ДОСЛІДЖЕННЯ МЕТОДІВ ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ У СХОВИЩАХ ДАНИХ

УДК 004.65

DOI: https://doi.org/10.35546/2313-0687.2020.27.54-68

Раїса Захарченко,
к.т.н, доцент кафедри Програмних засобів і технологій,
Херсонський національний технічний університет, Херсон, Україна,
E-mail: zraissa2@gmail.com, ORCID 0000-0003-4650-3095

Леонід Захарченко,
аспірант кафедри Програмних засобів і технологій,
Херсонський національний технічний університет, Херсон, Україна,
ORCID: 0000-0001-9984-696X

Тетяна Кірюшатова,
к.т.н, доцент кафедри Програмних засобів і технологій,
Херсонський національний технічний університет, Херсон, Україна,
E-mail: tanyakir1963@gmail.com, ORCID: 0000-0002-0000-0065

Ігор Кибалко,
к.т.н., провідний інженер кафедри Програмних засобів і технологій,
Херсонський національний технічний університет, Херсон, Україна,
E-mail: kybalko.igor@kntu.net.ua, ORCID: 0000-0002-6634-5277

Анотація. Робота присвячена аналізу теоретичних та методологічних основ побудови ефективного сховища даних для найбільш раціонального збереження інформації.

Методи дослідження. В роботі використані такі методи наукових досліджень: експеримент, аналіз результатів діяльності. Із теоретичних методів дослідження використані: аналіз, синтез, порівняння.

Основні результати дослідження. Досліджено способи і методи збереження інформації у сховищах даних. Розглянуто основні типи програмно-апаратної архітектури сховищ даних, визначено їх переваги та недоліки. Результатом дослідження стала класифікація методів збереження інформації, яка допомагає проектувальнику створити сховище даних з найменшими витратами. Представлені основні компоненти інформаційного сховища даних, що дозволило побудувати типову структуру сховища даних. Проведено порівняльний аналіз методів моделювання сховищ даних, який дозволяє користувачу скористатися на практиці більш ефективним методом при проектуванні та розробці сховищ даних для збереження інформації в організаціях і на підприємствах [1].

Наукова новизна. Розвиток інформаційних ресурсів і засобів доступу до них, стрімкий розвиток україномовного контенту Інтернету є факторами, які змінили спосіб і підхід до збереження великих обсягів інформації. Нау-

ковою новизною цієї теми є використання сучасних сховищ даних для збереження інформації в вищих навчальних закладах освіти.

Ця тема є актуальною оскільки якісне збереження інформації впливає на організацію роботи не лише окремого підрозділу організації, а й усієї організації. Швидкість і миттєва можливість звернення до потрібної інформації визначає якість роботи користувачів в організації, непрямим чином впливає на якість і собівартість продукції, що випускається або послуги, які надаються.

Практична значимість. Збільшення, з кожним роком, обсягів інформації, яку потрібно зберігати і обробляти призвело до необхідності проектування та розробки сховищ даних. Ця тема є дуже актуальною для вивчення тепер, коли сховища даних усе активніше починають впроваджуватися у різних сферах людської діяльності, таких як освіта, сільське господарство, навчання, медицина, економіка, зв'язок, безпека охоронних систем, обробка інформації тощо. Проведена класифікація може стати основою для розширення кількості сфер використання сховищ даних. Результати дослідження можна використовувати у навчальному процесі для наочного представлення переваг та недоліків методів збереження інформації в сховищах даних, принципів їх використання у промисловій та не промисловій сферах.

Ключові слова: *багатовимірне моделювання, моделювання тимчасових даних, моделювання «звід даних», схема «зірка», схема «сніжинка», схеми з декількома таблицями фактів.*

Постановка проблеми. Необхідно проаналізувати та виявити методи проектування сховищ даних з метою виявлення найбільш ефективного методу зберігання даних. Дослідження здійснювалось такими кроками:

- Визначення архітектури даних – перший етап, коли проектувальник сховища даних визначає елементи даних, їхні властивості та взаємозв'язки між ними. Одним з ключових моментів побудови архітектури даних є ступінь деталізації інформації при перетворенні її в елементи даних. Для даних OLTP-систем вирішення питань, пов'язаних з рівнем деталізації даних, не є настільки важливим, як в системах збереження даних. В базах даних OLTP-систем дані зазвичай детально структуровані.
- Рівень структуризації – другий етап (деталізації або гранулювання) даних (Data granularity). Рівень структуризації даних – це ступінь деталізації даних, що зберігаються, оптимальна з точки зору вирішення інформаційно-аналітичних завдань в рамках предметної області сховищ даних.
- Розбиття всього набору даних на певні класи з метою подальшої деталізації всередині виділеного класу. Для сховищ даних характерні три основні види даних (класу).
- Виділення фактичних даних (Real-time data), які представляють собою поточний стан кількісних і якісних показників діяльності організації. Джерелом таких даних є зазвичай OLTP-системи. Таким даним притаманий високий рівень структуризації. Для того щоб використовувати такі дані в сховищах даних, їх потрібно попередньо обробити за допомогою процедур очищення.
- Визначення похідних даних (Derived data), які представляють собою дані, що отримані в результаті підсумовування, агрегації та усереднення фактичних даних. Залежно від завдань аналізу такі дані можуть бути або детальними, або підсумковими.
- Визначення консолідованих даних (Reconciled data) – фактичних даних, які були очищені та являють собою інтегроване джерело даних для вирішення задач аналізу. Основна вимога до таких даних – їх узгодженість (consistency).
- Вибір методу моделювання сховища даних дає абстрактну модель сховища даних, що проектується.

Аналіз останніх досліджень і публікацій. Вчені досліджували і досліджують різні методи збереження інформації в сховищах даних, але цього недостатньо і проблема потребує більш детального розгляду.

Серед сучасних вітчизняних науковців варто виділити Гаймакіна Н.А., Архіпенко С., Маклакова С.В. та інших. Так, зокрема, в сферу діяльності Гаймакіна Н.А. входять питання класифікації та використання інформаційних систем в різних сферах діяльності та використання баз і банків даних [1].

Архіпенко С. [5] займається дослідженнями концепції та впровадження сховищ даних у сфері обробки інформації. Маклаков С.В. [6, 7] у своїх дослідженнях розглядає напрямки проектування реляційних сховищ даних, переваги і недоліки генератору звітів. Туманов В.Е. [7] знайомить з математичними основами проектування реляційних сховищ даних.

Зарубіжні автори (Инмон Б. [2]) звертаються до проектування сховищ даних в залежності від їх типу, реалізації та аналізу сховищ даних, що втілює конкретні обчислювальні структури, необхідні для вирішення складних проблем

В праці Лоуенд Ш., Хилсон С., Хоббс Л. [3] коротко описані та обговорюються напрямки розробки і експлуатації сховищ даних.

Rainardi V. [4] описує основні об'єкти SQL Server 2008, наводить приклади використання цих обєктів при побудові сховищ даних.

У праці Спірлі Е. [8] досліджено принципи роботи корпоративних сховищ. Моделі планування, розробки і реалізації сховищ даних при розробці корпоративних інформаційних систем.

Мета дослідження. Метою дослідження є виявлення найбільш ефективного методу при збереженні інформації в сховищах даних. Проблема вибору програмного забезпечення, на якому буде побудовано сховище даних, є ключовою і цей вибір залежить від цілого ряду факторів: які вимоги пред'являються до сховища даних, які функціональні характеристики повинні бути, на яких користувачів орієнтоване сховище даних, а також, якими засобами володіє замовник для придбання та підтримки функціонування необхідного устаткування.

