

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: **Розробка методів і програмний засіб аналізу якості даних при функціонуванні програмного забезпечення з базами даних і знань на основі стеку технологій HTML/CSS, PHP та MySQL**

Виконала: студентка 6 курсу, групи СПм-61

спеціальності 121 «Інженерія програмна забезпечення»
(шифр і назва спеціальності)

		Демида Х.М. (прізвище та ініціали)
Керівник		Пастух О.А. (прізвище та ініціали)
Нормоконтроль		Стоянов Ю.М. (прізвище та ініціали)
Завідувач кафедри		Петрик М.Р. (прізвище та ініціали)
Рецензент		 (прізвище та ініціали)

Тернопіль
2021

РЕФЕРАТ/ABSTRACT

Атестаційна робота магістра містить: 115 с., 21 рис., 22 табл., 33 джер.

КЛЮЧОВІ СЛОВА: МОДЕЛЬ, ЯКІСТЬ, ДАНІ, АНАЛІЗ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

У кваліфікаційній роботі досліджено методи та реалізовано інструментальний програмний засіб аналізу та оцінювання якості даних при функціонуванні програмного забезпечення з базами даних і знань.

Запропоновано математичне представлення структури моделі ISO 25012 у вигляді комплексу показників якості даних, що включає в себе три ієрархічні блоки і дає змогу оперувати якістю даних на різних рівнях, створено метод аналізу та оцінювання якості даних при функціонуванні програмних систем з базами даних і знань, що забезпечує можливість представлення якості даних у кількісному вигляді.

KEYWORDS: MODEL, QUALITY, DATA, ANALYSIS, SOFTWARE.

In the qualification work the methods and the tool software of the analysis and estimation of data quality at functioning of the software with databases and knowledge are investigated.

Mathematical representation of the structure of the ISO 25012 model in the form of a set of data quality indicators is proposed, which includes three hierarchical blocks and allows to operate with data quality at different levels, created a method of analysis and evaluation of data quality the ability to represent the quality of data in quantitative terms.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	5
ВСТУП.....	6
1 АНАЛІЗ СУЧАСНИХ ПРОГРАМНИХ СИСТЕМ З РІЗНИМИ ВИДАМИ БАЗ ДАНИХ.....	10
1.1. Аналіз особливостей організації та функціонування NoSQL баз даних.	10
1.2. Характеристик і формат даних у NoSQL базах даних	12
1.3. Аналіз характеристик реляційних баз даних	20
1.4. Оцінка результатів порівняння NoSQL баз даних та MySQL	24
2 ОБГРУНТУВАННЯ МОДЕЛІ ТА РОЗРОБКА МЕТОДУ АНАЛІЗУ ЯКОСТІ ДАНИХ ПРИ ФУНКЦІОНУВАННІ СИСТЕМ З БАЗАМИ ДАНИХ І ЗНАНЬ	30
2.1. Аналіз моделі стандарту ISO/IEC 25012 та його рекомендацій.....	30
2.2. Формалізація характеристик моделі якості даних стандарту ISO/IEC 25012.....	34
2.2.1. Характеристик якості даних з точки зору невід’ємності.....	34
2.2.2. Характеристик якості даних з невід’ємної і системозалежної точок зору	38
2.2.3. Характеристик якості даних із системозалежної точки зору	47
2.3. Комплексна модель якості даних на рівні характеристик та атрибутів.....	49
2.4. Розробка методу аналізу якості даних при функціонуванні систем з базами даних і знань	55
3 ПРОГРАМНИЙ ЗАСІБ АНАЛІЗУ ЯКОСТІ ДАНИХ ПРИ ФУНКЦІОНУВАННІ СИСТЕМ З БАЗАМИ ДАНИХ І ЗНАНЬ.....	61
3.1. Обґрунтування технологій реалізації програмної системи	61
3.2. Проектування та імплементація тестової бази даних	62

3.3. Реалізація системи оцінювання якості даних	68
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	77
4.1. Охорона праці.....	77
4.2. Фактори, що впливають на функціональний стан користувачів комп'ютера.....	77
ВИСНОВКИ.....	84
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	86
ДОДАТОК А Апробація результатів роботи	89
ДОДАТОК Б Фрагменти програмного коду програмного засобу аналізу якості даних	95
ДОДАТОК В Слайди презентації	106

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	База Даних
ІС	Інтелектуальна Система
ПС	Програмна Система
СКБД	Система Керування Базами Даних
CASE	Computer Aided Software Engineering
UML	Unified Modeling Language
CSS	Cascade Style Sheet
ISO	International Organization for Standardization
HTML	Hyper Text Markup Language
PHP	Hypertext Preprocessor
SQL	Structured Query Language
SQuaRE	Systems and software Quality Requirements and Evaluation

ВСТУП

Актуальність теми. Інформаційний контекст сучасного бізнесу стрімко розвивається в умовах інтенсивної глобальної конкуренції. Прийняття рішень керівниками в даний час значною мірою залежить від своєчасної, доступної та актуальної інформації у контексті вирішення поставлених проблем. Для ефективності роботи менеджера також необхідно забезпечувати максимальну повноту і точність наявної інформації, які формують його знання про стан бізнес-процесів. Знання, в свою чергу можна отримати або з власних ресурсів організації, або з наявних зовнішніх джерел.

Постійний розвиток інформаційно-комунікаційних технологій породив нові системи та платформи, які щосекунди генерують велику кількість даних. Дані потенційно можуть бути носієм важливої інформації, необхідної для створення подальших знань, пов'язаних з конкретною проблемою. Завдяки відкритості Інтернету та його взаємозв'язку майже з усіма пристроями як у споживачів, так і в підприємствах, неструктуровані дані набувають все більшого поширення. Неструктуровані дані – це такі дані, які не мають певної структури і які не зберігаються в моделі реляційної бази даних.

Великі обсяги неструктурованих даних мають характеристики, відмінні від даних, згенерованих власними системами підтримки бізнес-процесів та прийняття рішень. Реляційні структури баз даних (наприклад, сховищ даних) та відповідні інструменти не можуть ефективно опрацьовувати та аналізувати значні обсяги інформації з точки зору розумного використання часу та ресурсних витрат. Це стало основною причиною появи технології BigData.

Оскільки, на сьогодні для підтримки бізнес-процесів 80% ринку займають інформаційні системи з реляційними базами даних, а становлення сфери BigData та NoSQL баз даних все ще інтенсивно розвивається, то актуальною задачею в індустрії програмування є забезпечення якості даних, що є складним

комплексним показником і вимагає розробки та впровадження методів і засобів для оцінювання їх якості.

Функціонування сучасних прикладних систем та створення нових, особливо у сегменті машинного навчання, характеризується різнотипним і неструктурованим представленням даних. У багатьох випадках природа інформації та її представлення є неузгодженою та неоднорідною відносно форматів зберігання у комп'ютерних системах, може містити помилки і дефекти. Це негативно відображається на трактуванні результатів опрацювання даних, що призводить до прийняття не правильних рішень.

У зв'язку з цим актуальності набуває задача розробки методів і програмного засобу аналізу якості даних при функціонуванні програмного забезпечення з базами даних і знань.

Метою дипломної роботи є дослідження методів та інструментальних програмних засобів аналізу та оцінювання якості даних при функціонуванні програмного забезпечення з базами даних і знань.

Досягнення мети роботи відбувається шляхом розв'язання поставлених **задач**:

- аналіз наукових публікацій, стандартів і технологій проектування програмних систем з базами даних і знань;
- обґрунтування та математична формалізація моделі аналізу якості даних;
- розробка методу аналізу даних щодо відповідності критеріям якості даних;
- проектування структури системи автоматизації процесу аналізу даних у системах з базами даних і базами знань;
- програмна реалізація прототипу системи аналізу та оцінювання даних.

Методи дослідження. Для досягнення мети використано наступні методи і технології: аналіз та узагальнення – при проведенні аналізу існуючих підходів, методів і засобів побудови та функціонування програмних систем з базами даних і знань; математичного моделювання та експертних технологій – при формалізації

моделі якості даних; програмування і тестування – при реалізації системи автоматизованого аналізу якості даних та перевірки її працездатності.

Об’єктом дослідження є процеси аналізу та оцінювання якості даних при функціонування систем з базами даних і знань.

Предметом дослідження є технології аналізу якості даних у базах даних і знань.

Наукова новизна одержаних результатів.

– уперше запропоновано математичне представлення структури моделі ISO 25012 у вигляді комплексу показників якості даних, що включає в себе три ієрархічні блоки і дає змогу оперувати якістю даних на різних рівнях, а також спрощує реалізацію процесу автоматизації аналізу даних при функціонуванні систем з базами даних і знань.

– уперше створено метод аналізу та оцінювання якості даних при функціонуванні програмних систем з базами даних і знань, що забезпечує можливість представлення якості даних у кількісному вигляді та оперувати такими показниками як точність, ефективність, актуальність даних тощо.

Практична цінність результатів дослідження. Впровадження методу і системи аналізу якості даних, реалізованої за допомогою стеку технологій HTML/CSS та PHP дає можливість підвищити точність та ефективність прийняття рішень на основі наявної актуальності про бізнес-процеси і предметну область.

Публікації. Основні результати дослідження апробовано на X міжнародній науково - технічній конференції молодих учених і студентів «Актуальні задачі сучасних технологій» (24-25 листопада 2021 р.) Тернопільського національного технічного університету імені Івана Пулюя та на IX науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2021 року) у вигляді тез конференцій.

1. Пастух О.А., Василюшин В.І., Демида Х.М. Аналіз рівнів зрілості вимог при розробці програмного забезпечення. Матеріали X міжнародної науково - технічної конференції молодих учених і студентів «Актуальні задачі сучасних

технологій» (24-25 листопада 2021 р.) Тернопільського національного технічного університету імені Івана Пулюя. Тернопіль: ТНТУ. 2021. С. 123.

2. Пастух О.А., Демида Х.М., Василюшин В.І. Характеристики якості даних при функціонуванні програмного забезпечення з базами даних і знань. Матеріали X міжнародної науково - технічної конференції молодих учених і студентів «Актуальні задачі сучасних технологій» (24-25 листопада 2021 р.) Тернопільського національного технічного університету імені Івана Пулюя. Тернопіль: ТНТУ. 2021. С. 124.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 4 розділів, висновків, списку використаної літератури та додатків. Обсяг роботи: пояснювальна записка – 115 арк. формату А4, графічна частина – 30 слайди.

1 АНАЛІЗ СУЧАСНИХ ПРОГРАМНИХ СИСТЕМ З РІЗНИМИ ВИДАМИ БАЗ ДАНИХ

1.1. Аналіз особливостей організації та функціонування NoSQL баз даних

Сучасні технології програмування та створені на їх базі системи дозволяють збирати велику кількість інформації, більшість з яких – це дані, поширенні у різних джерелах, таких як Інтернет, соціальні сайти, веб-додатки тощо. Зазвичай, така інформація протягом довгого періоду часу зберігалась в СКБД на основі реляційного підходу. Однак на сьогодні такий підхід майже вичерпує себе і ресурсів уже не достатньо для ефективного опрацювання інформації в режимі реального часу. Більшість даних починають мігрувати у хмарні сховища, однак це в свою чергу створює багато проблем із безпекою.

