



В. В. Кирий¹, О. В. Калиниченко², О. В. Ляпота³

¹ХНУРЕ, м. Харків, Україна, oleh.liapota@nure.ua

²ХНУРЕ, м. Харків, Україна, valentyna.kyryi@nure.ua

³ХНУРЕ, м. Харків, Україна, olga.kalynychenko@nure.ua

ОСНОВНІ ПРИНЦИПИ ПРОЕКТУВАННЯ ТА РОЗРОБКИ СИСТЕМ ГЕНЕРАЦІЇ ЗВІТНОСТІ МЕДИЧНИХ ЗАКЛАДІВ

Розглянуто базові принципи проектування та розробки систем генерації звітності медичних закладів. Знайдено проблеми та запропоновано рішення щодо побудови архітектури систем: використання розподілених сервісів та методів управління ними для покращення продуктивності, застосування черг повідомлень для спілкування між ними. Розглянуто загальні методи збереження даних та використання сховищ даних для спрощення схеми баз даних, що використовуються для ведення звітності медичних закладів. Запропоновано принципи та підходи для створення тих частин систем, що відповідають за кінцеву обробку даних та генерацію звітів з підтримкою різних форматів виведення документів.

АРХІТЕКТУРА СИСТЕМИ, ЗВІТ, МЕДИЧНИЙ ЗАКЛАД, СЕРВІС, СХОВИЩЕ ДАНИХ

V.V. Kyrii, O.V. Kalynychenko, O.V. Liapota. Basic principles of designing and developing reporting systems for medical facilities. The basic principles of designing and developing reporting systems for medical facilities are discussed. Problems are identified and solutions are proposed for building the architecture of systems: the use of distributed services and methods of managing them to improve performance, the use of message queues for communication between them. The general methods of data storage and the use of data warehouses to simplify the scheme of databases used for reporting by medical institutions are considered. The principles and approaches for creating those parts of the systems responsible for final data processing and report generation with support for various document output formats are proposed.

SYSTEM ARCHITECTURE, REPORT, MEDICAL FACILITY, SERVICE, DATA WAREHOUSE

Вступ

Згідно з вимогами сучасної медичної діяльності, необхідність в створенні ефективних систем генерації звітів, які б враховували специфіку роботи медичних закладів, стала першорядною. Збір, аналіз, управління та візуалізація даних – це ті основні завдання, які вирішують ці системи, що забезпечують оперативне та точне інформування персоналу медичних закладів, влади та пацієнтів.

Використання систем генерації звітності найбільш поширене у приватних медичних закладах. Галузь приватних медичних закладів активно розвивається, наприклад, протягом 2019 року розмір приватного сектору медичних закладів в Україні збільшився на 28%. Активний розвиток галузі призводить до актуальності розробки новітніх систем, що використовуються у рамках відповідної діяльності.

Принципи побудови систем генерації звітів полягають в розробці сучасної архітектури системи, що враховує основні операційні процеси роботи медичних закладів; створенні сховища даних, де зберігаються та обробляються великі обсяги інформації; розробці підсистеми генерації звітів, яка включає в себе аналіз даних та формування звітів на їх основі, і це є основною метою використання системи генерації звітності.

Метою даної роботи є обґрунтування використання кращих практик під час проектування та впровадження інформаційних систем генерації звітності, що дозволить медичним закладам більш ефективно використовувати доступні ресурси, поліпшувати

якість обслуговування та підвищувати свою конкурентоспроможність.

1. Принципи створення архітектури системи

1.1. Визначення вимог та принципів щодо створення архітектури системи

Архітектура системи генерації звітності медичних закладів має важливу мету – створити структуроване та логічне середовище обробки даних, яке ефективно використовуватиме ресурси та максимально спростуватиме процес генерації звітів. У світі, де кількість даних щодня зростає, правильно побудована архітектура може служити основою для довготривалої та стабільної роботи системи.

Важливість архітектури системи набуває ще більшої актуальності, коли ми говоримо про медичні заклади, де вимоги до точності та достовірності даних є особливо високими. Крім того, звіти таких установ часто мають значну складність та велику кількість сторінок, що вимагає високої продуктивності та оптимізації процесів на всіх рівнях системи.

Архітектура системи генерації звітності медичних закладів має забезпечувати гнучкість та масштабованість, що дозволить адаптуватися до змінюваних вимог та викликів. Вона повинна передбачити інтеграцію з різними джерелами даних, підтримку різних форматів звітів та мати можливість легкого покращення їхнього вигляду з можливістю підтримки стилів, що вказуються користувачами. Також важливим аспектом є обробка великих обсягів даних,

мінімізація помилок при цій обробці та швидкість генерації результуючих звітів.

Саме завдяки правильно побудованій архітектурі можливе ефективне використання комп'ютерних ресурсів, забезпечення швидкої та якісної генерації звітів, що можуть включати велику кількість даних та бути складними з точки зору структури та формату. Це вкрай важливо, адже в медичному середовищі час та точність інформації часто грають вирішальну роль.