Виклад матеріалу дослідження. Сховище даних (англ. data warehouse) — предметно орієнтований, інтегрований, незмінний набір даних, що підтримує хронологію та здатний бути комплексним джерелом достовірної інформації для оперативного аналізу і прийняття рішень. Оперативні дані збираються з різних джерел, очищуються від непотрібного, інтегруються та складаються в реляційне сховище. При цьому вони вже доступні для аналізу за допомогою різних засобів побудови звітів. Потім дані (повністю або частково) готуються для OLAP-аналізу. Вони можуть бути завантажені в спеціальну базу даних OLAP або залишені в реляційному сховищі. Найважливішим його елементом є метадані, тобто інформація про структуру, розміщення та трансформацію даних. Завдяки ним забезпечується ефективна взаємодія різних компонентів сховища [2].

Компоненти, що входять в типове сховище, представлені на рис. 1.

Сховище даних складається з наступних компонентів [2]:

– ПЗ проміжного шару, що забезпечує мережевий доступ і доступ до баз даних. Сюди відносяться мережні та комунікаційні протоколи, драйвери, системи обміну повідомленнями тощо;

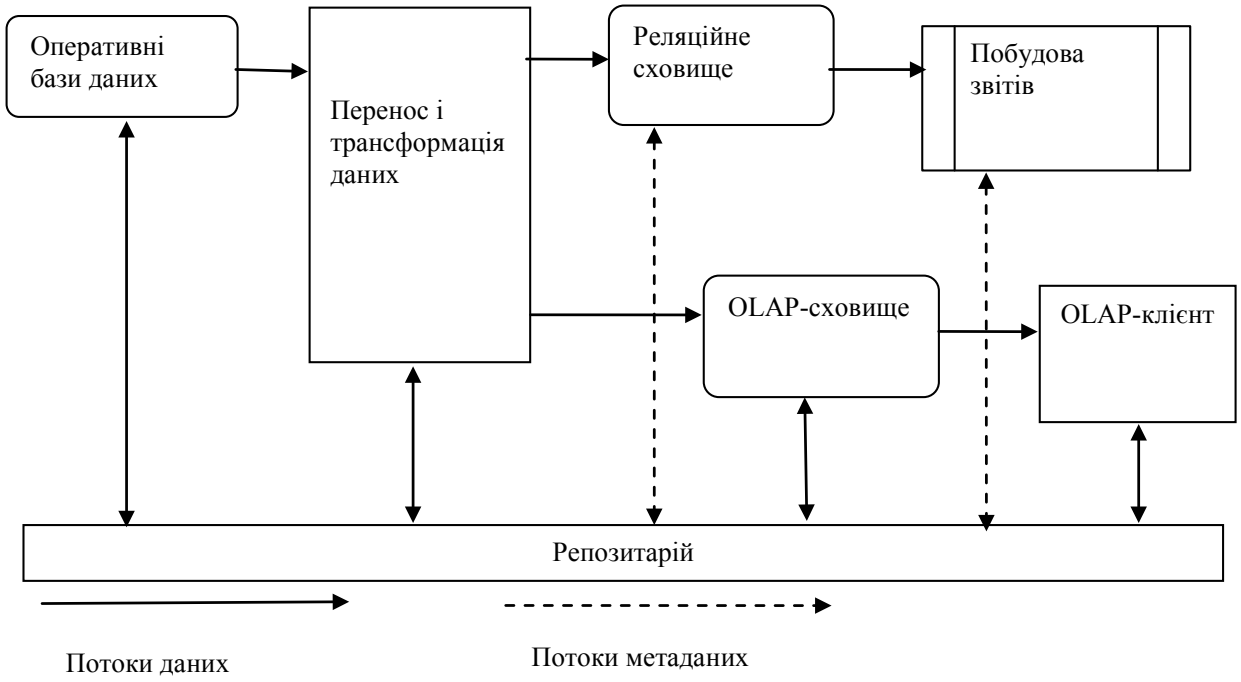

Рис. 1 – Структура сховища даних

– транзакційні БД і зовнішні джерела інформації використовуються у зв'язку з тим, що бази даних OLTP-систем історично призначалися для ефективної обробки структур даних у відносно невеликому числі чітко визначених транзакцій. Через обмеження цільової спрямованості «облікових» систем застосовувані в них структури даних погано підходять для систем підтримки прийняття рішень. Крім того, вік багатьох встановлених OLTP-систем досягає 10 – 15 років;

– рівень доступу до даних визначає програмне забезпечення, яке відноситься до цього рівня, забезпечує спілкування кінцевих користувачів з інформаційним сховищем і завантаження потрібних даних з транзакційних систем. Зараз універсальною мовою спілкування служить мова структурованих запитів (SQL);

– завантаження та попередня обробка передбачає набір засобів для завантаження даних з OLTP-систем та зовнішніх джерел. Виконується, як правило, у поєднанні з додатковою обробкою: перевіркою даних на чистоту, консолідацією, форматуванням, фільтрацією тощо;

– інформаційне сховище являє собою ядро всієї системи – один або декілька серверів БД;

– метадані (репозиторій, «дані про дані») відіграють роль довідника, що містить відомості про джерела первинних даних, алгоритми обробки, якими вихідні дані були оброблені тощо;

– рівень інформаційного доступу забезпечує безпосереднє спілкування користувача з даними OLAP за допомогою стандартних систем маніпулювання, аналізу і надання даних типу MS Excel, MS Access, Lotus 1-2-3 тощо;

– рівень керування (адміністрування) відслідковує виконання процедур, необхідних для поновлення інформаційного сховища чи підтримання його задовільного стану. Тут програмуються процедури підкачки даних, перебудови індексів, виконання підсумкових (підсумовуючих) розрахунків, реплікації даних, побудови звітів, формування повідомлень користувачам, контролю цілісності тощо.

Варіанти реалізації сховищ даних [2]:

– віртуальне сховище даних має в основі – репозиторій метаданих, які описують джерела інформації (бази даних транзакційних систем, зовнішні файли тощо), SQL-запити для їх зчитування та процедури обробки та надання інформації. Безпосередній доступ до останніх забезпечує програмне забезпечення проміжного шару. У цьому випадку надмірність даних нульова. Кінцеві користувачі фактично працюють з транзакційними системами безпосередньо зі всіма плюсами (доступ до «живих» даних в реальному часі) і мінусами (інтенсивний мережевий трафік, зниження продуктивності OLTP-систем та реальна загроза їх працездатності внаслідок невдалих дій користувачів-аналітиків);

– вітрини даних (Data Mart) за своїм визначенням – це набір тематично пов'язаних баз даних, які містять інформацію, що відноситься до окремих аспектів діяльності організації. По суті справи, вітрина даних – це полегшений варіант сховища даних, що містить лише тематично об'єднані дані. Цільова база даних максимально наближена до кінцевого користувача й може містити тематично орієнтовані агрегатні дані. Вітрина даних істотно менша за обсягом, ніж корпоративне сховище даних, і для його реалізації не потрібна особливо потужна обчислювальна техніка [2];

– глобальне сховище даних поєднує концепції сховища та вітрини даних в одній реалізації та використовує сховище даних в якості єдиного джерела інтегрованих даних для всіх вітрин даних. Тоді природною стає трирівнева архітектура системи.

На першому рівні реалізується корпоративне сховище даних на основі однієї з розвинених сучасних реляційних СКБД. Це сховище інтегрованих в основному деталізованих даних. Реляційні СКБД забезпечують ефективне збереження та управління даними дуже великого обсягу, але не дуже добре відповідають потребам OLAP-систем, зокрема, у зв'язку з вимогою багатовимірного представлення даних.