Основна задача хмарних сервісів і сховищ при роботі з великими даними полягає у захопленні, зберіганні, опрацюванні та інтерпретації інформації. Big Data характеризуються такими основними характеристиками як обсяг, швидкість, актуальність і різноманітність [2].

На сьогоднішній час технологія NoSQL відіграє важливу роль в аналізі та зберіганні даних. Хмарні обчислення виконуються на нових платформах, що дозволяють користувачеві використовувати їх джерело відповідно до своїх вимог. Деякі з традиційних методів виконання операцій полягають у використанні таких ресурсів, як комп'ютерна мережа та кластерні обчислення.

Застосування «хмари» забезпечує гнучкість реалізації програмних рішень, які пропонують різні постачальники цих послуг, зокрема таких технологічних рішень як:

- «інфраструктура як послуга (IaaS)»;
- «платформа як послуга (PaaS)»;
- «програмне забезпечення як послуга (SaaS)».

Ці моделі використовують як державні, так і приватні організації. Гнучкість, масштабованість, ефективність і можливість повторного використання є одними з переваг використання хмарних обчислень. До NoSQL баз даних належать, такі як HBase, Cassandra, MongoDB. Ці бази даних володіють як перевагами, так і недоліками.

Потрібно відмітити, що операція читання та запису у них в основному залежить від продуктивності та масштабованості як із простими, так і з складними запитами.

У [3] представлено життєво важливі характеристики, що забезпечуються застосуванням і перевагами підходу NoSQL, такими як доступність і масштабованість. У [4] досліджено виконання операцій читання та оновлення у різноманітних базах даних: Cassandra, HBase та Riak.

У HBase пропонується більша гнучкість і швидке читання, але в Cassandra є можливість швидкої вставки в операції запису. Тим часом, Riak демонструє масштабованість і продуктивність, враховуючи тип і метод доступу до даних.

Більшість дослідників зосереджуються на оцінках ефективності систем для роботи з базами даних. У [4-6] наведено результати оцінювання продуктивності технології MongoDB і бази даних MS-SQL. Індекс, реплікація, спільний доступ, запит, сучасний гнучкий режим даних – це деякі аспекти, які надаються більшістю інтернет-додатків. Порівнюючи продуктивність обох баз даних, NoSQL забезпечує більшу ефективність, ніж MS-SQL. Завдяки швидкому вдосконаленню підходу великих даних подальша тенденція залежить від інтеграції із зовнішніми сервісами на основі нереляційних баз даних.

У [7] проаналізовано дві відомі бази даних з підтримкою підходу NoSQL, зокрема, MongoDB і Cassandra. Аналіз проводився в контексті оцінки продуктивності за часом і розміром бази даних. Процеси читання/оновлення, тільки читання, читання з оновленням записів та їх суміш були оцінені на основі сформованих тестів, які виконуються в робочому навантаженні. Через збільшення розміру даних MongoDB починає знижувати продуктивність, показує низький результат, а Cassandra дозволяє працювати з більшою кількістю даних. Після

виконання різних робочих навантажень необхідно проаналізувати продуктивність як читання, так і оновлення, це має потенціал для операції оновлення.

Cassandra швидше, ніж MongoDB, володіє вищою продуктивністю, а розмір бази даних вимагає проведення додаткових досліджень. Виконання повного аналізу характеристик MongoDB забезпечує меншу кількість дефектів і помило із більшою кількістю записів, однак Cassandra володіє ширшою функціональністю у порівнянні з MongoDB. У майбутньому автор [7] пропонує виконувати більшу кількість операцій в секунду порівняно з розміром бази даних.

Для оцінювання і підтвердження продуктивності нереляційних баз даних використовується теорема CAP[6].

У [8] описано теорему CAP, вона стверджує, що для розподіленої системи, неможливо одночасно забезпечити властивості узгодженості, доступності і розподіленої стійкості. Узгодженість базується на перегляді усіх вузлів з подібними даними за однаковий час [7]. Більшість баз даних орієнтовані лише на забезпечення доступності і розподіленої стійкості,

1.2. Характеристик і формат даних у NoSQL базах даних

Технологія NoSQL вперше реалізована у Big Table [9]. Вона базується на використанні трьох ключових характеристик, таких як рядок, стовпець і мітка часу і застосовується для створення ефективного багатовимірного представлення даних. Ключ стовпця відображає категорію і виконується одним блоком. Big Table і Amazons' Dynamo [10] є одними з представників нереляційних БД, що функціонують як у відкритому, так і в закритому вигляді. Легкість доступу, швидкість, масштабованість стають все більш популярними із застосуванням NoSQL.

Представлення даних у NoSQL, у більшості випадків, залежить від пар ключ-значення і також може використовуватись при створенні зовнішніх ключів на основі вмісту значення. Дані сімейства стовпців залежать від пари значень ключів. Ключова адреса складається з стовпця та суперстовпця. Супер-стовпець

формується із стовпця і виконує лише окремі функції, якийсь інший ключовий тип на основі документо-орієнтованої бази даних складається з простого значення і має можливість зберігати об'єкт.

XML, JSON і BSON є форматами, які використовуються в Objects. Нижче наведено унікальні характеристики NoSQL бази даних:

- зберігання значно більшої кількості інформації у порівнянні з традиційними БД;
- мова SQL не використовується у NoSQL базах даних;
- здатність до функціонування за відсутності узгодженості у розподіленій платформі;
- не відповідність властивостям ACID;
- більш гнучка структура даних порівняно з реляційними БД;
- зростання продуктивності завдяки горизонтальній масштабованості;
- усі дані в БД NoSQL зберігаються без використання фіксованих схем.

Mongo DB зберігає дані у вигляді, що залежить від структури документа в базі даних. Документи групується у колекції у відповідності до структури, але в деяких випадках документ може бути частиною іншої структури.

Mongo DB дотримується визначеного стандартного формату для зберігання документів, наприклад BSON-Binary JSON, розмір кожного з яких обмежено об'ємом у 16 МБ. Довговічність і паралелізм є важливими характеристиками Mongo DB. Впровадження реплікації, як правило, забезпечує довговічність даних.

Mongo DB використовує механізм реплікації «Master-Slave», що передбачає наявність одного «ведучого» вузла і одного або більше «залежних». «Ведучий» буде читати або записувати дані, коли ведений допомагає підтримувати операції читання. У випадку, коли головний вузол виходить з ладу, підпорядкований йому вузол підвищується до «ведучого» і виконує усі реплікації. Таке формування реплік буде асинхронним у часі, а оновлення не будуть виконуватись безпосередньо. Налаштування реплікації здійснюється адміністратором одним з багатьох способів:

- єдиний залежний вузол – реплікація зберігатиме дані, але при умові формування позначок про виконання функцій вузлом «slave» задач «Master».

- прихований вузол – прихована реплікація не буде виконувати функції «Master» і є невидимою для користувацького ПЗ. Більшість вузлів можуть фіксувати резервне копіювання, і це лише за умови зчитування даних.

- вузол із затримкою копіювання – створює копію первинного блоку з репліки з певною затримкою, коли дані еквівалентні попередньому стану на головному вузлі.

Арбітри – тільки внутрішні вузли беруть участь лише у спілкуванні з іншими.

Вузли, які не мають права голосу – не беруть участі в реплікації і виступають у ролі великого кластеру, з кількістю більше 7 членів.

Cassandra представляє собою базу даних з відкритим вихідним кодом, що ідентифікує дані за значенням ключа і є представником БД подібних до послідовності стовпців у dynamo (зберігає дані в сімействі стовпців, як у великій таблиці).

Cassandra реалізовує процес управління великими структурованими даними, що забезпечує зручність їх використання в Apache. Деякі властивості БД Cassandra здатні забезпечити її гнучкість та лінійність.

Продуктивність баз даних на основі Cassandra буде зростати, коли в кластері будуть наявні декілька вузлів. Подібно до реляційної бази даних, Cassandra підтримує властивості ACID, вона швидко виконує запис інформації. Також ця технологія підтримує розподіл даних при реплікації даних у центрах обробки інформації. Мова програмування, що використовується в Cassandra - Java.

Hbase – це база даних сімейства стовпців, яку визначається за допомогою Java і подібна до таблиці великих даних[9]. Hbase реалізується поверх HDFS і забезпечує виконання таких функцій великої таблиці, як швидке опрацювання структурованих і неструктурованих даних, які зберігаються в HDFS.

Така технологія забезпечує пакетне опрацювання даних та володіє додатковою функціональністю, що підвищує ефективність використання індексів стовпців. Hbase подібна до Cassandra, тобто є лінійною і масштабованою, а також визначає індекси за допомогою master-slave. Коли Various HMaster реагує на збій HMaster, а навантаження залишилося на сервері регіону, інший Hbase автоматично виконує свою роботу. HBase дозволяє автоматично підтверджувати збій за заданими сценаріями. Hmaster також функціонує, коли одночасно виконується операція читання та запису на запит програми, що оперує з великими даними. У табл. 1.1 наведено переваги та недоліки NoSQL баз даних.

Таблиця 1.1 – Переваги та недоліки NoSQL баз даних

Переваги	Недоліки
Підтримка і простота масштабованості рішень	Незрілість
Не потребують адміністратора	Недостатня швидкість і гнучкість при навантаженнях
Працює з більшим простором даних	Складність при модернізації
Величезний вибір моделей даних	Відсутня стандартизована мова запитів
NoSQL, DBaaS, Riak, Cassandra запрограмовані для роботи при збоях апаратного забезпечення.	Лише кілька NoSQL баз даних взагалі не мають негативних відгуків

MySQL, Oracle є прикладами реляційних баз даних, які призначені для виконання таких операцій, як зберігання, видалення і маніпулювання даними. Реляційні БД, на відміну від NoSQL баз даних, використовують однакову мову запитів, що дозволяє одержувати відповідні результати на запити користувачів.

У базі даних NoSQL зберігається конкретна модель даних. Таблиця 1.2 ілюструє різницю в операціях читання, запису, видалення, що виконуються в

різних NoSQL базах даних і таких як MySQL. Зразки запитів для кожної операції наведено у табл. 1.2.

Таблиця 1.2 – Приклади операцій у різних базах даних

База даних	Операція читання	Операція запису	Операція видалення
My SQL	Insert into journal values ('scope',234, both open and close)	Update set of Journal Id=345 where 357	Delete from list of journal where Name='pharmacy'
MongoDB	Db.journal.insert ({jname:"scopex", id:235,type:"both open and closed"})	Db.journal update({}('\$set ' :id' :id'}));	Db.Journal.delete()
Cassandra	Insert into journal values('scopex ".234,'both open and close');	Update Set of journal Id=234 where id=350;	Delete set of journal id=234 where id=350
Hbase	Assign journal', 'row2', 'id:a', 'scope',	Similar to insert operation	Disable 'set of journal'

Таблиця 1.3 показує порівняльний аналіз різних баз даних NoSQL з урахуванням різних аналітичних параметрів, включаючи:

- використовувану мову програмування;
- відмовостійкість;
- використовувані носії даних;
- фреймворк MapReduce, який використовується;
- тип використовуваної мови запитів;
- різні доступні режими реплікації;
- протокол, що використовується на транспортних і прикладних рівнях системи.