Враховуючи це, можна висунути наступне твердження щодо важливості створення архітектури для системи генерації звітності: архітектура системи у такому випадку не лише технічна основа для оптимізації роботи з даними, вона є стратегічним інструментом, який допомагає медичним установам виконувати свої ключові задачі, покращувати ефективність роботи та забезпечувати високу якість послуг.

Тому для створення адекватної архітектури варто дотримуватись наступних принципів:

- виконувати ретельний аналіз потреб медичних закладів у цільовій країні;
- визначати вимоги до звітності;
- розглядати усі вірогідні джерела даних, що можуть бути запропоновані медичними закладами;
- враховувати обсяги та можливості обробки даних;
- брати до уваги динаміку розвитку медичної галузі на цільовому ринку та загальних технологій для роботи з даними.

Останній пункт особливо важливий для досягнення такого рівня системи, на якому вона змогла б без великих витрат адаптуватись до нових викликів і не втрачати своєї актуальності з часом.

1.2. Встановлення ключових компонентів системи

Системи, призначені для генерації звітності медичних закладів, загалом можна розділити на декілька окремих блоків, кожен з яких відповідає за окрему частину роботи:

- блок адміністрування: забезпечує контроль, налаштування та управління усією системою загалом; через цей блок можливе управління доступом до даних, налаштування внутрішніх параметрів генерації звітів, моніторинг стану системи та виконання нею усіх задач тощо;

- блок завантаження та первинної обробки даних: його завдання полягає в отриманні вхідних даних з різних джерел, їх валідація, фільтрація та первинна обробка; цей блок також забезпечує зберігання даних у необхідному для системи та зручному для подальшого використання форматі;

- блок генерації звітів: відповідає за обробку даних у відповідності до обраного типу звіту, що може включати агрегацію, аналіз даних та формування інформації для кінцевого звіту; цей блок використовує відповідні дані, отримані від попереднього блоку, та генерує фінальні звіти.

Кожен з трьох блоків може бути представлений у вигляді окремого модулю та розміщуватись на власному вузлі у системі (див. рис. 1), взаємодія з якими відбувається через будь-який користувацький інтерфейс, зокрема це може бути:

- графічний веб-інтерфейс;
- застосунок під будь-яку операційну систему стаціонарних пристроїв;
- застосунок під будь-яку операційну систему мобільних пристроїв;
- інтерфейс командного рядка.

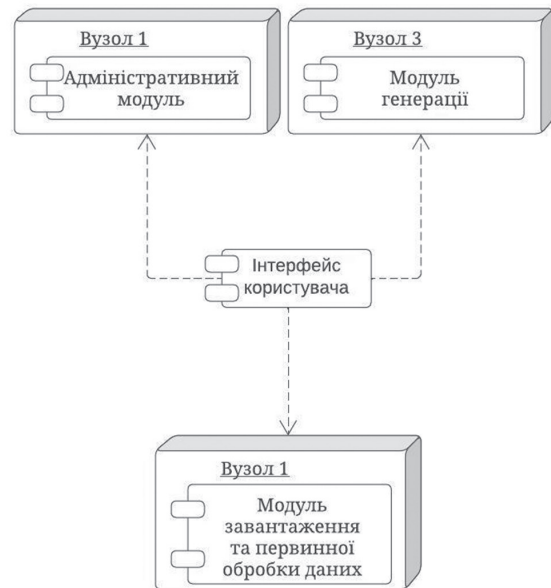


Рис. 1. Модулі на вузлах системи (2 рази вузол 1)

Завдяки використанню окремих вузлів для розміщення модулів системи: це можуть бути як повноцінні сервери, так і віртуальні машини або контейнери, – досягається їхня мінімальна зв'язність та можливість розширення кількості використовуваних вузлів за кожним окремим модулем (див. рис. 2).

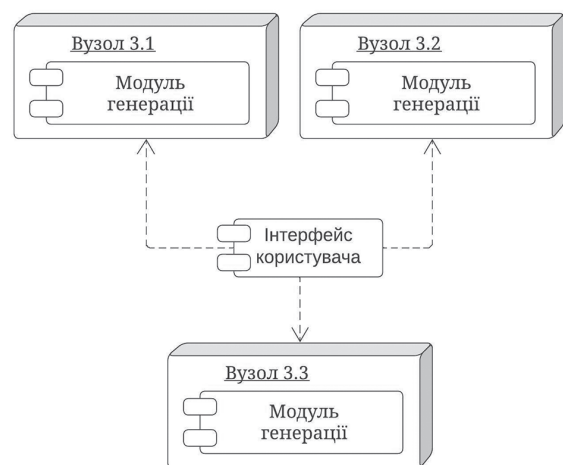


Рис. 2. Розширена кількість вузлів системи

Кожен з основних трьох модулів може складатись з власного набору сервісів, які також можуть розміщуватись на окремих вузлах. Зокрема, модуль генерації звітів може складатись з трьох частин:

- сервіс генерації та перевірки шаблонів для звітів;
- сервіс кінцевої обробки даних для звіту;
- сервіс генерації звітних документів.