На другому рівні підтримуються вітрини даних на основі багатовимірної системи управління базами даних (прикладом такої системи є Oracle Express Server). Такі СКБД майже ідеально підходять для цілей розробки OLAP-систем, але поки що не дозволяють зберігати надвеликі обсяги даних. В цьому випадку це й не потрібно, оскільки мова йде про вітрини даних. Зауважимо, що вітрина даних не обов'язково повинна бути повністю сформована. Вона може містити посилання на сховище даних і добирати звідти інформацію по мірі надходження запитів. Звичайно, це дещо збільшує час відгуку, але натомість знімає проблему обмеженого обсягу багатовимірної бази даних [3, 4].

Нарешті, на третьому рівні знаходяться клієнтські робочі місця кінцевих користувачів, на яких встановлюються засоби оперативного аналізу даних.

При зберіганні даних у багатовимірних структурах виникає потенційна проблема «розбухання» за рахунок зберігання порожніх значень. Адже якщо в багатовимірному масиві зарезервовано місце під всі можливі комбінації міток вимірювань, а реально заповнена лише мала частина (наприклад, ряд продуктів використовується тільки в невеликому числі регіонів), тоді більша частина куба буде порожньою, хоча місце буде зайняте. Сучасні OLAP-продукти дозволяють справлятися з цією проблемою.

До сховищ даних застосовуються такі основні методологічні підходи [3]:

- «згори до низу» (Top down design);
- «знизу вгору» (Bottom down design);
- «з середини» (Middle of design).

На вибір підходу до реалізації сховища даних впливають такі чинники: стан поточної інформаційної інфраструктури організації, наявні ресурси; вимоги щодо повернення інвестицій; потреби організації в інтегрованому уявленні даних про свою діяльність; швидкість реалізації.

Вибір методологічного підходу до реалізації сховищ даних впливає на обсяг і ретельність проектування.

Підхід «згори до низу» вимагає детального планування та проектування сховища даних в рамках IT-проекту до початку виконання проекту. Це пов'язано з тим, що необхідно залучати всіх потенційних користувачів сховища даних для з'ясування їхніх інформаційних потреб в аналітичній обробці даних, приймати рішення про джерела даних, безпеку, структури даних, стандарти даних. Всі ці роботи повинні бути задокументовані та узгоджені. При цьому підході модель сховища даних повинна бути розроблена до початку реалізації. Зазвичай такий підхід практикують при створенні глобального сховища даних. Якщо кіоски даних включаються в конфігурацію, то вони можуть бути побудовані пізніше.

Перевагою такого підходу є отримання більш узгоджених визначень даних і бізнес-правил організації на самому початку роботи над створенням сховища даних. Вартість початкового планування та проектування може виявитися досить високою. Для цього підходу характерні великі витрати часу, що відкладає початок реалізації та затримує повернення інвестицій. Підхід «згори до низу» добре застосовувати в організаціях з чітко організованою інформаційно-обчислювальною структурою, коли програмно-апаратна платформа визначена й існують злагоджено працюючі джерела даних.

При використанні підходу «знизу вгору» починають з планування та проектування кіосків даних підрозділів без попередньої розробки глобальної інформаційно-обчислювальної інфраструктури організації. Це не означає, що така глобальна інфраструктура не буде розроблена пізніше. Такий підхід є більш прийнятним у багатьох випадках, оскільки він швидше призводить до кінцевих результатів. У нього є і недоліки: дані можуть дублюватися та бути неузгодженими в різних кіосках даних. Щоб уникнути цього, необхідно ретельне планування та проектування.

Підхід «проектування з середини» є комбінацією перерахованих вище підходів, які застосовуються наче по спіралі. Спочатку створюється ядро системи (підхід «згори до низу»), а потім воно поетапно нарощується за рахунок додавання нової або додаткової функціональності (підхід «знизу вгору»). Таким чином, на кожному витку спіралі може бути використаний кожен з двох зазначених вище підходів [24].

Існують й інші комбінації. Вибір підходу до реалізації сховища даних поряд з вибором архітектури сховища даних визначає тактичні рішення в проектуванні та управлінні проектом створення системи збереження даних. До таких рішень відносяться планування реалізацією та управління проектом.

Для логічного проектування реляційних сховищ даних застосовуються такі методи:

- Метод моделювання «сутність-зв'язок» (ER modeling) дає абстрактну модель предметної області, використовуючи в такому значенні: сутності (entities), взаємозв'язки (relationships) між сутностями і атрибути (attributes) для подання властивостей сутностей і взаємозв'язків.
- Метод багатовимірного моделювання (Dimensional modeling) дає абстрактну модель предметної області, використовуючи в такому значенні: показники або метрики (measures), факти (facts) і вимірювання (dimensions).
- Методи моделювання тимчасових даних (Temporal data modeling) дають абстрактну модель фра-

гмента предметної області, що представляє тимчасові ряди даних, і використовують у такому значенні: тимчасові мітки (timestamps), часовий ряд (time series), дата, діапазон дат, класи.

• Метод моделювання «звід даних» (Data Vault) дає абстрактну модель фрагмента предметної області, грунтуючись на математичних принципах нормалізації відносин, і використовує в такому значенні: сутність-концентратори (Hub Entities), що зв'язують суті (Link Entities), сутності-сателіти (Satellite Entities).

Результатом моделювання методом «сутність-зв'язок», або ER-моделювання, є ER-модель. ER-модель представляється за допомогою ER-діаграм, які є графічним представленням для абстрагування даних у вигляді сутностей, взаємозв'язків і атрибутів. Таким чином, семантика предметної області представляється в ER-моделі в термінах суб'єктивних засобів опису – сутностей, атрибутів, ідентифікаторів сутностей, супертипу, підтипів тощо.

Сутність предметної області є результатом абстрагування реального об'єкта шляхом виділення та фіксації набору його властивостей. Таким чином, сутність представляє клас об'єктів, який є результатом абстрагування реального об'єкта. Зазвичай вони позначаються іменником природної мови. Сутність описується за допомогою даних, іменованих властивостями або атрибутами (attributes) сутності. Як правило, атрибути є визначеннями у висловленні про сутність і позначаються іменниками природної мови. Сутності вступають в зв'язку один з одним через свої атрибути. Кожна група атрибутів, що описують один реальний прояв сутності, являє собою екземпляр сутності (instance). Іншими словами, примірник сутності – це реалізації сутності, що відрізняються одна від одної й допускають однозначну ідентифікацію. Іменування суті в однині полегшує в подальшому читання моделі. Фактично, ім'я сутності дається по імені її примірника.

Одним з основних комп'ютерних способів розпізнавання сутностей в інформаційних системах є присвоєння сутностям ідентифікаторів (Entity identifier). Часто ідентифікатор сутності називають ключем. Завдання вибору ідентифікатора суті є семантично суб'єктивним завданням. Оскільки сутність визначається набором своїх атрибутів, для кожної сутності доцільно виділити таку підмножину атрибутів, яка однозначно ідентифікує цю сутність. Деякі суті мають природні ідентифікатори. Наприклад, природним ідентифікатором рахунку-фактури є його номер. Ідентифікатори суті можуть бути складними – що складаються з декількох атрибутів, і атомарними – що складаються з одного атрибута сутності. Унікальний ідентифікатор сутності – це атрибут сутності, що дозволяє відрізняти одну сутність від іншої. Якщо сутність має кілька унікальних ідентифікаторів, так званих можливих ключів, то проектувальник повинен обрати первинний ключ сутності.

Розрізняють однозначні та багатозначні атрибути. Однозначними є атрибути, які в межах конкретного екземпляра сутності мають тільки одне значення. В іншому випадку вони вважаються багатозначними. Важливим моментом вивчення моделі предметної області проектувальником є виділення багатозначних атрибутів сутності. Це пов'язано з тим, що реляційна модель не підтримує багатозначних атрибутів і вони повинні бути дозволені на наступних стадіях проектування [5].