Таблиця 1.3 – Порівняльний різних аналіз баз даних

Властивості БД	MongoDB	Cassandra	Accumulo	Couch DB	Hbase	Redis	Riak
Мова програмування	C++	Java	Java	Erlang	Java	C,C++	Erlang
Модель даних	BSON	Big Table	Big Table	JSON	Big Table та Dynamo	Структура даних	Структура даних
Відмовостійкість	Реплікація	Розподіленість і реплікація	Реплікація	Реплікація	Розподіленість і реплікація	Реплікація	Реплікація
Носій даних	Пам'ять, файлова платформа	Dynamo для зберігання даних	HDFS	Пам'ять, файлова платформа	HDFS	Файлова система	Пам'ять
Ком'юніті	AGPL	Facebook	Apache	Apache	Apache	BSD	Apache
MapReduce	+	+	+	+	+	-	+
Мова запитів	-	API виклики	Java API, Thrift API	-	XML, Thrift API	API виклики	Javascript
Моделі реплікації	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація
Протокол	TCP/IP	Thrift	Thrift	HTTP/ REST	Thrift, API	Бінарний протокол, подібний до telnet	REST

Різні проблеми NoSQL баз даних, такі як аутентифікація, авторизація, атаки, протоколи, наведені у таблиці 1.4.

Таблиця 1.4 – Показники засобів безпеки у NoSQL БД

NoSQL бази даних	Тип	Аутенти-фікація	Автори-зація	Атаки	Протокол
Redis	Ключ-значення	+	-	-	Без шифрування
DynamoDB	Ключ-значення	+	-	-	Https
Voldemort	Ключ-значення	-	-	-	-
MongoDB	Документо-орієнтований	-	-	Ін'єкція скрипта	SSL
CouchDB	Ключ-значення	+	-	DOS	SSL
Cassandra	Документо-орієнтований	+	-	DOS та ін'єкція скрипта	SSL
Hbase	Column Based	+	+	Відсутні звіти про DOS та ін'єкції	SSH
HyperTable	Column Based	-	-	-	-
Neo4j	Графовий	-	-	-	SSL

Проведемо аналіз характеристик та особливостей застосування реляційних баз даних.

1.3. Аналіз характеристик реляційних баз даних

Вибір засобів-кандидатів для оцінювання продуктивності є важливою складовою у процесі досягнення оптимального рішення щодо ефективності сучасних реляційних СКБД. Перш за все варто сформулювати список можливих кандидатів для тестування ефективності.

На даний час у сфері інформаційних технологій серед реляційних СКБД існує величезна різноманітність варіацій концепцій, архітектури та реалізацій ПЗ для обслуговування процесів управління даними. На основі критеріїв розповсюдженості, технічної завершеності та документованості відібрано 4 засоби-кандидати: MySQL, Oracle, IBM, MS SQL Server. Було проведено їх короткий огляд та аналіз, результати якого представлені в табл.1.5.

Таблиця 1.5 – Порівняння реляційних СКБД

Характеристика	MySQL	Oracle	IBM DB2	MS SQL Server
Поширеність	середня	лідер	низька	висока
Призначення	Невеликі офіси, веб-розробка	Різнноманітні сфери застосування	Промислові бази даних	Різнноманітні сфери застосування
Документованість	повна	повна	повна	повна
Доступність	висока	висока	висока	висока
Тип ліцензії	вільна/комерційна	вільна/комерційна	вільна/комерційна	вільна/комерційна

Продовження табл. 1.5

Характеристика	MySQL	Oracle	IBM DB2	MS SQL Server
Підтримка операційними системами	Кросплат-формна	Кросплат-формна	Кросплат-формна	Windows
Підтримка збережених процедур	слабка	повна	слабка	повна
Тести TPC	-	TPC-H (1), TPC-C(1)	TPC-C(2)	TPC-E(1), TPC-H(2)
Доступ	ODBC	OracleClient	ODBC	SqlClient

Для перевірки продуктивності СКБД обрано Oracle та MS SQL Server, як цільові СКБД для виконання оцінювання. Даний вибір було зроблено на основі поширеності даних СКБД, хорошої підтримки засобами розробки, підтримки ними основних концепцій РСКБД, високих показників тестів TPC.

Тести проводились на реляційних таблицях із однаковим складом полів: int/number(10), varchar(10)/varchar2(10), varchar(max)/varchar2(max), char(10), bit/number(1), float. Атрибути для оцінки було обрано швидкість вставки та швидкість вибірки даних, обрано кількість ітерацій рівну 10000, випробування проводились для прямих викликів та із використанням збережених процедур. Результати оцінювання та порівняння із офіційними тестами TPC та інших організацій подано в вигляді таблиць (табл. 1.6 – табл. 1.10).

Таблиця 1.6 – Результати оцінювання вибірки даних (прямі запити)

Діапазон ітерацій	Oracle, час вибірки, мс	MS SQL Server, час вибірки, мс
1000	0,003	0,009
2000	0,02	0,007
3000	0,02	0,008
4000	0,01	0,009
5000	0,004	0,007
6000	0,005	0,012
7000	0,02	0,01
8000	0,006	0,014
9000	0,004	0,007
10000	0,009	0,008
Середнє значення	0,0101	0,0091

Таблиця 1.7 – Результати оцінювання вставки даних (збережені процедури)

Діапазон ітерацій	Oracle, час вставки, мс	MS SQL Server, час вставки, мс
1000	0,009	0,005
2000	0,011	0,009
3000	0,008	0,008
4000	0,01	0,006
5000	0,004	0,006
6000	0,007	0,005
7000	0,02	0,006
8000	0,006	0,004
9000	0,007	0,009
10000	0,009	0,005
Середнє значення	0,0091	0,0063

Таблиця 1.8 – Результати оцінювання вибірки даних (прямі запити)

Діапазон ітерацій	Oracle, час вибірки, мс	MS SQL Server, час вибірки, мс
1000	0,007	0,004
2000	0,007	0,006
3000	0,005	0,005
4000	0,004	0,006
5000	0,007	0,005
6000	0,005	0,008
7000	0,008	0,004
8000	0,007	0,006
9000	0,009001	0,004
10000	0,007	0,008
Середнє значення	0,0065	0,0056

Таблиця 1.9 – Результати оцінювання вставки даних (збережені процедури)

Діапазон ітерацій	Oracle, час вставки, мс	MS SQL Server, час вставки, мс
1000	0,003	0,005
2000	0,006	0,003
3000	0,004	0,004
4000	0,004	0,007
5000	0,007	0,004
6000	0,002	0,008
7000	0,008	0,008
8000	0,007	0,005
9000	0,008	0,003
10000	0,005	0,004
Середнє значення	0,0054	0,0051

Таблиця 1.10 – Порівняння отриманих результатів та офіційних тестів

Назва операції	TPC Oracle	Oracle	TPC MS SQL Server	MS SQL Server
Вставка	2,575 TpsE	105 p/мс	3,141 TpsE	130 p/мс
Вибірка	123,323 QphH	153 p/мс	102,050 QphH	178 p/мс
Відношення коефіцієнтів по вставці	0,82		0,81	
Відношення коефіцієнтів по вибірці	1,2		0,85	

У результаті виконання проведено тестування СКБД Oracle та MS SQL Server, проаналізовано результати цього оцінювання. На основі результатів оцінювання якості СКБД було встановлено як співпадіння, так і розбіжності із результатами сучасних методик оцінювання.

1.4. Оцінка результатів порівняння NoSQL баз даних та MySQL

Оцінювання баз даних передбачає виконання ряду операцій над записами, зокрема, операцій зчитування і запису у середовищах Cassandra і MySQL. Cassandra показує кращий результат у порівнянні з MySQL.

Генератор робочого навантаження

На основі аналізу і тесту продуктивності одна програма вимагає безперервної роботи з більшою кількістю поточкових даних. Операції читання і запису виконуються в додатку на основі порівняльного аналізу.

Виконання тестів щодо визначення продуктивності виконується на основі двох типів тестів – запису та читання.

Під час запису виконується завантаження набору записів в обидві бази даних за допомогою підключення JDBC. Користувач створює різні потоки для завантаження даних паралельно з обома базами даних, такими як Cassandra, MySQL. Потік збільшує кількість пропускної здатності.

Під час зчитування відбувається вибірка даних із множини завантажених у БД, які можна отримати за допомогою простих запитів на основі «select».

Збір метрик продуктивності виконується шляхом ведення журналу логування під час запису даних у сховище. Контроль продуктивності забезпечується шляхом використання міток часу.

YCSB у NoSQL БД виконується із застосуванням командного рядка і створює потоки запитів до фреймворку. Оцінку продуктивності він реалізує на основі аналізу критеріїв пропускної здатності та виконаних операцій за секунду, а затримка виконує цю операцію на основі аналізу процесу запису.

У базі даних дозволяється генерувати нетривалу операцію запису для цього процесу, щоб дані були максимально швидкими. Результати порівняння щодо продуктивності запису даних у NoSQL БД Cassandra та реляційній БД MySQL показано у табл. 1.11. та на рис. 1.1.

Таблиця 1.11 – Результати порівняння продуктивності запису даних

Кількість записів	Cassandra(MS)	MySQL
100	1	5
200	2	9
500	4	19

Продовження таблиці 1.11

Кількість записів	Cassandra (MS)	MySQL
1000	8	43
10,000	60	400
100,000	456	3,000
200,000	918	6,000
500,000	2280	15,000
100,0000	4560	30,000

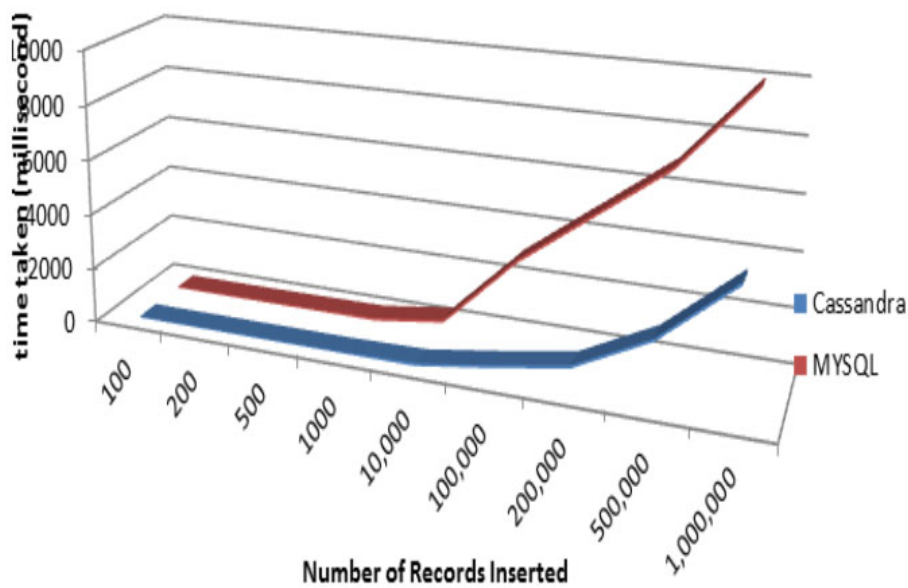


Рисунок 1.1 – Результати продуктивності запису у БД

Тривалість пошуку і читання записів є більшою в Cassandra, однак вона поступово зростає і в MySQL при збільшенні кількості записів, які розміщуються в апаратній конфігурації (табл. 1.12). Більше того, MySQL показує кращий результат у процесі пошуку порівняно з Cassandra.