Для зовнішньої взаємодії з цими частинами також створюється окремий програмний інтерфейс – Application Programming Interface (API).

Таким чином сервіс генерації звітів перетворюється на окремий повноцінний застосунок у рамках системи, сервіси якого також можуть існувати у декількох екземплярах (див. рис. 3) для покращення продуктивності та підтримки великої кількості користувачів та запитів до системи для генерації звітів.

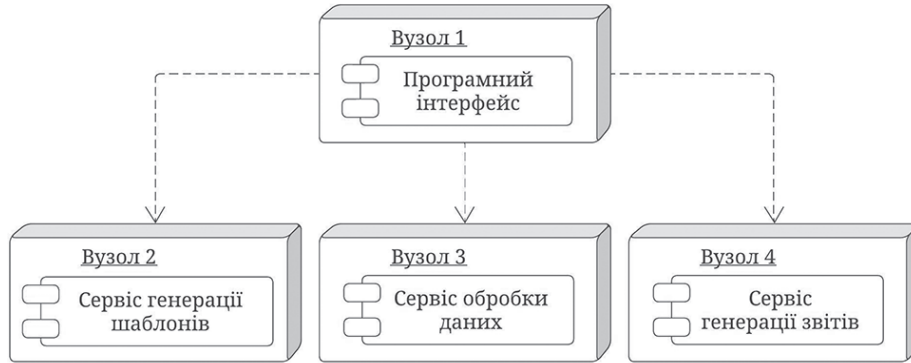


Рис. 3. Сервіси модулю генерації звітів

Таким чином досягається розділення системи на окремі сервіси, завдяки чому таку архітектуру можна вважати сервіс-орієнтованою – Service-Oriented Architecture (SOA) [1].

Наступні переваги [2] сервіс-орієнтованої архітектури реалізуються у системах генерації звітів медичних закладів:

- реалізується концепція слабо зв'язаних сервісів, що дозволяє з легкістю модифікувати або замінювати кожен з них;
- забезпечується економічна ефективність у випадку інтеграції вже існуючих відокремлених систем у нову;
- досягається зменшення бізнес-ризиків завдяки покращенню видимості окремих бізнес-операцій, що розподіляються між сервісами.

1.3. Визначення зв'язків між компонентами системи

Взаємодія між сервісами у такій системі, що складається в вище описаних компонентів, може виконуватись за певною низкою протоколів, кожен з яких має власні переваги та недоліки у порівнянні один з одним. Зокрема, найчастіше використовуються наступні:

- Hyper Text Transfer protocol (HTTP): базовий протокол, що використовується для клієнт-серверної взаємодії та виконується у форматі «запит-відповідь»;
- Simple Object Access Protocol (SOAP): протокол для доступу до об'єктів у мережі, що зазвичай використовується поверх HTTP протоколу, та на якому базується технологія веб-сервісів (веб-служб); являє собою XML структуру, що повинна відповідати усім вимогам кінцевого сервісу, до якого надсилається;
- Advanced Message Queuing Protocol (AMQP): протокол для передачі повідомлень між компонентами системи, заснований на ідеї використання брокера повідомлень як проміжної ланки між постачальником та споживачем повідомлень; за такої взаємодії

постачальник повідомлень відправляє їх брокеру на відомий відкритий API, брокер обробляє ці повідомлення та заносить їх у відповідну чергу, повідомлення у якій у свою чергу відстежує та обробляє споживач.

Саме останній із запропонованих протоколів більш доцільно використовувати у модулях, що потребують значного часу на обробку, зокрема з раніше наведених це модуль генерації звітів з його сервісами генерації шаблонів та, власне, звітів.

У такій системі брокером виступає програмне забезпечення RabbitMQ [3], яке реалізує взаємодію за протоколом AMQP.

У результаті отримуємо асинхронну генерацію кожної зі складових, що необхідні у процесі генерації документу зі звітом (див. рис. 4).

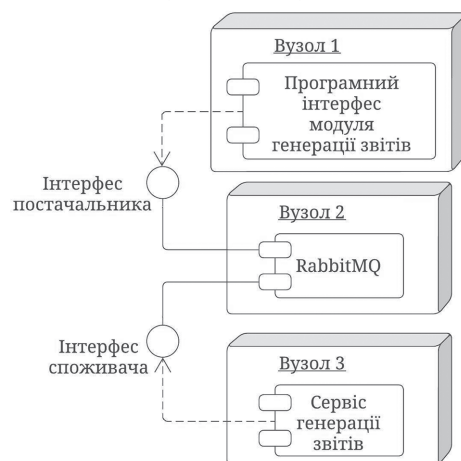


Рис. 4. Брокер повідомлень на схемі розгортання частини модулю генерації звітів

1.4. Можливості подальшого покращення та розвитку архітектури

Подальше покращення подібної архітектури можливе за рахунок продовження розділення модулів та

їхніх внутрішніх сервісів на менші сервіси за рахунок виділення окремих задач, які можуть вирішуватись самостійно в обмеженому контексті більш глобальних задач.

