів. Таблиці розмірностей для тренера і спортсмена містять детальну інформацію про них і, відповідно, інші таблиці розмірностей: для груп, вправ, параметрів – містять детальну інформацію про них.

Прогресивні веб-застосунки (PWA) пропонують декілька варіантів локального зберігання даних на стороні клієнта для того, щоб користувачі могли продовжувати роботу, навіть якщо мережеве з'єднання стає нестабільним або переходить у режим офлайн. Способи, за допомогою яких PWA може зберігати дані на пристрої: локальне сховище, Cache API або IndexedDB. Індексована БД є оптимальним рішенням для зберігання даних у браузері: база даних документів JavaScript noSQL, яка може зберігати всі типи даних JS. В індексовану БД записана інформація для спортсмена, яка повторює у спрощеному виді модель сховища даних на сервері. У базу даних переписується лише та частина даних, яка стосується спортсмена.

Аналіз та статистика результатів тренувань є важливими як для спортсменів, які прагнуть покращити свої досягнення та досягти високих результатів у своїй дисципліні, так і для тренерів. Побудовані два сховища даних на стороні сервера та на стороні клієнта дозволяють писати більш прості запити для отримання аналітичних звітів ніж у випадку написання запитів до бази даних. Логіка приєднання схеми зірки, як правило, простіша, ніж логіка об'єднання, необхідна для отримання даних з високо нормалізованої транзакційної схеми бази даних. В результаті ETL перетворень даних і написання запитів відповідно до питань, які були сформульовані до сховищ даних (див.вище) отримуємо візуалізовані звіти у веб-застосунку написаному з використанням прогресивних web-технологій рис.4.

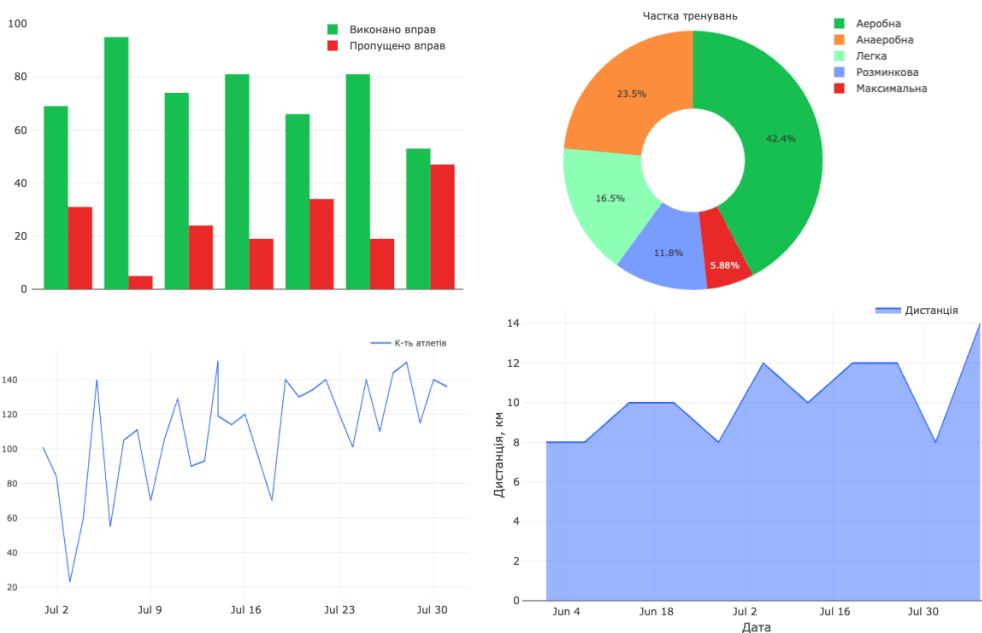


Рис. 4. Візуальне представлення даних для тренера і спортсмена.

На рис.4 зліва представлено частину аналітичного звіту для тренера: діаграма про кількість виконаних і не виконаних вправ спортсменом (на рис. зверху) та графік кількості спортсменів на занятті. На рис. 4 справа частина аналітичного звіту для спортсмена: колова діаграма та лінійний графік часової залежності параметра (дистанція).

Висновок

Великі обсяги даних, які накопичуються в компаніях, можуть бути використані для виявлення тенденцій розвитку компаній, розуміння поведінки клієнтів компаній, визначення сильних та слабких сторін бізнесу. Сховища даних дозволяють виконувати складні аналітичні запити та створювати звіти на основі цих даних. У великих компаніях є відділи аналізу даних та ВІ аналітики, які розробляють звіти, дашборди та панелі управління і допомагають керівництву та іншим зацікавленим сторонам отримати доступ до важливої інформації.