Таблиця 1.12 – Продуктивність процесу зчитування даних

Кількість записів	Cassandra(MS)	MySQL(MS)
100	2	1
200	3	2
500	5	2
1,000	8	5
10,000	10	6
100,000	12	8
200,000	24	16
500,000	60	40
100,0000	120	80

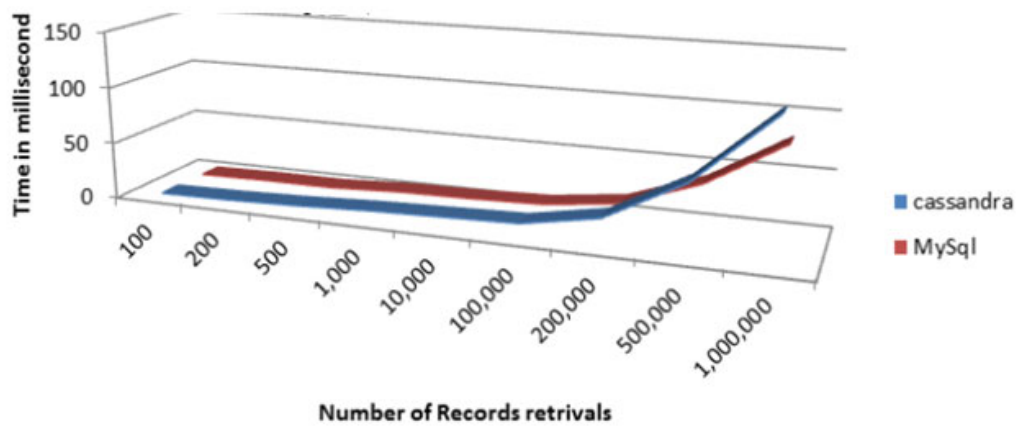


Рисунок 1.2 – Візуалізація результатів продуктивності зчитування даних

Багато показників залежать від правильності та ефективності організації і структури бази даних. MySQL не відповідає вимогам масштабованості та доступності реальних даних. База даних NoSQL підтверджує відповідність критеріям масштабованості, послідовності, доступності і відмовостійкості.

У даному розділі отримано наступні основні результати:

1. Проведено аналітичний огляд наукових праць і практичних порад щодо проектування та функціонування програмних систем з базами даних і знань, у результаті якого визначено роль і місце баз даних в інформаційній інфраструктурі організацій, що дало змогу обґрунтувати актуальність задач аналізу якості даних при прийнятті управлінських рішень.

2. Проаналізовано характеристики та особливості форматів представлення даних при використанні NoSQL та реляційних баз даних і знань, що дало змогу виявити фактори впливу на якість даних та в подальшому врахувати їх при побудові моделі аналізу та оцінювання якості.

3. Проведено порівняльний аналіз реляційних СКБД та NoSQL баз даних на предмет якості виконання операцій запису, зчитування та оновлення даних, а також відповідності показникам точності і продуктивності, у результаті чого встановлено, що ефективність їхнього використання залежить від кількості даних, цілей і задач, які передбачено програмною системою.

4.

2 ОБГРУНТУВАННЯ МОДЕЛІ ТА РОЗРОБКА МЕТОДУ АНАЛІЗУ ЯКОСТІ ДАНИХ ПРИ ФУНКЦІОНУВАННІ СИСТЕМ З БАЗАМИ ДАНИХ І ЗНАНЬ

2.1. Аналіз моделі стандарту ISO/IEC 25012 та його рекомендацій

Міжнародний стандарт ISO/IEC 25012 визначає загальну модель якості даних, що зберігаються в структурованому форматі комп'ютерної системи. Він фокусується на якості даних як частини комп'ютерної системи і визначає якісні характеристики для цільових даних, використовуваних людьми і системами.

Цільовими даними є ті, які потрібно аналізувати і перевіряти шляхом застосування моделі. Термін нецільових даних охоплює два випадки: перший відноситься до даних, які не є постійними (наприклад, дані обробляються операційною системою), а другий належить до даних, які можуть бути в рамках стандарту, але організація вирішує не застосовувати стандарт до них.

На рис. 2.1 зображено місце даних в комп'ютерній системі: вона може містити інформаційні системи, які в свою чергу можуть включати в себе одну або декілька комп'ютерних систем.

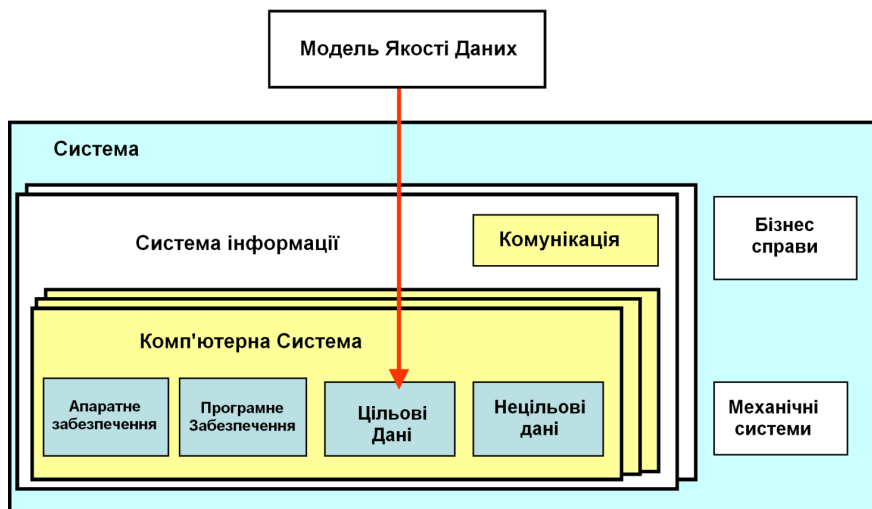


Рисунок 2.1 – Цільова область моделі якості даних

При використанні цього стандарту з іншими стандартами з ряду SQuaRE, можна встановити вимоги до якості даних, визначити метрики якості даних, а також здійснити планування і оцінку якості даних.

Вимоги до якості даних і їх вимірювання можуть бути класифіковані за характеристиками якості даних і використані в процесі оцінювання для аналізу даних незалежно від інших компонентів комп'ютерної системи.

Стандарт [1] спрямований на підтримку реалізації процесів життєвого циклу систем, що визначені у [4].

Цей стандарт враховує всі типи даних (наприклад, символьні рядки, текстові, дати, цифри, зображення, звуки), присвоєні значення даних і зв'язки між даними (наприклад, узгодженості між даними в тій же або в різних сутностях). Сфера застосування не включає дані, отримані з вбудованих пристроїв або датчиків реального часу, що не збереглися для подальшої обробки або історичних цілей.

Стандарт [1] не визначає фізичну організацію даних (системи управління базами даних), при цьому діяльність концептуальної, логічної та фізичної схеми проектування є за рамками даного стандарту. Усі процеси і результати, пов'язані з такими даними, отримують вигоду від застосування цього стандарту.

Відповідність даних вимогам проектування включені до даного стандарту. Модель якості даних, що представлена в міжнародному стандарті ISO/IEC 25012 призначена для об'єднання з іншими стандартами ряду міжнародних стандартів розроблених SQuaRE, що представлені на рис. 2.2 (адаптовано зі стандартом [5]).

ISO/IEC 2500n – «Розділ управління якістю». Міжнародні стандарти, що формують цей розділ, визначають всі спільні моделі, терміни та визначення щодо подальших, як і всіх інших, міжнародних стандартів ряду SQuaRE.

Довідкові документи і високий рівень практичних пропозицій в застосуванні правильних стандартів для спеціальних прикладних випадків надає допомогу всім типам користувачів. Розділ також визначає вимоги та інструкції

щодо підтримки функції управління специфікацією вимог та оцінювання програмного продукту.



Рисунок 2.2 – Організація міжнародних стандартів SQaRE

ISO/IEC 2501n – «Розділ моделі якості». Міжнародні стандарти, що формують цей розділ надають деталізовані моделі якості для програмних продуктів і даних. Крім того, в моделі якості програмного забезпечення внутрішніх і зовнішніх характеристик якості розкладаються на підхарактеристики. Також наведено посібник з практичного використання моделей.

ISO/IEC 2502n – «Розділ вимірювання якості». Міжнародні стандарти, які формують цей розділ містять метрики якості програмного продукту еталонної моделі, математичні визначення метрик якості і практичні інструкції для їх застосування. Представлені метрики для внутрішньої якості програми, зовнішньої якості програми і якості у використанні. Елементи вимірювань якості формують основу для подальших визначених і представлених метрик.

ISO/IEC 2503n – «Розділ якості вимог». Міжнародний стандарт, який формує цей розділ допомагає визначити вимоги якості. Ці вимоги можуть бути використані в процесі виявлення вимог до програмного продукту, що буде розроблений, або в якості норм в процесі оцінювання. Процес визначення вимог поєднується з технічними процесами, що визначені у міжнародному стандарті [4].

ISO/IEC 2504n – «Розділ оцінювання якості». Міжнародні стандарти, що формують цей розділ, забезпечують вимоги, рекомендації та керівні принципи для оцінювання програмного продукту. Підтримка для документування метрик якості модуля оцінювання також представлена у цьому розділі.

ISO / IEC 25050 – зарезервовані для SQuaRE стандартом ISO / IEC 25099 розширення міжнародних стандартів, технічних умов, загальнодоступних специфікацій (PAS) або технічних звітів.

Якість даних – ступінь, в якій характеристики даних задовольняють зазначені передбачувані потреби при використанні в зазначених обставинах [1].

Модель якості даних, що визначена у стандарті ISO/IEC 25012, викладає п'ятнадцять якісних характеристик відповідно з невід'ємної і залежної від системи точок зору.

Таблиця 1.1 – Характеристики моделі якості даних

Характеристики	Якість даних	
	Невід'ємна	Залежна від системи
Чіткість	+	
Повнота	+	
Послідовність	+	
Надійність	+	
Актуальність	+	
Доступність	+	+
Узгодженість	+	+
Конфіденційність	+	+

Ефективність	+	+
Точність	+	+
Відстежуваність	+	+
Зрозумілість	+	+
Готовність		+

Продовження таблиці 2.1

Характеристики	Якість даних	
	Невід’ємна	Залежна від системи
Портативність		+
Відновлюваність		+

Деякі характеристики є релевантними з обох точок зору. Характеристики якості даних будуть мати різну важливість і пріоритет для різних зацікавлених сторін. Характеристики якості даних розрізняють з точки зору невід’ємності, невід’ємної і залежної від системи точки зору, системозалежної точки зору. Застосування моделі якості даних буде також ефективною при побудові систем машинного навчання, оскільки дозволить підвищити точність і повноту одержання результатів при прогнозуванні та побудові експертних чи рекомендаційних систем.