Так, модуль додавання та первинної обробки даних можна розділити на окремі сервіси додавання даних, кожен з яких буде підтримувати власне джерело даних, як от, наприклад, вручну введені користувачем данні, додані за допомогою завантаження файлів з даними або ж завантажені через відповідний програмний інтерфейс з іншого застосунку.

Такий розподіл допоможе отримати вже архітектуру, яку можна вважати мікросервісами. Та, відповідно, кожен з модулів загальної системи генерації звітності можливо представити у такому вигляді задля досягнення найменшої сфери застосування кожного з них з подальшою можливістю горизонтального розширення окремих сервісів, що мають найбільше навантаження та найчастіше використовуються користувачами під час роботи із завантаженням даних та генерацією аналітичних та статистичних звітів для покращення ефективності роботи медичних закладів.

2. Організація збереження даних

Основний принцип реалізації систем бізнес звітності, або звітності підприємства включає в себе процедури вилучення, перетворення та завантаження даних (ETL), сховище даних (data warehouse) для збереження даних та використання одного, або більше засобів генерації звітності.

2.1. Структура сховища даних

Структура сховища даних (див. рис. 5) містить кілька рівнів, кожен з яких відіграє свою роль.

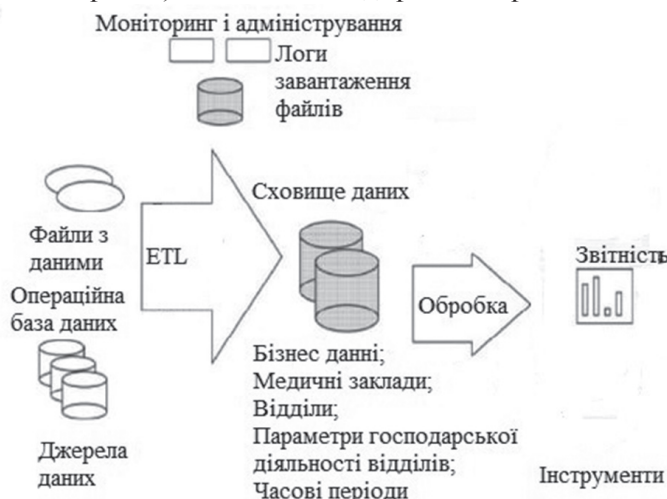


Рис. 5. Фундаментальна структура сховища даних

Робота сховища даних розпочинається, коли відкрито одне чи декілька зовнішніх джерел даних або операційних баз даних. Дані з джерел вилучаються, очищуються та трансформуються, щоб отримати структуру, яка відповідає структурі сховища даних. За допомогою інструменту завантаження зібрані дані імпортуються в сховище даних, після цього сховище

даних оновлюється за допомогою інструмента оновлення, як випливає з його назви [4].

Система генерації звітності медичних закладів передбачає роботу по заповненню сховища даних, під час процесу закриття часового періоду. Як джерело даних використовується низка файлів та операційна база даних, що містить бізнес дані, зібрані у рамках відповідного медичного закладу, та зв'язані з ними системні данні (інформація про медичні заклади, відділи, параметри господарської діяльності відділів, часові періоди).

На етапі вилучення з бази даних отримуються бізнес дані та інформація зі зв'язаних таблиць. На етапі очистки та трансформації зі зв'язаних таблиць отримуються тільки ті дані, що необхідні для суб'єкту, тобто будь-якого зовнішнього інструменту, що у подальшому оперує цими даними.

У системі генерації звітності медичних закладів зовнішній інструмент представляє собою інструмент генерації звітності, інструмент отримує доступ до даних після проходження усіх попередніх етапів.

Інструмент завантаження додає у сховище бізнес дані та системні дані у разі відсутності відповідних. У разі наявності бізнес даних у системі інструмент оновлення модифікує ті дані, що уже наявні у сховищі. Відповідність бізнес даних визначається по набору зв'язаних з ними системних даних.

У системі наявна база, в якій зберігається інформація про файли з даними, які представляють собою зовнішнє джерело, та логи завантажень файлів. Наведена база використовується для моніторингу та адміністрування.

2.2. Компоненти архітектури сховища даних

Архітектура сховища даних складається з:

а) сервісу завантаження, який відповідає за збір даних із функціональних структур. Він змінює конфігурацію даних так, щоб залежні компоненти могли їх використовувати. Сервіс містить загальні інструменти та додатки, які повинні накопичувати дані з функціональних структур, налаштувати їх і потім передавати в сховище даних. В системі генерації звітності медичних закладів наявно дві реалізації сервісу завантаження: першого та другого рівня:

1) сервіс завантаження першого рівня відповідає за збір структурованих даних з файлу. Сервіс зчитує дані згідно певної структури, так званої мапи, відповідно до типу мапи менеджер взаємодіє з одною чи декількома таблицями операційної бази даних, завантажуючи рядки, та додаючи зв'язки між таблицями;

2) сервіс завантаження другого рівня відповідає за збір даних з операційної бази даних. Сервіс працює у рамках процесу закриття часового періоду. У ході роботи менеджера, зчитуються дані з таблиць які відносяться до певного часового періоду, зі зчитаних даних обираються орієнтовані на суб'єкт (зовнішній інструмент генерації звітності) данні, які подаються на завантаження до сховища даних.