Кожен атрибут має домен (domain). Домен – це вираз, який визначає значення, дозволені для цього атрибута. Іншими словами, домен – це область значень атрибута. Для кожного атрибута сутності повинен бути визначений домен. На рівні логічного моделювання даних призначення домену атрибуту носить загальний характер. Наприклад, атрибут текстовий, числовий, бінарний, дата або «не визначений». В останньому випадку аналітик повинен дати опис домену. На наступних стадіях тип домена конкретизується, сенс поняття домену в фізичної моделі сховища даних за допомогою механізму обмеження домену, а СУБД не розуміє невизначених доменів.

Сутності не існують окремо одна від одної. Між ними є реальні відносини (Relationship), які повинні бути відображені в моделі предметної області. При виділенні відносин акцент робиться на фіксацію зв'язків та їхніх характеристик. Ставлення (зв'язок) являє собою з'єднання (взаємовідношення) між двома або більше сутностями. Кожен зв'язок реалізується через значення атрибутів сутностей. Зазвичай зв'язок позначається дієсловом. Кожний зв'язок також повинен мати свій унікальний ідентифікатор зв'язку [5].

У реляційної моделі відносини реалізуються тільки через обмеження цілісності по зовнішньому ключу. Вибір ключів сутностей – одне з найважливіших проектних рішень, яке належить зробити проектувальнику при переході до фізичної моделі бази даних.

Зв'язки характеризуються ступенем зв'язку і класом приналежності сутності до зв'язку. Ступінь (потужність) зв'язку – це відношення числа сутностей, що беруть участь в утворенні зв'язку. Наприклад, «один до одного», «один до багатьох», «багато до багатьох». На рівні логічної моделі допускається невизначений або невирішений зв'язок. Клас приналежності сутності – це характер участі сутності в зв'язку. Розрізняють обов'язкові та необов'язкові класи приналежності сутності до зв'язку. Обов'язковим є такий клас приналежності, коли екземпляри сутності беруть участь у встановленні зв'язку в обов'язковому порядку. В іншому випадку сутність належить до необов'язкового класу приналежності.

Відносини, що зв'язують сутність саму з собою, називаються рефлексивними. Типовим прикладом рефлексивних відносин є визначення структури підпорядкованості щодо «Співробітники». Рефлексивні відносини найчастіше відображають ієрархічні відносини всередині структури даних. З точки зору відносин виокремлюють слабкі сутності (weak). Слабкі суті – це сутності, які не можуть бути присутніми в базі даних, поки не існує пов'язаного з нею примірника іншої сутності. Прикладом такої сутності є замовлення, яке не може існувати без клієнта. Слабкі сутності мають обов'язковий клас приналежності, і ступінь зв'язку такої сутності не може дорівнювати нулю. Зв'язок «замовлення-клієнт» є обов'язковим.

Багатовимірне моделювання (Dimensional modeling) [5] простіше для розуміння, ніж ER-моделювання. Багатовимірне моделювання є методом моделювання та візуалізації даних як множини числових або лінгвістичних показників або параметрів (measures), які описують загальні аспекти діяльності організації. Як правило, при багатовимірному моделюванні основна увага фокусується на числових даних, таких як число продажів, баланс, прибуток, вага, або на об'єктах, які можна перерахувати, таких як статті, патенти, книги.

Багатовимірне моделювання має багато спільного з моделюванням методом «сутність-зв'язок» для реляційної моделі, але відрізняється цілями. Реляційна модель акцентується на цілісності та ефективності введення даних. Багатовимірна модель (Dimensional model) орієнтована в першу чергу на виконання складних запитів до БД [5].

Метод багатовимірного моделювання базується на таких основних поняттях: факти, атрибути, вимірювання, параметри (метрики), ієрархія, гранулювання.

Факт (fact) – це набір пов'язаних елементів даних, що містять метрики і описові дані. Кожен факт зазвичай представляє елемент даних, чисельно описує діяльність організації, бізнес-операцію або подію, яка може бути використана для аналізу діяльності організації або бізнес-процесів. В сховищах даних факти зберігаються в базових таблицях реляційної БД. Наприклад, вартість товару, кількість одиниць товару тощо.

Атрибут (Attribute) – це опис характеристики реального об'єкта предметної області. Як правило, атрибут містить заздалегідь відоме значення, що характеризує факт. Зазвичай атрибути представляються текстовими полями з дискретними значеннями. Наприклад, габарити упаковки товару, запах товару.

Вимірювання (dimension) – це інтерпретація факту з деякою точки зору в реальному світі. Вимірювання, подібно атрибутам, містять текстові значення, які сильно пов'язані за змістом між собою. Зазвичай вимірювання представляються як осі багатовимірного простору, точками якого є пов'язані з ними факти. У багатовимірної моделі кожен факт пов'язаний з однією або декількома осями. Вимірювання зазвичай представляють нечислові, лінгвістичні змінні, такі як філії організації, співробітники організації, покупці тощо.

Наприклад, при аналізі продажів продукції, виробленої або тієї, яка продається організацією, такими вимірами зазвичай вступають час, покупці, продавці, місце продажу або складування товару.

Вимірювання задаються перерахуванням своїх елементів (members). Елемент вимірювання (dimensional member) – унікальне ім'я або ідентифікатор (лінгвістична змінна), яка використовується для визначення позиції елемента. Наприклад, вимір «Час» може містити такі елементи: «всі місяці», «квартали», «роки».

Часто елементи вимірювання перебувають у відношенні «частина-ціле» або «батько-нащадок», що дозволяє ввести на вимірі одну або кілька ієрархій. Кожна ієрархія може мати кілька рівнів ієрархії (hierarchy levels). Кожен елемент вимірювання повинен належати тільки одному рівню ієрархії, породжуючи таким чином розбиття на підмножини. Прикладом може служити ієрархія на вимірі «Час»: рік, півріччя, квартали, місяці та дні. Елемент вимірювання «тиждень» може

належати двом місяцям, тому для нього слід визначити іншу ієрархію.

Параметр, метрика або показник (measure) – це числова характеристика факту, який визначає ефективність діяльності або бізнес-дії організації з точки зору вимірювання. Як правило, метрика містить заздалегідь невідоме значення характеристики факту. Конкретні значення метрики описуються за допомогою змінних.

Гранулювання (Granularity) – це рівень деталізації даних, що зберігаються в сховищі даних. Наприклад, щоденні обсяги продажів.

З точки зору взаємозв'язку вимірів і фактів останні можна розбити на такі класи [6]:

- адитивні факти (Additive facts). Це факти, які має сенс використовувати з будь-якими вимірами для виконання операцій додавання з метою отримання будь-якого значимого результату. Наприклад, дискретні числові показники активності діяльності, такі як кількість продажів, обсяг продажів тощо;
- напівадитівні факти (Semiadditive facts). Це факти, які має сенс використовувати спільно з деякими вимірами для виконання операцій додавання з метою отримання будь-якого значимого результату. Наприклад, числові показники інтенсивності, такі як залишок на рахунку, рівень запасів на складі тощо;
- неадитивні факти (Non-additive facts). Це факти, які має сенс використовувати спільно з будь-яким виміром для виконання операцій додавання з метою отримання будь-якого значимого результату. Наприклад, вимір кімнатної температури;
- числові виміри інтенсивності (Numerical Measures of Intensity). Це факти, що є неаддитивними за часом, допускають агрегацію та підсумовування по деякому числу тимчасових періодів. Наприклад, залишок на рахунку.

Існують кілька схем для багатовимірного моделювання даних. Дві з них вважаються основними: схема «зірка» (star schema) і схема «сніжинка» (snowflake schema). У складніших випадках використовуються так звані «багатозіркові» схеми або схема з декількома таблицями фактів [7].