2.2. Формалізація характеристик моделі якості даних стандарту ISO/IEC 25012

2.2.1. Характеристик якості даних з точки зору невід’ємності

Проведемо аналіз характеристик якості даних з точки зору невід’ємності. До них відносяться: чіткість, повнота, послідовність, надійність та актуальність.

Згідно з [1], чіткість – це відносна величина, в якій дані мають атрибути, які правильно представляють справжню цінність ймовірних атрибутів концепції чи події в певному контексті використання. Вона має два основних аспекти:

- синтаксичну чіткість;
- семантичну чіткість.

Синтаксична чіткість визначена як близькість значень даних до набору значень, визначених у розглянутій області, що є синтаксично коректною.

Наприклад, низький рівень синтаксичної чіткості тоді, коли слово «Марі» збережене як «Мері».

Семантична чіткість визначена як близькість значень даних до набору значень визначених у розглянутій області, що є семантично правильними.

Наприклад, низьким рівнем семантичної чіткості є збереження імені Джон як Джордж. Обидва імені синтаксично правильні згідно з областю використання, але Джордж – ім'я іншої особи.

Розглянемо зразок виміру якості невід'ємних даних. Великою якістю даних характеристики чіткості може виступити поле запису синтаксичної чіткості. Для виміру даної характеристики використовується функція

$$f = \frac{A}{B}, \quad (2.1)$$

де A – кількість записів із заданою синтаксичною чіткістю;

B – загальна кількість записів.

Повнота – це відносна величина, в якій предмет даних асоційований з сутністю, що має значення для всіх очікуваних властивостей і пов’язаних зразків сутностей в певному контексті використання.

Наприклад, для бази даних робітників повнота зменшується, якщо записи певних працівників не мають даних про те, як з ними можна зв’язатися у випадку нагальної потреби.

Величиною якості даних характеристики повноти може виступити повнота даних щодо файлу, при цьому для її вимірювання функція

$$f = \frac{A}{B}, \quad (2.2)$$

де A – кількість даних, потрібних для часткового опису у файлі даних;

B – кількість даних, потрібних в зазначеному контексті використання.

Як зазначено в [1], послідовність – це відносна величина, при якій дані не мають суперечностей і є логічно зв’язними з іншими даними в певному контексті використання. Це може бути для одного або обох об’єктів даних стосовно певної сутності серед схожих даних для порівняння сутностей.

Частковим випадком непослідовності є синоніми: потрібно використовувати словники термінів та визначень, щоб уникнути невідповідностей.

Наприклад, день народження працівника не може бути пізніше дати його прийняття на роботу.

Метрикою якості даних, що описує характеристику послідовність, може бути послідовність даних файлу. Для кількісного її вираження застосовується формула:

$$f = \frac{A}{B}, \quad (2.3)$$

де A – кількість послідовних даних у файлі;

B – кількість всіх даних у файлі.

Надійність – це відносна величина, в якій дані мають властивості, що розглядаються користувачами як істинні і правдоподібні в певному контексті використання.

Надійність включає концепцію вірогідності (правдивість джерел, встановлення авторства, зобов'язання).

Наприклад, дані, що засвідчені незалежною і довіреною організацією, повинні вважатися надійними.

Актуальність – це відносна величина, що визначає, скільки даних мають властивості потрібного віку в певному контексті використання.

Наприклад, розклад залізничної станції повинен оновлюватися з такою частотою, щоб при зміні часу прибуття чи зміні платформи пасажир міг потрапити на потрібний потяг.

Метрикою, для обчислення якості даних характеристики актуальність, може виступати метрика – актуальність значення поля даних, що обчислюється за формулою

$$f = \frac{A}{B}, \quad (2.4)$$

де A – кількісне значення полів даних, при яких значення величин даних

узгоджуються з поточними потребами;

B – загальна кількість полів даних.

2.2.2. Характеристик якості даних з невід’ємної і системозалежної точок зору

Розглянемо характеристики якості даних з невід’ємної і системозалежної точок зору. До них відносяться доступність, узгодженість, конфіденційність, ефективність, точність, відстежуваність та зрозумілість.

Доступність – це відносна величина, що відображає дані, до яких має доступ користувач, якому потрібні спеціальні технології або спеціальні налаштування через певні обмеження.

Наприклад, дані, які керуються читачем екрану (скрін рідером), не повинні зберігатися як зображення.

В якості метрики та величини характеристики доступності може виступити доступність до звукових даних. Для виміру даної характеристики використовується функція

$$f = \frac{A}{B}, \quad (2.5)$$

де A – кількість даних, збережених виключно як «звук» (без текстового представлення звуку);

B – кількість даних, представлених як звук.

Метрикою якості даних характеристики доступності може виступити багатоканальна доступність даних. Для виміру даної характеристики використовується функція

$$f = \frac{A}{B}, \quad (2.6)$$

де A – кількість даних, до яких користувачі з різними можливостями мають доступ;

B – загальна кількість доступних даних.

Узгодженість – це відносна величина, що вказує на те, які дані мають властивості, що задовольняють стандарти, конвенції чи діючі правила і подібні норми, маючи відношення до якості даних в певному контексті використання.

Наприклад, дані ризику кредитування банку повинні підпорядковуватися певним законам і стандартам.

Проведемо аналіз метрик якості невід’ємних даних і тих, що залежать від програмних систем. Великою якістю даних характеристики узгодженості може виступити недотримання закону про конфіденційність даних як значення (для невід’ємних даних) та як архітектура (для даних, залежних від систем). В обох випадках для виміру використовується функція

$$f = A, \quad (2.7)$$

де A – кількість пунктів, що не відповідають твердженням закону про

конфіденційність даних внаслідок вмісту даних (для невід’ємних даних) та кількість пунктів, що не задовольняють твердженням закону про конфіденційність через архітектурні збої (для даних, залежних від систем).

Конфіденційність – це відносна величина, що характеризує кількість властивостей даних, які є досяжними і зрозумілими лише для авторизованих користувачів в певному контексті використання [11, 21].

Наприклад: дані, що відповідають за персональну або конфіденційну інформацію (здоров'я, прибуток), повинні бути доступні тільки авторизованим користувачам, або записані в зашифрованому вигляді.

Конфіденційність – це аспект інформаційної безпеки (разом із доступністю і цілісністю), що визначена у стандарті ISO/IEC 13335-1:2004.

Розглянемо приклад метрик якості невід'ємних даних. метрикою якості даних характеристики конфіденційності може бути ступінь використання шифрування, який обчислюється як:

$$f = \frac{A}{B}, \quad (2.8)$$

де A – кількість зашифрованих полів бази даних;

B – кількість полів бази даних, котрі повинні бути зашифрованими.

Розглянемо зразок виміру якості даних, залежних від системи. Великою характеристикою якості даних характеристики конфіденційності може виступити невразливість. Для виміру даної характеристики використовується функція

$$f = 1 - \frac{A}{B}, \quad (2.9)$$

де A – кількість успішних проникнень під час формального тесту

проникнення;

B – кількість невдалих спроб проникнення.

Ефективність – це ступінь, яка характеризує властивості даних, що можуть бути оброблені, забезпечивши очікуваний рівень продуктивності та використовуючи оптимальну кількість ресурсів і типів ресурсів в певному контексті використання.

Наприклад, використання для даних більшого місця, ніж це необхідно, може призвести до марнування місця сховища, пам'яті та часу.

В якості вимірюваної величини щодо ефективності даних можуть виступити числа, збережені як стрічка, тобто:

$$f = A, \quad (2.10)$$

де A – кількість даних, збережених як string.

Перейдемо до метрик якості даних, які є залежними від системи. Метрикою якості даних характеристики ефективності може виступити марнування місця сховища. Для виміру даної характеристики використовується функція

$$f = \sum B - A, \quad (2.11)$$

де A – еталонна, середня кількість простору для ефективного зберігання

даних в базі;

B – використане місце для даних в будь-яких файлах бази даних.

Згідно з [1], точність – це ступінь, в якій дані мають властивості, що є строго відповідними або забезпечують розрізнення в певному контексті використання.

Наприклад, точність до п'ятого знаку після коми дає більшу функціональність, ніж точність до 2 знаку після коми.

Метрикою якості даних характеристики, що належить до класу невід'ємних від системи належить точність. Для вимірювання даної характеристики використовується функція

$$f = \frac{A}{B}, \quad (2.12)$$

де A – кількість значень даних з потрібною точністю;

B – загальна кількість значень даних.

До залежних від системи метрик, що описує точність, належить метрика точність полів бази даних. Формально визначення даної характеристики обчислюється за допомогою функції

$$f = \frac{A}{B}, \quad (2.13)$$

де A – кількість полів записів бази даних, визначених з потрібною точністю;

B – загальна кількість полів бази даних.

Відстежуваність – це ступінь, в якому дані мають атрибути, що надають відомості доступу до даних і про будь-які зміни, зроблені з даними в певному контексті використання.

Наприклад, органи державного управління повинні зберігати інформацію про здійснення доступу користувачами для визначення, хто переглядав чи редагував конфіденційні дані.

Метрикою якості даних характеристики відстежуваності може виступити відстежуваність значень даних. Для виміру даної характеристики використовується функція

$$f = \frac{A}{B}, \quad (2.14)$$

де A – кількість даних, для яких необхідна відстежуваність значень є

доступною;

B – загальна кількість значень для яких тестується відстежуваність.

Ще однією метрикою якості даних характеристики відстежуваності може виступити автоматична відстежуваність. При її вимірюванні та оцінюванні застосовується функція

$$f = A, \quad (2.15)$$

де A – кількість даних, що відслідковується автоматично (використовуючи

ресурси системи).

Зрозумілість – це ступінь, в якій дані мають атрибути, що дозволяють бути прочитаними і поясненими користувачами, висловлені в доречній мові, символами і одиницями мови для конкретного контексту використання. Деяка інформація про зрозумілість даних забезпечується метаданими.

Наприклад, для представлення області всередині держави звичайна аббревіатура буде зрозумілішою, ніж представлення числовим кодом.

Величиною, що інтерпретує характеристику зрозумілості при дослідженні якості даних може виступати зрозумілість головних даних завдяки наявним метаданим. Обчислення значення метрики даної характеристики забезпечує функція

$$f = \frac{A}{B}, \quad (2.16)$$

де A – кількість даних з метаданими в головних файлах даних;

B – загальна кількість даних головних файлів даних.

Інша метрика якості даних характеристики зрозумілості представляє та описує зрозумілість головних даних завдяки сполученим метаданим:

$$f = \frac{A}{B}, \quad (2.17)$$

де A – кількість полів записів, що мають автоматично прив'язані метадані

до відповідних даних;

B – загальна кількість полів.