– сервіс сховища даних організовує дані так, що кожен користувач, або програмний модуль, міг легко проаналізувати їх або знайти необхідну інформацію. Сервіс реалізує підтримку уніфікації та незмінності даних. У системі генерації звітності медичних закладів такий сервіс працює в тандемі з сервісом завантаження другого рівня: отримані у результаті роботи сервісу завантаження дані організовуються у сутності сховища даних та завантажуються у відповідні таблиці;

– сервіс запитів є останньою точкою підключення. Він дозволяє кінцевим клієнтам отримати доступ до даних у сховищі даних [5]. Цей сервіс у рамках системи генерації звітності медичних закладів реалізується шляхом генерації та виконання SQL запитів.

2.3. Реалізація процедур вилучення перетворення та завантаження даних

Основним підходом до реалізації процедур вилучення, перетворення та завантаження даних представляє собою:

– вилучення даних: техніка, необхідна для збору даних. Джерела даних включають у себе прості файли (.txt, .csv), або реляційні системи управління базами даних. У рамках системи генерації звітності медичних закладів вилучення даних виконується шляхом зчитування з файлів рядків, розділення їх на окремі фрагменти за відповідним символом роздільника, та процедури зв'язування відповідних фрагментів з сутностями операційної бази даних згідно з мапою, у разі невідповідності даних мапі витяг не виконується, щоб уникнути втрати даних;

– трансформація даних: для цього кроку розглядається кожна значуща можливість зміни та узгодження даних. Це стандартизує формат розрізнених даних для їх консолідації та об'єднання. На етапі трансформації даних у рамках системи генерації звітності медичних закладів, до даних, отриманих у рамках вилучення додаються дані, що залежать від контексту завантаження (інформація про користувача, клієнта, медичний заклад, часовий період), отримані у результаті структури даних завантажуються в операційну базу даних;

– очищення даних: процес, який реалізує відповідність даних вимогам суб'єкту, що зберігаються в сховищах даних. Точність даних є важливою, оскільки вони використовуються в динамічних циклах обробки даних, таких як генерація звітності. У рамках системи генерації звітності медичних закладів очищення даних виконується після вилучення їх з операційної бази даних. З вилучених даних видаляється інформація, яка не є необхідною для роботи суб'єкту (зовнішнього інструменту генерації звітності), після проходження очистки заповнюються структури, які відповідають вимогам таблиць сховища даних;

– завантаження даних: після виконання стадій витягу, трансформації та очищення даних, вони поміщуються до сховища даних. У рамках системи

генерації звітності медичних закладів завантаження даних реалізується через виконання SQL запитів, до реляційної бази даних, у якій зберігається сховище даних;

– оновлення даних: інструмент, який реалізує оновлення сховища даних в будь-який момент часу. У рамках системи генерації звітності медичних закладів оновлення даних, виконується у разі повторного завантаження даних, при цьому нові дані не замінюють старі, а доповнюють їх.

2.4. Розробка схеми сховища даних

Структура конфігурації схеми сховища даних характеризується зв'язками між fact і dimension таблицями.

Схема формату зірки (див. рис. 6) представляє собою найпростіший варіант побудови схеми сховища даних. Схема формату зірки має найбільшу, відносно інших реалізацій схеми сховища даних, швидкість виконання запитів. Dimension таблиці містять властивості певного виміру, корисні для кінцевого користувача дані, які не являються числовими даними. Fact таблиця містить посилання на dimension таблиці, числове значення, на базі якого реалізується робота кінцевого користувача, та міститься в центрі конфігурації.

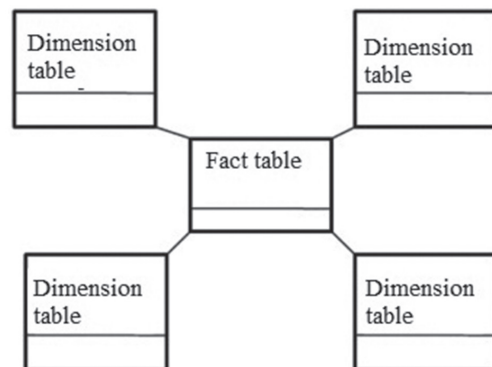


Рис. 6. Схема формату зірки

Для системи генерації звітності медичних закладів використовується сховище даних формату зірки (див. рис. 7).

У fact таблиці містяться посилання на dimension table та числові дані. У dimension таблиці department_instances знаходиться інформація про відділи: назва стандартного відділу, назва конкретного відділу, коротка назва відділу, назва cost center. У dimension таблиці facilities знаходиться інформація про медичні заклади: назва, аббревіатура, поштовий індекс, основна адреса, допоміжна адреса, місто, контактний телефон, контактна електронна пошта. У dimension таблиці time_periods наведено інформацію про часовий період: назву часового періоду, коротку назву часового періоду, дату початку, дату завершення, код типу (Q – квартальний, Y – річний). У dimension таблиці measures знаходиться інформація про міри: назву параметру господарської діяльності відділу.