Схема «зірка» має одну таблицю фактів і кілька таблиць вимірів. Таблиці вимірювань є денормалізованими. Схема «сніжинка» має одну таблицю фактів і кілька нормалізованих таблиць вимірів. Схема «зірка» має ті ж самі елементи, що діаграми «сутність-зв'язок». Це – сутності, атрибути, первинні та зовнішні ключі, взаємозв'язки, кардинальність зв'язку.

Метод моделювання «Зведення даних» (Data Vault), за визначенням, є орієнтованим на деталі набором нормалізованих пов'язаних таблиць, які забезпечують інформаційну підтримку однієї або більше предметних областей діяльності організації. Цей підхід є комбінацією методики реляційного проектування (до третьої нормальної форми – 3NF) і методики багатовимірного проектування. Метод моделювання «Зведення даних» був розроблений для створення моделей даних глобальних сховищ даних масштабу підприємства. Він заснований на математичних принципах, які підтримують нормалізовані моделі даних. По суті модель «Зведення даних» відповідає нормалізованої до 3NF схемою «зірка», включаючи вимірювання, зв'язку «багато до багатьох» і таблиці стандартної структури. Різниця в детальнішому поданні взаємозв'язків і елементів даних, структурованих і деталізованих у тимчасову зміну. Цей метод проектування був розроблений, щоб об'єднати гнучкість структур обробки даних OLTP-систем з потужністю аналітичної обробки даних в OLAP-системах. Він є масштабованим і легко адаптованим методом розробки структур даних для вирішення задач аналізу даних в масштабах підприємства [8].

Модель проектування «Зведення даних», аналогічно методам багатовимірного моделювання або «сутність-зв'язок», містить ряд структурних компонент, новими з яких є сутності-концентратори, або хаби, сутності-зв'язки і сутності-сателіти. Проектування цим методом фокусується на функціональних предметних областях діяльності організації. Кожна така область характеризується бізнес-ключем і представляється в концентраторі первинним ключем. Сутності-зв'язки забезпечують інтеграцію операцій між хабами. Сутності-сателіти забезпечують контекст первинного ключа хаба. Кожна з цих сутностей сконструйована для забезпечення максимальної гнучкості та масштабованості моделі даних сховища даних масштабу підприємства.

Сутності-концентратори (Hub Entities), або просто хаби (hubs), є таблицею, яка містить мінімальний список бізнес-ключів (натуральних ключів). Це ключі, які використовуються організацією в кожній щоденній операції: наприклад, номер рахунку, табельний

номер співробітника, номер покупця, номер виробу і номер автомобіля. Якщо в процесі діяльності такий ключ був втрачений, то, як правило, губляться і посилання на контекст, і супутня інформація. Сутності-концентратори не можуть бути пов'язані відношенням «один до багатьох» (батько-нащадок). Для побудови взаємозв'язків між концентраторами використовуються суті-зв'язки. Сутність-зв'язок (Link Entitiy) є фізичним представленням взаємозв'язку «багато до багатьох» в 3NF. Зв'язок є взаємовідношеннь або операцією між двома або більше бізнес-компонентами або бізнес-ключами.

Сутності-сателіти (Satellite Entities) містять описову інформацію про ключі концентраторів, а саме коли, чому, що, де і хто створює операції та бізнес-ключі. Наприклад, на відміну від номера автомобіля, його колір, марка тощо, можуть змінюватися в часі, а, отже, структура даних повинна відображати ці зміни на кожному рівні структурування інформації (гранулювання) [8] .

Сутність-міст (Bridge) містить тимчасові мітки останнього завантаження. Ця сутність подібна суті «Момент часу», але охоплює всю предметну область або схему даних [8].

Бізнес-користувачі часто хочуть бачити дані, згруповані по-різному. У найпростішому випадку один підрозділ (наприклад, відділ маркетингу) має свою ієрархію покупців, а інший підрозділ (наприклад, відділ продажів) має іншу ієрархію тих же покупців. Можна включити обидві ієрархії в вимір «Покупець». Однак кілька ієрархій, вбудованих прямо у вимір, зроблять його малопридатним для використання.

Бізнес-вимога більш гнучкої реалізації додаткових ієрархій виникає тоді, коли кільком підрозділам необхідно групувати ті ж дані за різними схемами класифікації в кількох різних варіантах. В такому випадку необхідно попрацювати з користувачами і визначити найбільш поширене угрупування даних. Це угрупування стане стандартною ієрархією, яка використовується за замовчанням, і буде вбудована прямо в основну таблицю виміру. Так само можна вчинити ще з кількома найбільш широко використовуваними ієрархіями для простоти роботи користувачів.

Для підтримки додаткових ієрархій в сховищах даних створюється окрема таблиця, за допомогою якої користувач може згрупувати дані по кожній із наявних ієрархій. Це і є сутність-міст, або таблиця-міст (bridge table) [8] .

Для спрощення аналізу і створення звітів проміжна таблиця повинна містити опис та стандартну ієрархію. Стандартна ієрархія стає використовуваною за замовчуванням в усіх попередньо налаштованих звітах, але користувачеві надається можливість переключитися на іншу ієрархію.

Прикладом використання сховищ даних є університетський репозитарій ХНТУ, який створений з використанням програми DSpace.

Проект DSpace створить у світі можливість вибору програмного забезпечення для сховищ, що забезпечує засоби для відкритої інформації та простоти управління.

DSpace – програмне забезпечення, яке обирають академічні, некомерційні та комерційні організації, що створюють відкриті цифрові сховища. Його можна безкоштовно та легко встановити “з коробки” та повністю налаштувати відповідно до потреб будь-якої організації. DSpace зберігає та забезпечує легкий та відкритий доступ до всіх типів цифрового вмісту, включаючи текст, зображення, рухомі зображення, mpeg та набори даних. Завдячуючи постійно зростаючому співтовариству розробників, що віддані постійному розширенню та вдосконаленню програмного забезпечення, кожна нова версія DSpace перевершує попередню.

DSpace є у вільному доступі як програмне забезпечення з відкритим кодом.

Репозитарій знаходиться в розділі «Мережеві ресурси» сайту ХНТУ (рис. 2)

Репозитарій ХНТУ має декілька розділів, перелік яких представлений на рис. 3.

Інформацію в репозитарії ХНТУ можна переглядати в такому вигляді: за факультетами, за авторами, за предметом, за роком створення (рис. 4).

На кожному факультеті інформацію можна переглядати по кафедрам та авторам (рис. 5).

По кожній кафедрі можна вносити інформацію різного напрямку: матеріали конференцій, монографії, навчальні видання, статті (рис. 6).

Крім того є можливість збереження кваліфікаційних робіт бакалавра та магістра в інстутиційному репозитарії ХНТУ та перегляду цієї інформації за спеціальністю, за автором, за предметом дослідження або за роком створення (рис. 7).