У роботі запропоновано використати модель сховища даних для зберігання даних у web-застосунку, який реалізує взаємодію спортсмена і тренера. Аналіз та статистика результатів тренувань мають велику важливість як для спортсменів, які прагнуть покращити свої досягнення та досягти високих результатів у своїй дисципліні, так і для тренерів. Для розробки веб застосунку використано прогресивні web технології та бібліотека React. Стратегія зберігання даних враховує особливості розробки рwa застосунків та моделі інженерії даних для побудови сховищ даних.

У застосунку використано методи трансформації даних за допомогою процесу ETL для їх змістовного зберігання на сервері і клієнті. На сервері за моделлю зірки DWH побудовано сховище для тренера. На клієнті з використанням IndexedDB створена модель сховища даних для спортсмена. Показано, що побудовані два сховища даних, дозволяють оптимально зберігати більшу кількість даних і писати більш прості запити для отримання аналітичних звітів ніж у випадку написання запитів до бази даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] *Ratten V.* Sport Data Analytics and Social Media: A Process of Digital Transformation, Sport Entrepreneurship, Emerald Publishing Limited, Bingley, 2020, p. 107-119. DOI: <https://doi.org/10.1108/978-1-83982-836-220201016>
- [2] *Novak M., Rabuzin K.* Prototype of web ETL tool *International // Journal of Advanced Computer Science and Applications* 5(6), 2014. DOI: [10.14569/IJACSA.2014.050614](https://doi.org/10.14569/IJACSA.2014.050614)
- [3] *EI_Sappagh S., Hendawi A., EI_Bastawissy A.* proposed model for data warehouse ETL processes // *Journal of King Saud University - Computer and Information Sciences* 23(2):91–104, 2011. DOI: [10.1016/j.jksuci.2011.05.005](https://doi.org/10.1016/j.jksuci.2011.05.005)
- [4] *Hanine M., Lachgar M., Elmahfoudi S., Boutkhoul O.* MDA Approach for Designing and Developing Data Warehouses: A Systematic Review & Proposal // *International journal of online and biomedical engineering* Vol.17, N 10, 2023. DOI: <https://doi.org/10.3991/ijoe.v17i10.24667>
- [5] *Dhaouadi A., Bousselmi K., Gammoudi M., Monnet S., Hammoudi S.* Data Warehousing Process Modeling from Classical Approaches to New Trends: Main Features and Comparisons// *Data* 2022, 7(8), 113. DOI: <https://doi.org/10.3390/data7080113>
- [6] *Souibgui M., Atigui F., Zammali S., Cherfi S., Yahia S.* Data quality in ETL process: A preliminary study // *Procedia Computer Science* Volume 159, 2019, P.676-687. <https://doi.org/10.1016/j.procs.2019.09.223>

DATA STORAGE OPTIMIZATION IN WEB APPLICATIONS USING DWH MODELS

M. Fostyak, L. Demkiv

*Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine
lidia.demkiv@gmail.com*

Large volumes of data that are accumulated in companies can be used to identify trends in the development of companies, understand the behavior of companies' customers, and determine business strengths and weaknesses. With the help of data warehouses, perform complex analytical queries and create reports based on this data. Large companies have data analysis and BI analytics departments to create reports, dashboards and dashboards and help management and other stakeholders gain access to critical information.

In the work, it is proposed to use a data storage model for data storage in a web application that implements the interaction between an athlete and a coach. Analysis and statistics of training results are important both for athletes who seek to improve their achievements and achieve high results in their discipline, and for coaches. Progressive web technologies and the React library were used to develop the web application. The data storage strategy takes into account the specifics of pwa application development and data engineering models for building data warehouses.

The application uses methods of data transformation using the ETL process for their meaningful storage on the server and client. At the stage of transformation, data on unimportant types of tasks are removed: warm-up, hitch, stretching, massage. Aggregation of data by types of exercises takes place and percentages of performed exercises are calculated. Comments and parameters about the achieved results are read and saved in a structured format: distance, pace, heart rate. After the data is transformed, it is loaded into the data warehouse model. A storage for the coach is built on the server based on the DWH star model. A data storage model for an athlete was created on the client using IndexedDB. Two data warehouses were built. These stores optimally store a larger amount of data and provide an opportunity to write simpler queries to obtain analytical reports than in the case of writing queries to the database. been

Keywords: pwa, React, DWH data storage models, IndexedDB, data analysis, dashboard.

Стаття надійшла до редакції 11.09.2023

Прийнята до друку 15.09.2023