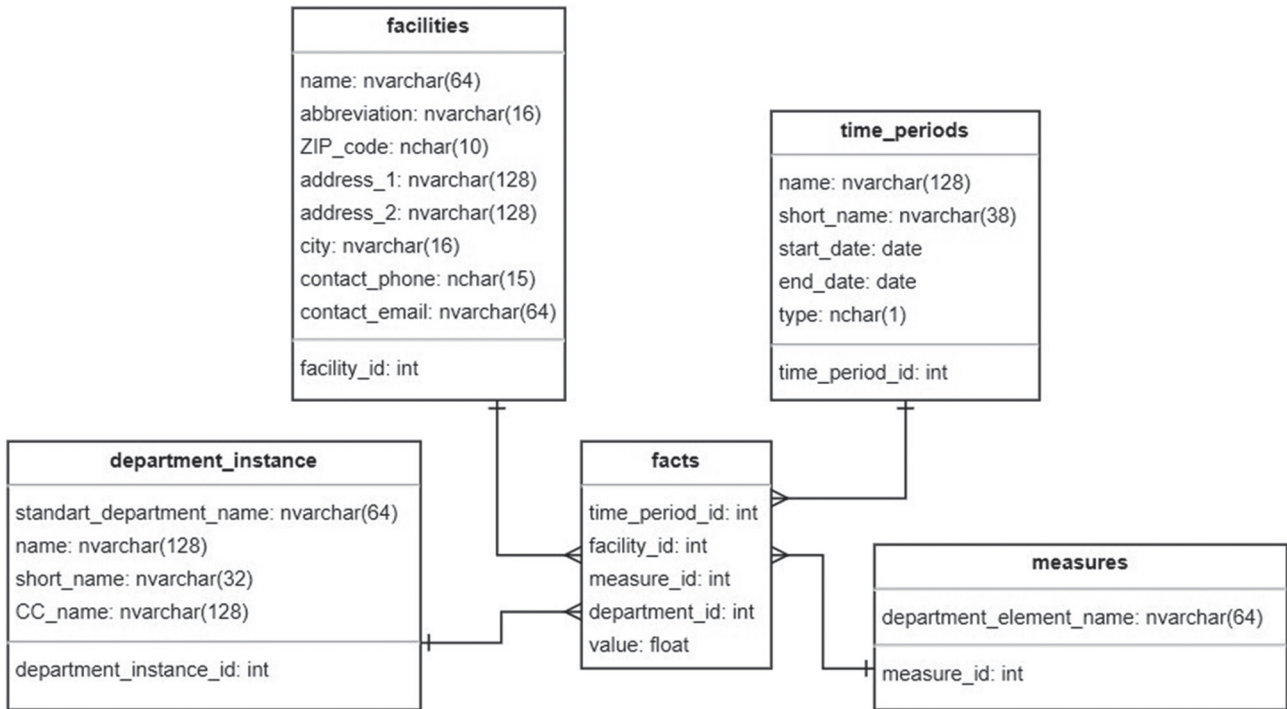


Рис. 7. Схема формату зірки для системи генерації звітності медичних закладів

Зовнішній інструмент генерації звітності може використовувати сховище даних для реалізації свого функціоналу.

3. Принципи обробки даних та генерації звітів

3.1. Принципи генерації звітів

Звіти мають враховувати особливості та потреби даних, особливо при обробці великої кількості інформації. Для зменшення обсягу даних, з якими працюємо, можна застосовувати фільтрацію за певними параметрами. Наприклад, якщо ми створюємо звіт з контактною інформацією медичних закладів, ми можемо використовувати фільтри для відбору лише тих закладів, які відповідають заданим критеріям, наприклад, за стандартним відділом або порівняльною групою. Це допомагає зосередитись на суттєвих даних, які відповідають нашим потребам у звіті.

Гнучкість також є важливим аспектом при обробці даних для генерації звітів. Генерація звітів повинна бути гнучкою, щоб користувачі могли настроювати аналіз даних відповідно до своїх потреб та інтересів. Це означає, що ми можемо використовувати вибіркові змінні для аналізу даних, що дозволяє користувачам настроювати аналіз відповідно до їхніх потреб та інтересів. Наприклад, ми можемо вибрати лише заклади з певного міста або з конкретними контактними телефонами або електронною адресою. Такий персоналізований підхід дозволяє отримати більш точні та релевантні результати аналізу, а також враховувати унікальні потреби користувачів при генерації звітів.

Важливою частиною процесу є візуальне представлення потоку даних, яке може бути подібним до використання тегів у системі автоматизованої генерації

звітів (Automated Report Generation System - ASGTR) [6]. Це допомагає проілюструвати, як саме відбувається обробка даних та їх потік з бази даних. Рисунок 8 надає візуальне уявлення про цей потік даних.