Рис. 2 – Доступ до інстутиційного репозитарію ХНТУ

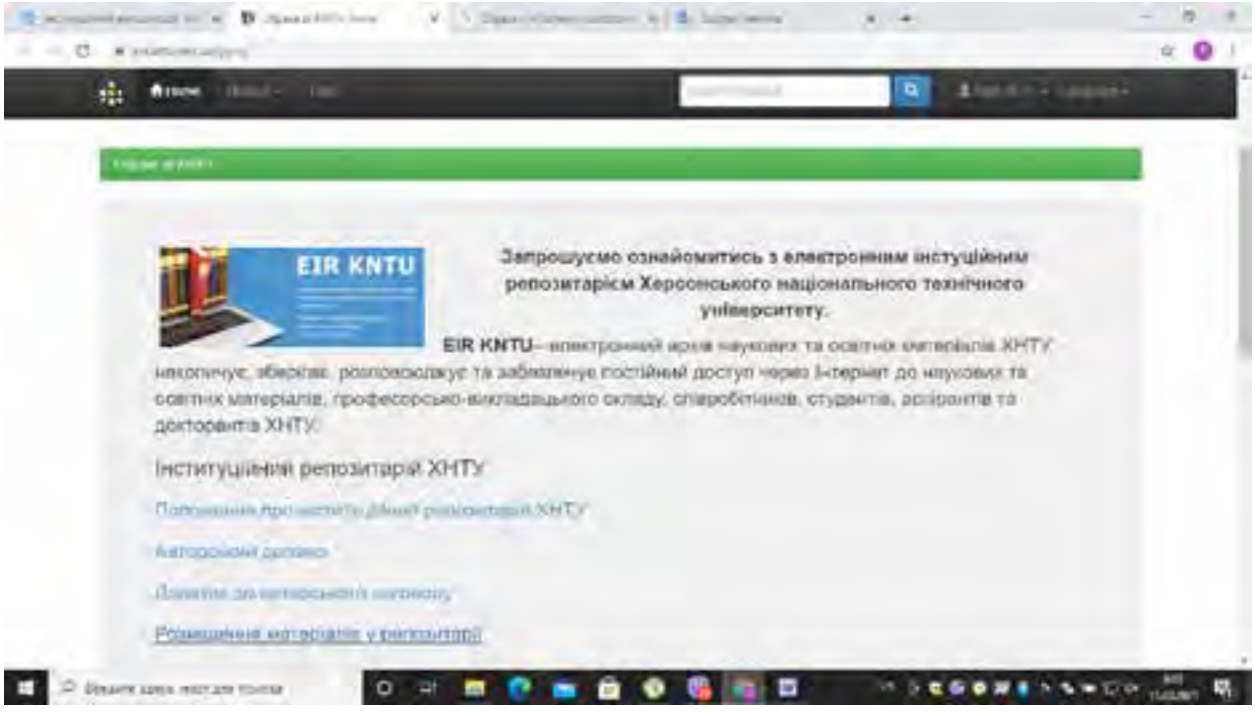

Рис. 3 – Розділи інстутиційного репозитарію ХНТУ

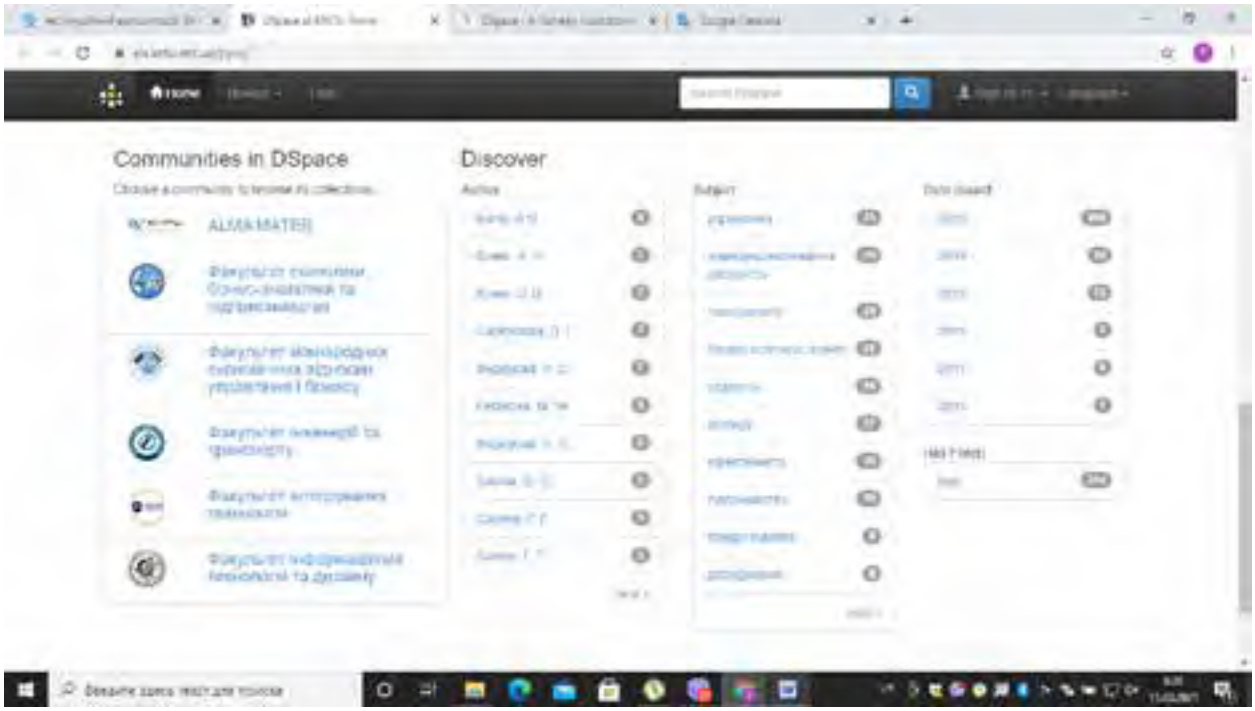

Рис. 4 – Види перегляду інстутиційного репозитарію ХНТУ

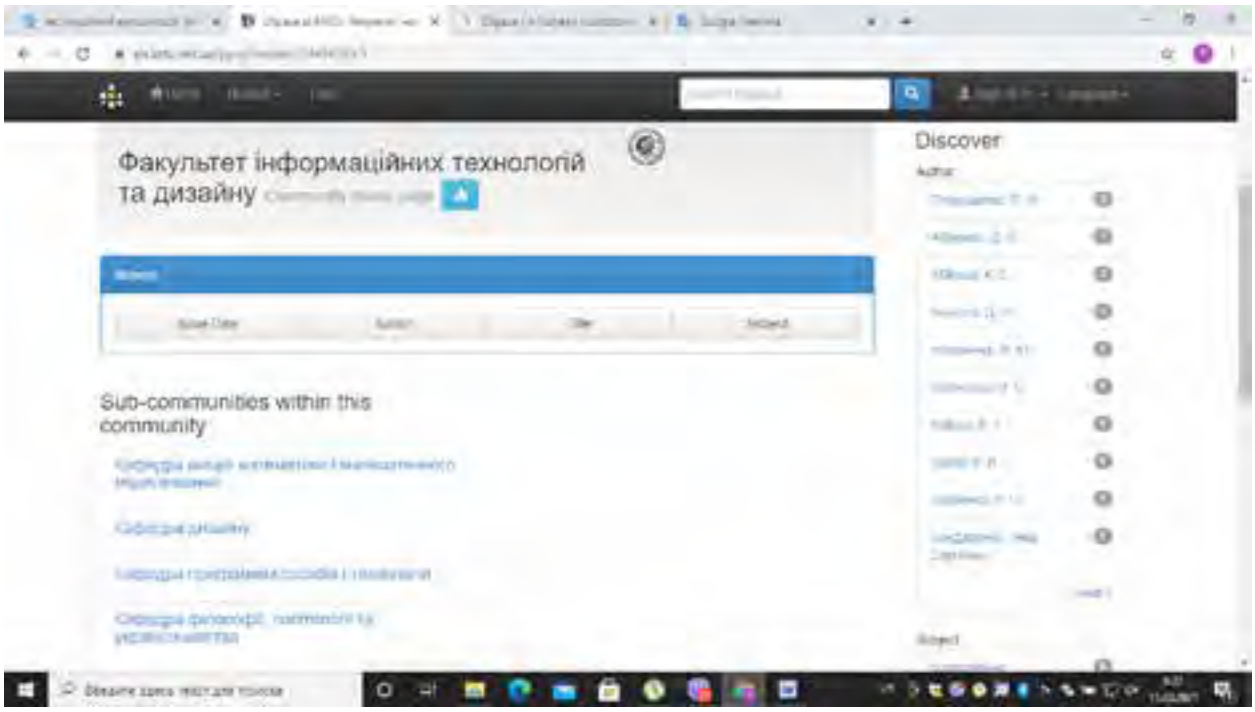

Рис. 5 – Перегляд інформації по факультету в інстутиційному репозитарії ХНТУ

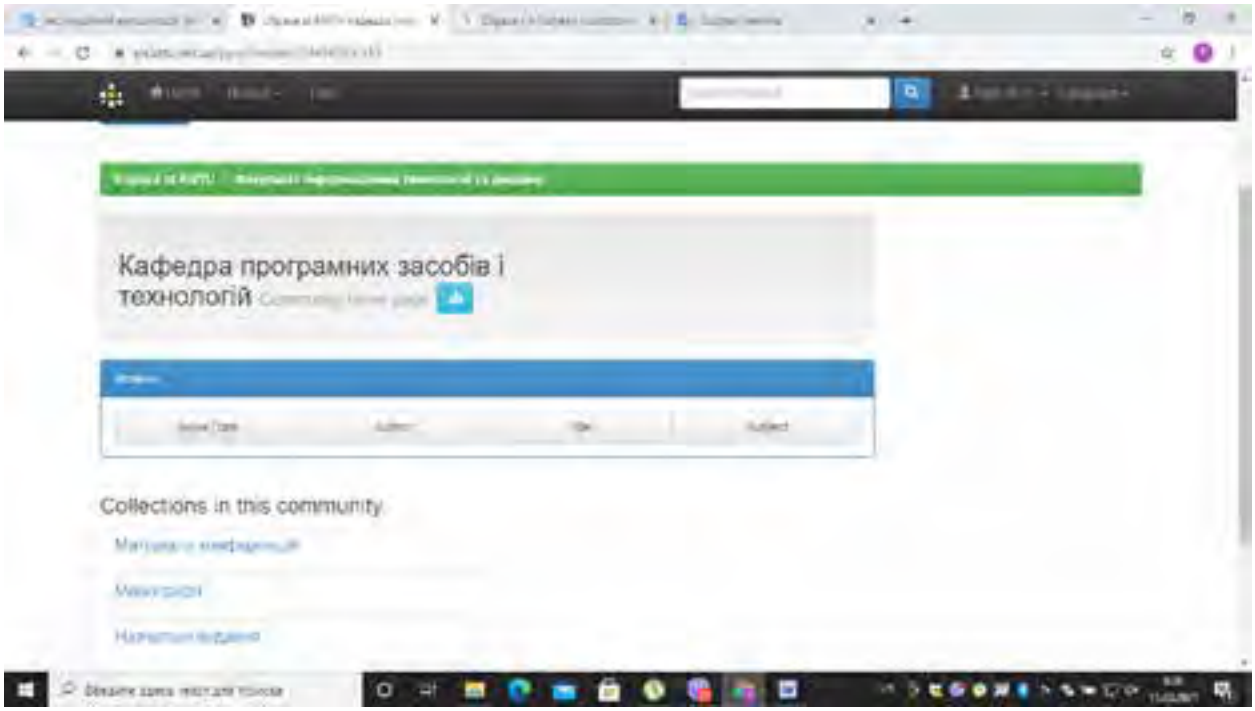

Рис. 6 – Розділи збереження інформації по кафедрі в інстутиційному репозитарії ХНТУ

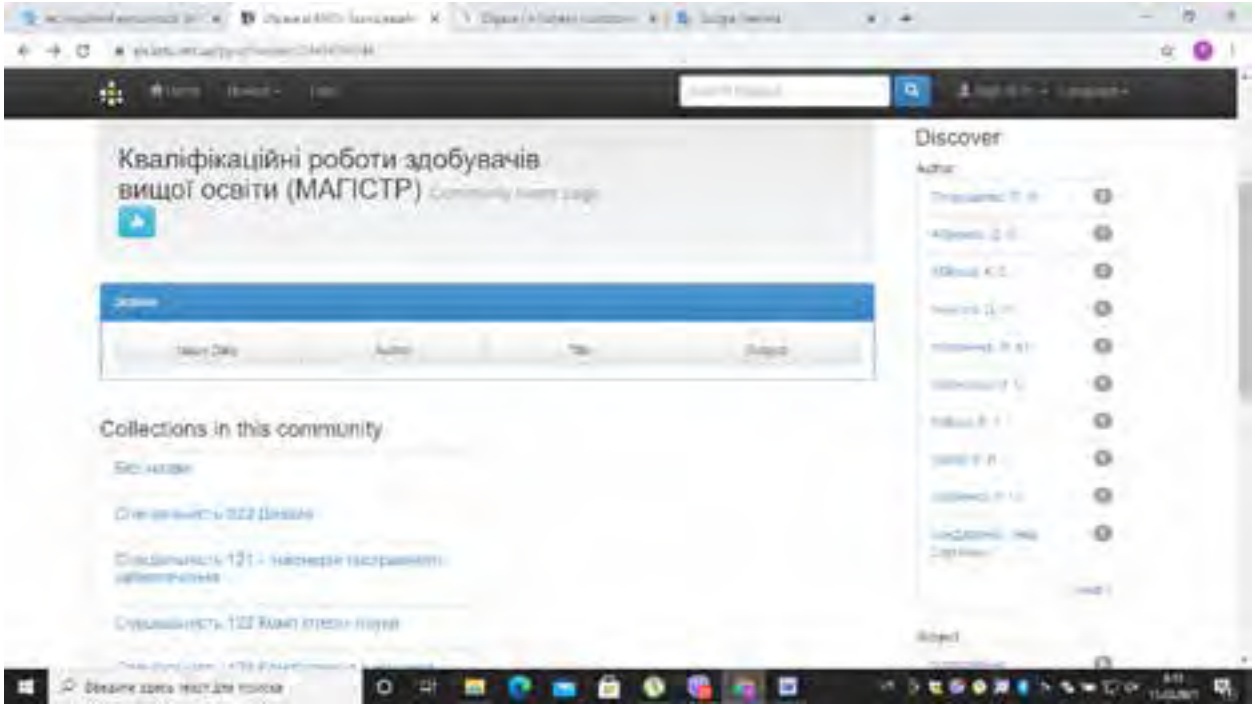

Рис. 7 – Збереження кваліфікаційних робіт магістрів в інстутиційному репозитарії ХНТУ

Висновки. Досліджено способи і методи збереження інформації в сховищах даних. Наведені основні переваги та недоліки збереження інформації в сховищах даних різними способами. Наведено приклад використання програмного продукту DSpace для збереження інформації в інституційному репозитарії ХНТУ. Програмний продукт може використовуватися в освітньому процесі у якості наочного подання інформації за факультетами, кафедрами та спеціальностями, для збереження інформації за різними напрямками, що дозволило пришвидшити доступ до потрібної інформації та структурувати її за потрібними напрямами.