2.2.3. Характеристик якості даних із системозалежної точки зору

До складу характеристики якості даних із системозалежної точки зору них відносяться: готовність, портативність та відновлюваність.

Готовність – це ступінь даних, що можуть бути надані уповноваженим користувачам чи додаткам, у певному контексті використання.

Наприклад, дані повинні бути доступними навіть при операціях резервного копіювання.

Частковим випадком готовності є одночасний доступ, для читання/запису даних, більше ніж одним користувачем чи програмою. Ще одним випадком готовності є здатність даних бути наявними в певний період часу.

В якості метрики для визначення кількісного значення характеристики готовності може використовуватись готовність одиниці даних, яка обчислюється за формулою:

$$f = \frac{A}{B}, \quad (2.18)$$

де A – кількість даних, що доступні під час резервного копіювання чи

відновлення;

B – кількість даних, що резервується чи відновлюється.

Портативність – це ступінь даних в певному контексті використання, що мають властивість при встановленні, заміні чи переміщенні з однієї системи на іншу зберігати наявну якість.

Метрикою якості даних характеристики портативності може виступати портативність даних, обчислити яку можна за формулою

$$f = \frac{A}{B}, \quad (2.19)$$

де A – кількість даних, що зберегли якість після перенесення на іншу комп'ютерну систему;

B – кількість перенесених даних.

Відновлюваність – це відносна величина даних, що мають властивості підтримувати і зберігати відповідний рівень операцій і якості, навіть у випадку відмови, в залежності від випадку застосування.

Відновлюваність може бути надана такими функціями, як проведення синхронізації (з іншими системами), створення резервних копій і здатність повертати проведені зміни.

Наприклад, при відмові медіа пристрою, дані, що збережені на ньому, повинні бути відновними.

Розглянемо зразок виміру якості даних характеристики відновлюваності, залежних від системи. Для виміру даної характеристики використовується функція

$$f = \frac{A}{B}, \quad (2.20)$$

де A – кількість даних, що були успішно резервовані чи відновлені;

B – загальна кількість даних, що резервували чи відновлювали.

Таким чином, застосування моделі якості даних стандарту ISO/IEC 25012 є ефективною при побудові систем, які використовують бази даних і знань, що дозволить підвищити показники продуктивності як самих комп'ютерних систем, так і ефективність прийняття рішень експертами.

2.3. Комплексна модель якості даних на рівні характеристик та атрибутів

Інформація вважаються повною тоді, коли вона містить мінімальний, але достатній для прийняття правильного рішення набір показників. Як неповна, так і надлишкова інформація знижує ефективність прийнятих на підставі такої інформації рішень [19].

Наприклад, для регулювання ходу виконання важливого замовлення може знадобитися інформація про його стан на кожній операції. В іншому ж випадку достатньо мати лише так звану «негативну» інформацію, що відображає негативне відхилення від заданої програми (плану).

Тому повнота інформації не зводиться до її кількості. Можна зібрати як завгодно багато даних, але при цьому ні на крок не просунутися до вирішення завдання. Звідси випливає, що повна інформація – це той її мінімально необхідний обсяг, який забезпечує реалізацію конкретної функції управління.

В якості атрибутів повноти як характеристики якості даних виділимо:

- розмір файлу;
- розміри в контексті слів і речень.

Дані вважаються послідовними в тому випадку, коли вони не мають суперечностей і є логічно зв'язними з іншими даними.

Наприклад, певні підхарактеристики сутності можуть бути представлені лише після того, як були представлені основні (батьківські) характеристики тої чи іншої сутності. Також користувач не може заповнити таблицю, доки не будуть заповнені всі необхідні словники та довідники.

В якості атрибутів послідовності як характеристики якості даних виділимо:

- наявність довідників та словників;
- порівняння дат.

Надійність інформації в першу чергу залежить від надійності носія інформації. Це так звана апаратна надійність, вона досягається технічними засобами [21].

Якісна характеристика надійності інформації розкривається через її правдиве подання, нейтральність, повноту.

В якості атрибутів надійності як характеристики якості даних виділимо:

- джерело одержання даних;
- експертну оцінку.

Актуальність даних – це ступінь відповідності даних поточному моменту часу. Із актуальністю, як і з повнотою, пов'язують комерційну цінність даних. Оскільки інформаційні процеси розтягнуті в часі, то достовірні і точні, але застарілі (неактуальні) дані системи машинного навчання можуть призводити до помилкових рішень.

В якості атрибутів актуальності як характеристики якості даних виділимо:

- частоту оновлення даних;
- частоту перевірки та зміни даних;
- стабільність оновлення даних.

Дані вважаються узгодженими в тому випадку, коли вони мають властивості, що задовольняють стандарти, діючі правила та інші норми щодо якості даних.

В якості атрибутів узгодженості як характеристики якості даних виділимо:

- відповідність стандартам конфіденційності;
- залежність від зовнішнього середовища.

Дані вважаються конфіденційними в тому випадку, коли доступ до них мають лише користувачі з відповідними правами доступу; такі дані не бувають загальнодоступними [23].

Задля дотримання конфіденційності даних системи машинного навчання необхідно реалізувати міри захисту від несанкціонованого доступу до даних, впливу на них, а також впливу на саму інформаційну систему.

В якості атрибутів конфіденційності як характеристики якості даних виділимо:

- наявність алгоритмів шифрування;
- наявність алгоритмів дешифрування;
- надійність методу шифрування;
- вразливість.

Дані вважаються ефективними в тому випадку, коли вони можуть бути оброблені, забезпечивши очікуваний рівень продуктивності та використовуючи доречну кількість ресурсів і типів ресурсів комп'ютерної інформаційної системи.

Наприклад, для зберігання символу у змінній не буде доречним використання стрічкового типу `string`, так як цілком достатньо використати символьний тип `char`. У великих масштабах виділення більшої кількості ресурсів, ніж це потрібно, буде негативно відображатись на ефективності даних та самої інформаційної системи.

В якості атрибутів ефективності як характеристики якості даних виділимо:

- відповідність розміру даних до виділеного для них місця у сховищі;
- відповідність типів даних значенням у файлі;
- наявність засобів підвищення пошуку.

Точність інформації означає, що вона є достатнім для вирішення конкретного завдання управління наближенням відображає ситуацію.

Наприклад, для вирішення завдань, що стосуються заробітної плати працівників, використовується інформація про тарифи, розцінки, ставки з округленням до копійок. Менша точність тут неприпустима. А при аналізі рентабельності, обсягу продажів, досить мати інформацію з точністю до гривень.

В якості атрибутів точності як характеристики якості даних виділимо:

- розрядність числових типів;
- відповідність значення даних їх типу (розміру);
- достовірність зв'язків між сутностями (атрибутами);
- точність обчислень.

Дані вважаються такими, які можна відстежити, в тому випадку, коли надаються відомості доступу до даних і про будь-які зміни, зроблені з даними під час їх роботи з ними.

Відстежуваність даних тісно пов'язана з іншими характеристиками, наприклад, із конфіденційністю. В системи машинного навчання обов'язково повинні бути реалізовані стеження за доступом до конфіденційних даних і фіксація їх змін.

В якості атрибутів вістежуваності як характеристики якості даних виділимо:

- логи;
- інтервал фіксації зміни даних.

Дані вважаються зрозумілими в тому випадку, коли вони представлені в доречній мові, символами і одиницями мови для конкретного користувача і його потреб. Цінність і якість інформації значно зменшується, коли вона представлена недоречно.

В якості атрибута зрозумілості як характеристики якості даних виділимо наявність опису метаданими.

Якщо комп'ютерна інформаційна система перебуває в стані стабільного функціонування, то а дані цієї системи можуть бути безперебійно надані користувачу з відповідними правами доступу, то готовність таких даних буде вважатись задовільною.

В якості атрибутів готовності як характеристики якості даних виділимо:

- можливість одночасного читання;
- можливість одночасного запису;
- заборона одночасного читання;
- заборона одночасного запису;
- здатність до резервування.

Дані вважаються портативними в тому випадку, коли вони зберігають свою якісь при переміщенні з однієї системи на іншу, або в межах компонентів однієї системи машинного навчання.

В якості атрибутів портативності як характеристики якості даних виділимо:

- можливість міграції даних;
- кількість даних, які однозначно інтерпретуються після міграції;
- кількість даних, які неоднозначно інтерпретуються після міграції.

Дані вважаються відновлюваними в тому випадку, коли вони зберігають свою якісь навіть у випадку відмови роботи інформаційної системи чи збою певних її компонентів.

В якості атрибутів відновлюваності як характеристики якості даних виділимо:

- резервування копій;
- швидка відміна внесених змін;
- синхронізація із зовнішніми засобами.

У табл. 2.1 наведено характеристики та відповідні атрибути якості даних при функціонування систем з базами даних знань.

Таблиця 2.1 – Модель якості даних

Характеристика	Атрибут
Повнота	Розмір файлу
	Розмір в контексті слів та речень
Послідовність	Наявність довідників та словників
	Порівняння датами
Надійність	Джерело одержання даних
	Експертна оцінка
Актуальність	Частота оновлення даних
	Частота перевірки та зміни даних
	Стабільність оновлення даних
Узгодженість	Відповідність стандартам конфіденційності
	Залежність від зовнішнього середовища
Конфіденційність	Наявність алгоритмів шифрування
	Наявність алгоритмів дешифрування
	Надійність методу шифрування
	Вразливість

Продовження таблиці 2.1

Характеристика	Атрибут
Ефективність	Відповідність розміру даних до виділеного для них місця у сховищі
	Відповідність типів даних значенням у файлі
	Наявність засобі підвищення пошуку
Точність	Розрядність числових типів
	Відповідність значення даних їх типу (розміру)
	Достовірність зв'язків між сутностями (атрибути)
	Точність обчислень
Відстежуваність	Логи
	Інтервал фіксації зміни даних
Зрозумілість	Опис метаданими

Готовність	Можливість одночасного читання
	Можливість одночасного запису
	Заборона одночасного читання
	Заборона одночасного запису
	Здатність до резервування
Портативність	Можливість міграції даних
	Кількість даних, які однозначно інтерпретуються після міграції
	Кількість даних, які неоднозначно інтерпретуються після міграції
Відновлюваність	Резервування копій
	Швидка відміна внесених змін
	Синхронізація із зовнішніми засобами

2.4. Розробка методу аналізу якості даних при функціонуванні систем з базами даних і знань

Визначення якості даних при функціонуванні систем з базами даних і знань запропоновано виконувати на основі методу, який передбачає реалізацію двох стадій: проектування оцінювання якості даних та безпосереднє вимірювання показників.

На рис. 2.1 продемонстровано алгоритм за яким проектується сценарій щодо аналізу якості.