Рис. 8. Потік даних в системі генерації звітності медичних закладів

Спочатку аналізатор документу перевіряє кожен з підходів обробки даних на його необхідність, що задана попередньо користувачем. Це дозволяє врахувати особливі потреби користувача та забезпечити виконання лише необхідних підходів. Наприклад, якщо користувач не обрав фільтрацію даних, аналізатор пропускає цей підхід і переходить до наступного етапу обробки документу. Таким чином, користувач отримує інформацію лише згідно обраного шаблону для генерації звіту.

Дані, що оброблюються у процесі генерації звіту, можуть включати різні типи даних, залежно від конкретної задачі та потреб користувача. Основними типами даних, що можуть бути оброблені, це дані про різні відділи, які складаються з різних закладів. Інформація про окремі медичні заклади, включаючи назву, адресу, місто, поштовий індекс, контактні дані (телефон, електронна пошта) та інші релевантні деталі.

Також оброблювальними даними є дані про групу закладів, які використовуються для порівняння. Деякі різні показники, які використовуються для вимірювання та аналізу, такі як активи, зобов'язання, витрати, доходи, обсяги, години роботи тощо. Усі показники даних мають різний часовий період, тобто це розподіл даних за певний період часу, наприклад, за місяць, квартал, рік тощо.

Інші параметри фільтрації та вибору – користувач може обирати додаткові параметри для фільтрації та вибору даних, наприклад, стандартний відділ (якщо є), тип порівняльної групи тощо.

Одним з важливих методів аналізу даних в системі генерації звітів є використання перцентилів. Перцентиль [7] – це статистичний термін, що використовується для визначення значення, нижче якого певний відсоток спостережень знаходиться. Наприклад, перцентиль 75% вказує на значення, яке перевершує 75% спостережень, а 25% спостережень менше за нього.

Перцентилі можуть бути корисними при аналізі даних медичних закладів. Наприклад, можна використовувати перцентилі для визначення медіани (50% перцентиль), що вказує на середнє значення в розподілі даних. Також можна аналізувати верхні та нижні квартилі (25% та 75% перцентилі відповідно), що дозволяють оцінити розмах та розподіл даних.

Наприклад, при аналізі фінансових показників медичних закладів можна використовувати перцентилі для визначення верхньої та нижньої границі для доходів або витрат. Це дозволить ідентифікувати заклади, які виходять за межі звичайного розмаху та потенційно потребують додаткового аналізу.

Перцентилі також можна застосовувати до даних про часові періоди, щоб оцінити розподіл та тренди в часі. Наприклад, використання перцентилів дозволить ідентифікувати певні місяці або квартали, коли показники даних медичних закладів були особливо високими або низькими, що може свідчити про сезонні варіації або зміни тенденцій.

Таким чином, використання перцентилів у системі генерації звітів медичних закладів дозволяє отримувати більш детальний та об'єктивний аналіз даних, встановлювати розмах та розподіл показників, а також ідентифікувати варіації в часі та між різними закладами.

Шаблон документа є важливою складовою процесу генерації звітів. Він представляє собою набір

інструкцій та налаштувань, які використовуються для створення звіту. Шаблон визначає, які дані включати в звіт, який формат і структуру має мати звіт, а також які візуальні елементи, такі як графіки, діаграми та таблиці, використовувати для представлення даних.

Використання шаблонів має декілька переваг. Воно дозволяє стандартизувати формат звіту, що полегшує сприйняття та розуміння інформації користувачем. Крім того, шаблони допомагають забезпечити послідовність та консистентність у генерації звітів, особливо якщо використовуються у командній роботі або для повторюваних завдань.

Один з основних аспектів шаблону документа – це його візуальне оформлення. Графічна візуалізація даних, така як графіки, діаграми та таблиці, є важливим елементом звіту, оскільки допомагає легше сприймати та розуміти дані. Візуальні елементи допомагають встановлювати зв'язки між даними, виділяти тенденції та ілюструвати важливу інформацію. Використання графічної візуалізації сприяє ефективному аналізу результатів, полегшує прийняття рішень та покращує комунікацію з зацікавленими сторонами.

Збереження шаблонів та звітів у базі даних MongoDB є одним із популярних варіантів. MongoDB є нереляційною базою даних, яка зберігає дані у вигляді документів у форматі BSON (Binary JSON). Ця база даних використовує гнучку схему (див. рис. 9), що дозволяє зберігати різноманітні дані без необхідності дотримуватися фіксованої схеми таблиць, як у традиційних реляційних базах даних.



Рис. 9. Схема бази даних MongoDB

MongoDB [8] пропонує API для збереження та отримання документів з бази даних, яке може бути використано для збереження шаблонів документів та звітів. Кожен шаблон документа може бути представлений як окремий документ у колекції шаблонів, яка містить необхідну структуру та налаштування для генерації звіту. Аналогічно, створені звіти можуть бути збережені у колекції звітів з відповідними даними та візуальним оформленням.

Дана база даних також надає можливості для розширеного пошуку та запитів до бази даних, що полегшує виконання пошукових запитів для знаходження певних шаблонів чи звітів за різними критеріями.