СПИСОК ЛІТЕРАТУРИ:

1. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки данных. Вводный курс. – М.: Гелиос АРВ, 2002. – 368 с.
2. Инмон Б. Типы хранилищ данных Перевод Intersoftlab, 2001.
3. Лоуенд Ш., Хилсон С., Хоббс Л. Oracle9iR2. Разработка и эксплуатация хранилищ баз данных. М: КУДИЦ-ОБРАЗ, 2004. – 592 с.
4. Rainardi V. Building a Data Warehouse: With Examples in SQL Server 2008, APRESS – P. 541
5. Архипенков С., Голубев Д., Максименко О. Хранилища данных. От концепции до внедрения М: Диалог-МИФИ, 2002. – 528 с.
6. Маклаков С.В., Матвеев Д.В. Анализ данных. Генератор отчетов Crystal Reports СПб: БХВ-Петербург, 2003. – 496 с.
7. Маклаков С.В., Туманов В.Е. Проектирование реляционных хранилищ данных М: Диалог-МИФИ, 2007. – С. 333.
8. Спирли Э. Корпоративные хранилища данных. Планирование, разработка, реализация М: Издательский дом «Вильямс», том I, 2001. – С. 400

RESEARCH OF METHODS OF INFORMATION STORAGE IN DATA STORES

Raisa Zakharchenko,
Ph.D., Associate Professor of Software and Technology, Kherson National Technical University, Kherson, Ukraine,
e-mail: zraissa2@gmail.com, ORCID 0000-0003-4650-3095

Leonid Zakharchenko,
graduate student of the Department of Software and Technologies Kherson National Technical University, Kherson, Ukraine,
ORCID 0000-0001-9984-696X

Tetyana Kiryushatova,
Ph.D., Associate Professor of Software and Technology, Kherson National Technical University, Kherson, Ukraine,
e-mail: tanyakir1963@gmail.com, ORCID 0000-0002-0000-0065

Kybalko Ihor Ivanovych,
Ph.D., Senior Engineer of Software and Technology, Kherson National Technical University, Kherson, Ukraine,
e-mail: kybalko.igor@kntu.net.ua, ORCID 0000-0002-6634-5277

Abstract. The goal is to study the theoretical and methodological foundations of building an effective data storage for the most rational preservation of information.

Research methods. The methods of scientific research are used in the work such as: experiment, analysis of the results of activity. From theoretical research methods used: analysis, synthesis, comparison.

The main results of the study. Methods and methods of information storage in data storages are investigated. The main types of software and hardware architecture of data storages are considered, their advantages and disadvantages are defined. The result of the study was the classification of methods for storing information, which helps the designer to create a data store with the lowest costs. The main components of the information storage are presented, which made it possible to build a typical data storage structure. Comparative analysis of methods of data storage modeling is carried out, which allows the user to use in practice a more effective method in the design and development of data storage facilities for storing information in organizations and enterprises.

Scientific novelty. The development of information resources and means of access to them, the rapid development of Ukrainian-language Internet content are factors that have changed the way and approach to preserving large amounts of

information. The scientific novelty of this topic is the use of modern data storages for storing information in higher educational institutions.

Practical significance. Every year, the increasing amount of information that needs to be stored and processed has led to the need to design and develop data stores. This topic is very relevant for study nowadays, when data stores are increasingly beginning to be implemented in various spheres of human activity, such as education, agriculture, education, medicine, economics, communication, security of security systems, information processing, etc. The results of the study can be used in the educational process to a clear picture of the advantages and disadvantages of methods of storing information in data stores, the principles of their use in the industrial and non-industrial spheres.

Keywords: *multidimensional modeling, modeling of temporary data, modeling of "data output", scheme "star", scheme "snowflake", diagrams with several tables of facts.*

ИССЛЕДОВАНИЕ МЕТОДОВ СОХРАНЕНИЯ ИНФОРМАЦИИ В ХРАНИЛИЩАХ ДАННЫХ

Раиса Захарченко,
к.т.н., доцент кафедры Программных средств и технологий,
Херсонский национальный технический университет, Херсон, Украина,
e-mail: zraissa2@gmail.com, ORCID 0000-0003-4650-3095

Леонид Захарченко,
аспирант кафедры Программных средств и технологий,
Херсонский национальный технический университет, Херсон, Украина,
ORCID 0000-0001-9984-696X

Татьяна Кирюшатова,
к.т.н., доцент кафедры Программных средств и технологий,
Херсонский национальный технический университет, Херсон, Украина,
e-mail: tanyakir1963@gmail.com, ORCID 0000-0002-0000-0065

Кибалко Игорь Иванович,
к.т.н., ведущий инженер кафедры Программных средств и технологий,
Херсонский национальный технический университет, Херсон, Украина,
e-mail: kybalko.igor@kntu.net.ua, ORCID 0000-0002-6634-5277

Аннотация. Целью является исследование теоретических и методологических основ построения ефективного сховища даних для наиболее рационального сохранения информации.

Методы исследования. В работе использованы методы научных исследований такие как: эксперимент, анализ результатов деятельности. Из теоретических методов исследования использованы: анализ, синтез, сравнение.

Основные результаты исследования. Исследованы способы и методы сохранения информации в хранилищах данных. Рассмотрены основные типы программно-аппаратной архитектуры хранилищ данных, определены их преимущества и недостатки. Результатом исследования стала классификация методов хранения информации, которая помогает проектировщику создать хранилище данных с наименьшими затратами. Представлены основные компоненты информационного хранилища данных, что позволило построить типовую структуру хранилища данных. Проведен сравнительный анализ методов моделирования хранилищ данных, который позволяет пользователю воспользоваться на практике более эффективным методом при проектировании и разработке хранилищ данных для хранения информации в организациях и на предприятиях.

Научная новизна. Развитие информационных ресурсов и средств доступа к ним, стремительное развитие украиноязычного контента Интернета являются факторами, которые изменили способ и подход к сохранению больших объемов информации. Научной новизной данной темы является использование современных хранилищ данных для хранения информации в высших учебных заведениях.

Практическая значимость. С каждым годом все больший объем информации, которую нужно хранить и обрабатывать привело к необходимости проектирования и разработки хранилищ данных. Данная тема очень актуальна для изучения в наше время, когда хранилища данных все более активно начинают внедряться в различные сферы человеческой деятельности, такие как образование, сельское хозяйство, обучение, медицину, экономику, связь, безопасность охранных систем, обработку информации и т. д. Проведена классификация может стать основой для расширения количества сфер использования хранилищ данных. Результаты исследования можно использовать в учебном процессе для наглядного представления преимуществ и недостатков методов сохранения информации в хранилищах данных, принципов их использования в промышленной и не промышленной сферах.

Ключевые слова: *многомерное моделирование, моделирование временных данных, моделирование «свод данных», схема «звезда», схема «снежинка», схемы с несколькими таблицами фактов.*

REFERENCES:

1. Haidamakyn N.A. Avtomatizirovannye informatsionnye systemy, bazy i banki dannykh. Vvodniy kurs. – M.: Helios ARV, 2002. – 368 p.
2. Inmon B. Tipy khranilishch dannykh Perevod Intersoftlab, 2001.
3. Louend Sh., Khilson S., Khobbs L. Oracle9iR2. Razrabotka i ekspluatatsiya khranilishch baz dannykh. M: KUDITs-OBRAZ, 2004. – 592 p.
4. Rainardi V. Building a Data Warehouse: With Examples in SQL Server 2008, APRESS – 541 p.
5. Arkhipenkov S., Holubev D., Maksimenko O. Khranilishcha dannykh. Ot kontseptsii do vnedreniya M: Dialoh-MIFI, 2002. – 528 p.
6. Maklakov S.V., Matveev D.V. Analiz dannykh. Henerator otchetov Crystal Reports SPb: BKhV-Peterburh, 2003. – 496 p.
7. Maklakov S.V., Tumanov V.E. Proektirovanie reliatsionnykh khranilishch dannykh M: Dialoh-MIFI, 2007. – 333 p.
8. Spirli E. Korporativnye khranilishcha dannykh. Planirovanie, razrabotka, realizatsiya M: Izdatelskiy dom «Vyliams», tom I, 2001. – 400 p.