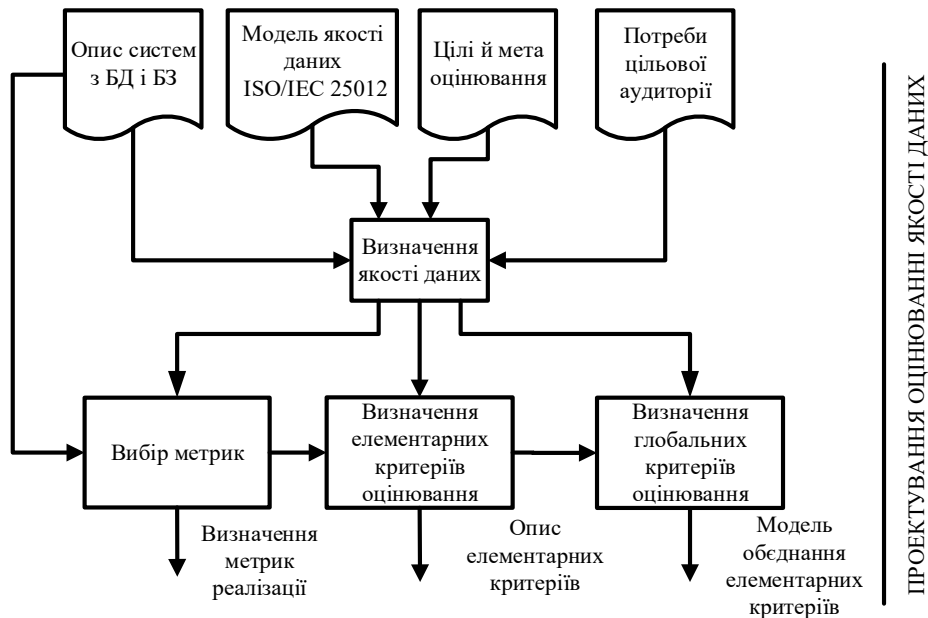


Рисунок 2.3 – Проектування як складова процесу оцінювання якості даних

Як видно з алгоритму, показаного на рис. 2.3, при проектуванні процедури аналізу якості даних необхідно володіти інформацією:

- про програмні системи, що використовують бази даних і знань та СКБД
- про потреби цільових користувачів щодо критеріїв якості даних з їхньої точки зору
- про цілі та мету аналізу якості даних
- про детальну структуру моделі якості даних, починаючи з рівня комплексних характеристик і закінчуючи метриками і мірами.

У даному випадку (рис. 2.1), множина елементарних критеріїв, які описують однакову сутність, формують часткові та/або комплексні показники якості даних. Результатом етапу проектування процесу аналізу якості даних є модель, на основі буде проводитись безпосереднє вимірювання кількісних показників якості даних.

Процес безпосереднього вимірювання якості даних при функціонування систем з БД і БЗ складається з 3-ох фаз:

- безпосереднє вимірювання показників якості на основі метрик;

- аналіз та оцінювання результатів елементарних атрибутів моделі якості даних;
- часткове та узагальнене оцінювання комплексу визначених показників.

На рис. 2.2 схематично представлено структуру безпосереднього вимірювання показників якості даних при функціонуванні систем з БД і БЗ.

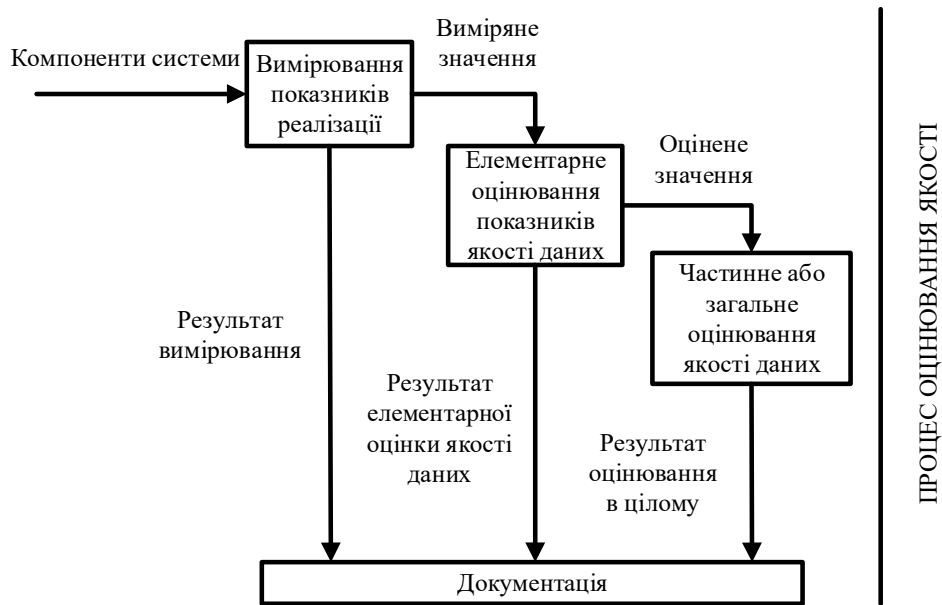


Рисунок 2.2 – Процес безпосереднього вимірювання та оцінювання якості даних

Основними показниками, значення яких вимірюють, як показано на рис. 2.2, є атрибути моделі якості даних. Такі вимірювання можуть бути проведені експертами вручну, або за допомогою автоматизованих інструментальних засобів, що визначається типом метрик і/або наявних апаратних ресурсів. Одержане кількісне значення оцінки атрибута не дає змоги робити висновок про те, чи відповідає він критеріям якості. Тому потрібно відобразити значення усіх метрик атрибутів на деяку проранжовану шкалу, яка представляю рівні якості даних.

Наприклад, нехай маємо атрибут, що інтерпретує кількість зв'язків ізольованих чи відсутніх даних у полях БД. В даному випадку, можна скористатися такою непрямою метрикою:

$$X = \frac{\text{Кількість пошкоджених записів}}{\text{Загальна кількість записів в БД}} \quad (2.21)$$

У результаті обрахунку одержимо конкретне кількісне значення. Для того, щоб інтерпретувати якість атрибута даних пропонується використати деяку елементарну функцію:

$$g(X) = \begin{cases} 1, \text{ якщо } X = 0; \\ \frac{X_{\max} - X}{X_{\max}}, \text{ якщо } 0 \leq X \leq X_{\max}; \\ 0, \text{ якщо } X \geq X_{\max}. \end{cases} \quad (2.22)$$

де $g(X)$ – елементарна функція для визначення оцінки;

X – метрика атрибута якості даних;

$X_{\max}$ – погоджена верхня границя, що характеризує допустимість задоволеності якості даних.

Крім того, при визначенні якості атрибута можна використовувати коефіцієнт прийнятності, тоді оцінку можна обчислити за формулою

$$q(X) = g(X) \cdot A \quad (2.23)$$

де $q(X)$ – локальний показник якості;

A – коефіцієнт прийнятності, визначений для деякого атрибута в результаті експертного аналізу.

Коефіцієнт прийнятності A виражає якість, доцільність, зручність досліджуваного атрибута з точки зору учасників експертного оцінювання.

Припустимо, що процедура комплексного оцінювання якості, використовує лінійну адитивну модель. Тоді визначити частинну або загальну якість даних можна, використовуючи інтегральний показник. Інтегральний показник якості обчислюємо за формулою:

$$Q = \sum_{i=1}^N q_i(X) \cdot k_i \quad (2.24)$$

де Q – інтегральна оцінка якості даних;

k_i – ваговий коефіцієнт атрибута якості даних;

N – кількість атрибутів.

Отже, інтегральна оцінка якості даних представляє собою суму частинних показників якості окремих атрибутів якості визначених у моделі якості даних, помножених відповідно на вагові коефіцієнти атрибутів.

Ваговий коефіцієнт k – це коефіцієнт, що вказує на важливість окремо взятого атрибута в побудованій моделі якості залежно від області застосування. Ранжування ваги атрибутів для конкретної предметної області відбувається шляхом визначення саме вагового коефіцієнту.

У даному розділі кваліфікаційної роботи одержано наступні результати:

1. Обґрунтовано необхідність використання моделі стандарту ISO 25012 для аналізу та оцінювання якості даних при функціонуванні систем з базами даних і знань, що дало змогу структурувати показники оцінювання та забезпечити повноту і ефективність результатів у визначеній прикладній предметній області.
2. Запропоновано математичне представлення структури моделі ISO 25012 у вигляді комплексу показників якості даних, що включає в себе три ієрархічні блоки і дає змогу оперувати якістю даних на різних рівнях, а також спрощує реалізацію процесу автоматизації аналізу даних при функціонуванні систем з базами даних і знань.
3. Визначено та запропоновано показники другого блоку моделі якості даних у вигляді атрибутів, що дало змогу забезпечити ефективність процесів

аналізу та управління якістю даних при функціонування програмних систем з БД і БЗ.

4. Створено метод аналізу та оцінювання якості даних при функціонуванні програмних систем з базами даних і знань, що забезпечує можливість представлення якості даних у кількісному вигляді та оперувати такими показниками як точність, ефективність, актуальність даних тощо.

3 ПРОГРАМНИЙ ЗАСІБ АНАЛІЗУ ЯКОСТІ ДАНИХ ПРИ ФУНКЦІОНУВАННІ СИСТЕМ З БАЗАМИ ДАНИХ І ЗНАНЬ

3.1. Обґрунтування технологій реалізації програмної системи

Засоби для виконання процесу забезпечення та оцінювання якості даних при проектуванні систем машинного навчання повинні бути швидкодіючими, гнучкими, мати здатність до модернізації, а також мати хороші показники ефективності доступу до даних.

В якості СКБД пропонується використовувати MySQL – реляційну СКБД у вільному доступі. MySQL являє собою ідеальне рішення для малих і середніх додатків.

Програмну реалізацію оцінювання якості даних пропонується виконувати мовою програмування PHP.

Головною перевагою мови PHP є практичність. PHP надає програмісту засоби для швидкого та ефективного вирішення поставлених завдань. Практичний характер мови PHP обумовлений п'ятьма важливими характеристиками:

- традиційність – PHP являється Сі-подібною мовою програмування, тому перехід з інших мов програмування на PHP не складе труднощів програмісту;
- простота – немає необхідності довантажувати бібліотеки чи вказувати спеціальні параметри компіляції.. Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?), і продовжує виконання до того моменту, коли не зустріне парну екранує послідовність (?>);
- ефективність – движок PHP дозволяє обробляти сценарії з достатньо високою швидкістю;
- безпека – PHP надає в розпорядження розробників і адміністраторів гнучкі та ефективні засоби безпеки, які умовно діляться на дві категорії: засоби системного рівня і засоби рівня додатків;

– гнучкість – оскільки PHP є вбудованою (embedded) мовою програмування, вона відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча PHP зазвичай рекомендується використовувати в поєднанні з HTML, вона з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови [15, 16].

Мову програмування PHP пропонується використовувати у зв'язці із мовою гіпертекстової розмітки HTML для графічного відображення розроблюваного засобу оцінювання якості даних.

3.2. Проектування та імплементація тестової бази даних

Для створення та адміністрування СКБД MySQL пропонується використати веб-застосунок з відкритим кодом phpMyAdmin.

Необхідно створити базу даних із назвою диплома для розроблюваної системи оцінки якості даних. Даний процес зображено на рис. 3.1.