При аналізі великої кількості даних важливо мати ефективний спосіб візуалізації результатів. Навіть якщо інформація є розрахунками або статистикою, просте надання її у вигляді чисел або тексту може бути складним для сприйняття та розуміння. Графічна візуалізація даних допомагає представити результати аналізу в зрозумілій та доступній формі. Вона використовує графіки, діаграми, таблиці та інші візуальні елементи, що допомагають ілюструвати тенденції, встановлювати зв'язки та виділяти важливу інформацію. Візуалізація даних сприяє легшому сприйняттю результатів аналізу, полегшує процес прийняття рішень та покращує комунікацію між зацікавленими сторонами.

Крім того, генерація звітів може включати використання попередньо визначених шаблонів або створення власних шаблонів. Це дозволяє включати візуальні елементи в звіт, такі як графіки, діаграми, таблиці та інші візуалізації даних. Використання шаблонів допомагає стандартизувати формат звіту і зробити його більш зрозумілим для користувачів. Крім того, візуальні елементи допомагають ілюструвати важливу інформацію та встановлювати зв'язки між даними. Застосування шаблонів та візуалізація даних сприяють ефективнішому аналізу, зрозумілішому сприйняттю та кращому використанню результатів звіту у прийнятті рішень [9].

Після завершення обробки всього документу та застосування фільтрації та очищення даних з бази даних за обраними користувачем параметрами, створений документ може бути збережений у різних форматах, таких як текстові файли (txt), документи PDF, розширені таблиці (Excel) або XML-файли. Це дає користувачеві можливість обрати найбільш зручний формат для подальшого використання або розповсюдження звіту.

Висновки

У роботі були розглянуті особливості роботи систем генерації звітності, зокрема пов'язаних з медичними закладами, та було виділено три основні складові, що є критичними для систем медичної звітності: створена архітектура, яка дозволяє з плином часу не втрачати актуальність з урахуванням сучасних темпів розвитку як медичної галузі, так і технологій загалом; методи ефективного збереження даних для їхнього подальшого використання у звітах; підходи до кінцевої обробки даних та генерації звітних документів.

У контексті побудови архітектури систем генерації звітності було розглянуто: розподіл таких систем на окремі компоненти та сервіси, що вибудовують сервіс-орієнтовану архітектуру; найкращі методи взаємодії між частинами системи, зокрема використання брокера повідомлень RabbitMQ, заснованого

на протоколі AMQP, який надає черги, у які одні сервіси додають завдання, а інші їх отримують та виконують у відповідності до поточної черги; також було вказано на можливість подальшого розділення окремих модулів системи на мікросервісні підсистеми.

Було розглянуто структуру сховища даних, та його компоненти, такі як сервіс завантаження, сервіс сховища даних, сервіс запитів; підходи до реалізації процедур вилучення, перетворення та завантаження даних, а саме витяг, очистка, трансформація, завантаження та оновлення даних; було розглянуто структуру сховища даних та обґрунтовано вибір структури.

Принципи обробки даних було розглянуто такі, що використовують візуалізацію даних: графіки, діаграми та таблиці, для представлення результатів аналізу у зрозумілій та доступній формі. Генерація звітів включає використання попередньо визначених шаблонів або створення власних, що допомагає стандартизувати формат звіту і полегшує сприйняття та використання результатів. Після обробки документа та генерації звіту, його можна зберегти у різних форматах для подальшого використання або розповсюдження.

Список літератури:

- [1] Faten H. An Overview of Service Composition in Service Oriented Architecture // Modern Applied Science. – 2018. – № 8(12). – P. 172–178.
- [2] Kunjal B. Mankad, Priti S. Sajja. Utilization of Web Services for Service Oriented Architecture // Journal of Global Research in Computer Science. – 2010. – № 1(1). – P. 20–23.
- [3] Amar Ćatović, Nevzudin Buzadija, Samir Lemeš. Microservice development using RabbitMQ message broker // Science, Engineering and Technology. – 2022. – № 1(2). – P. 30–37.
- [4] Chaudhari S., Dayal U. An overview of data warehousing and OLAP technology // ACM sigmoid record – 1997. – №. 26. – P. 65-64.
- [5] Bhatia P. Data mining and data warehousing: principles and practical techniques // Cambridge University Press. – 2019. – P. 388-404.
- [6] Gjorgjevikj, D., Madjarov, G., Chorbev, I., Angelovski, M., Georgiev, M., Dikovski, B. // ASGRT – Automated Report Generation System // ICT Innovations. Communications in Computer and Information Science – 2010. – №1(83) – P. 369-376
- [7] Hyndman, R. J., & Fan, Y. Sample quantiles in statistical packages // The American Statistician. – 2016. – № 50(4). – P. 361-365.
- [8] Shannon Bradshaw. MongoDB: The Definitive Guide // O'Reilly Media – 2013. – № 50(4). – P. 361-365
- [9] Jane Smith. An Overview of Service Composition in Service-Oriented Architecture // Modern Applied Science, – 2018. – №8(12). – P. 172-178

Надійшла до редколегії 21.06.2024