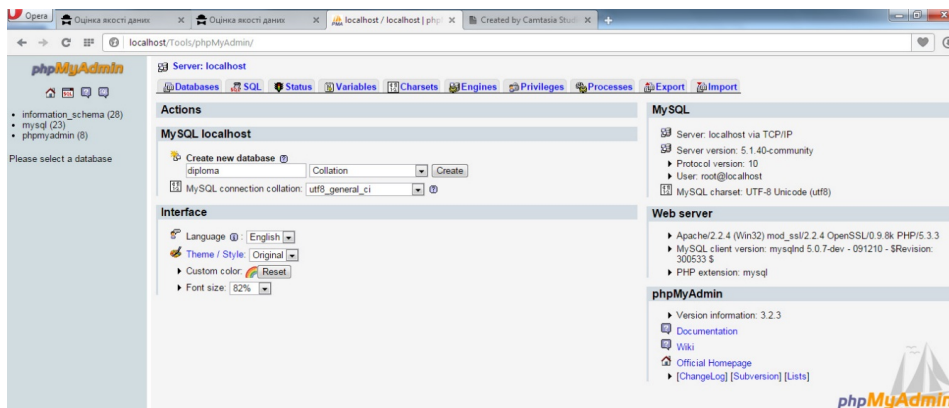


Рисунок 3.1 – Створення бази даних

Наступним кроком є створення таблиць у базі даних. Вирішено реалізувати дві таблиці із назвами «Студенти» (students) та «Успішність» (ratings). У першій будуть зберігатись загальні відомості про студентів університету, а друга

являтиме собою журнал успішності цих студентів. Процес створення таблиці students наведено на рис. 3.2.

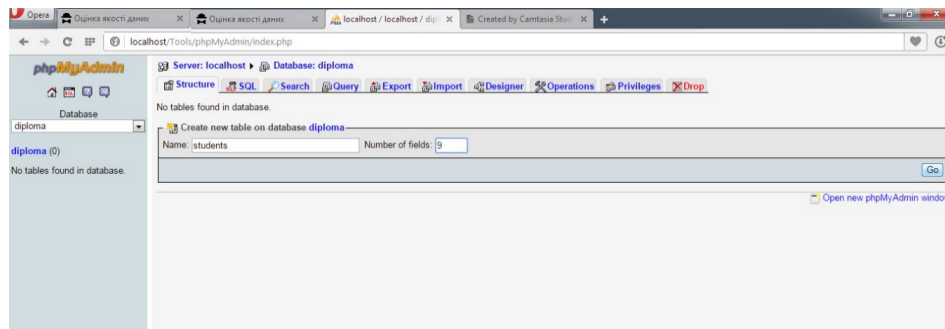


Рисунок 3.2 – Створення таблиці students

У таблиці 3.1 наведено опис таблиці students, а саме назви полів і відповідні їм типи SQL.

Таблиця 3.1 – Опис таблиці "Студенти"

Назва таблиці		Опис
Students		Таблиця містить відомості про студентів університету, а саме їх ППП, курс навчання, групу, домашню адресу, телефон та дату зарахування в університет
Назва поля	Тип поля SQL	
ID	int	
Name	varchar(25)	
Middlename	varchar(25)	
Lastname	varchar(25)	
Course	int	
Group	varchar(10)	
Address	varchar(50)	
Phone	varchar(10)	
Dateofentry	DateTime	

На рис. 3.3 зображено процес створення полів для таблиці students та присвоєння їм відповідних типів.

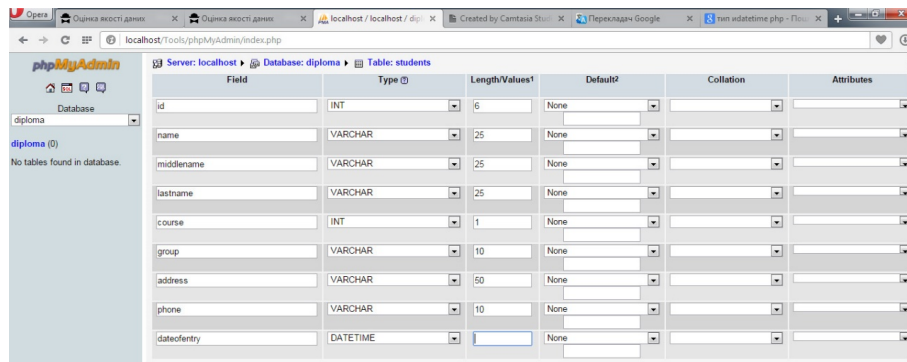


Рисунок 3.3 – Створення полів таблиці students

При створенні даної таблиці буде виконуватись наступний запит, що наведено у лістингу 3.1.

Лістинг 3.1. Запит на створення таблиці students

```
CREATE TABLE `diploma`.`students` (
  `id` INT( 6 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `name` VARCHAR( 25 ) NOT NULL ,
  `middlename` VARCHAR( 25 ) NOT NULL ,
  `lastname` VARCHAR( 25 ) NOT NULL ,
  `course` INT( 1 ) NOT NULL ,
  `group` VARCHAR( 10 ) NOT NULL ,
  `address` VARCHAR( 50 ) NOT NULL ,
  `phone` VARCHAR( 10 ) NOT NULL ,
  `dateofentry` DATETIME NOT NULL
) ENGINE = MYISAM ;
```

Після того, як таблицю було створено, необхідно заповнити її відповідними даними. На рис. 3.4 зображено створену і заповнену даними таблицю students.

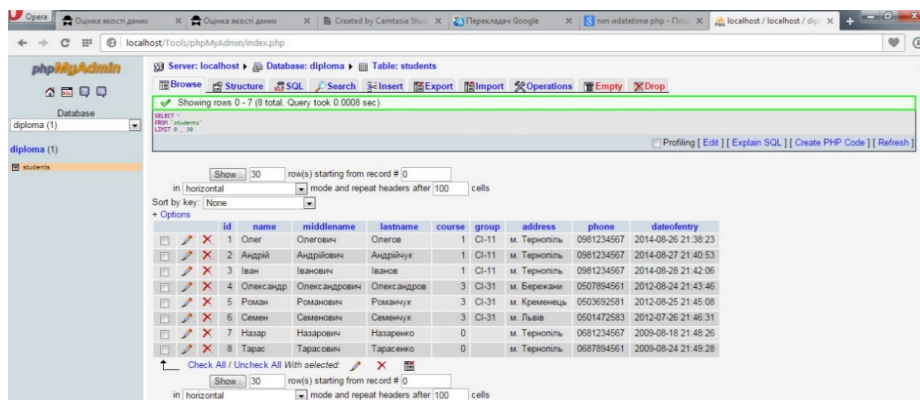


Рисунок 3.4 – Створена та заповнена таблиця students

Наступним кроком необхідно створити таблицю успішності студентів – ratings. У таблиці 3.2 наведено опис таблиці ratings, а саме назви полів і відповідні їм типи SQL.

Таблиця 3.2 – Опис таблиці "Успішність"

Назва таблиці		Опис
Ratings		Таблиця містить відомості про студентів університету, а саме їх ППІ, середній бал за час навчання, а також дату закінчення університету та отримання диплому
Назва поля	Тип поля SQL	
ID	int	
Name	varchar(25)	
Middlename	varchar(25)	
Lastname	varchar(25)	
Averaging	float	
Graduation	DateTime	

На рис. 3.5 зображено процес створення полів для таблиці ratings та присвоєння їм відповідних типів.

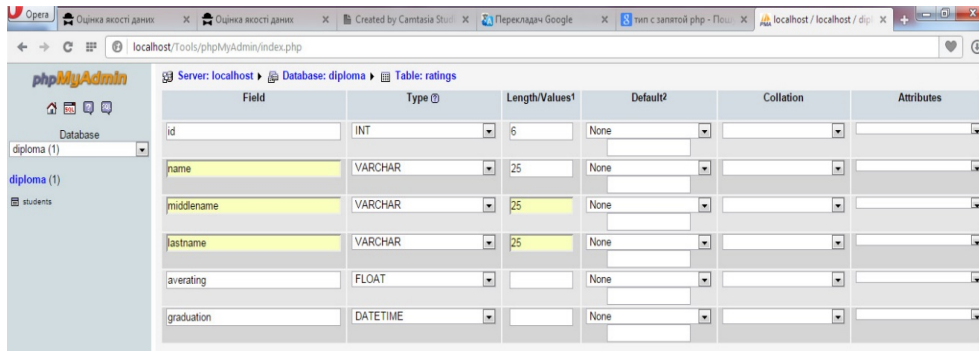


Рисунок 3.5 – Створення полів таблиці ratings

При створенні даної таблиці буде виконуватись наступний запит, що наведено у лістингу 3.2.

Лістинг 3.2. Запит на створення таблиці ratings

```
CREATE TABLE `diploma`.`ratings` (
  `id` INT( 6 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `name` VARCHAR( 25 ) NOT NULL ,
  `middlename` VARCHAR( 25 ) NOT NULL ,
  `lastname` VARCHAR( 25 ) NOT NULL ,
  `averating` FLOAT NOT NULL ,
  `graduation` DATETIME NOT NULL
) ENGINE = MYISAM ;
```

Після того, як таблицю було створено, необхідно заповнити її відповідними даними. На рис. 3.6 зображено створену і заповнену даними таблицю ratings.

Після успішного створення бази даних і таблиць, що в ній містяться, необхідно створити нового користувача для роботи із сервером і надати йому відповідні привілеї. Даний процес зображено на рис. 3.7.

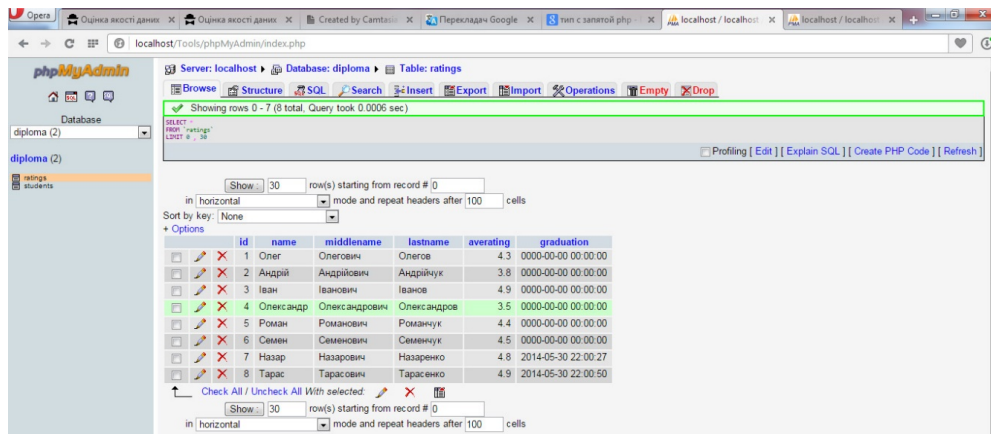


Рисунок 3.6 – Створена та заповнена таблиця ratings

Засіб phpMyAdmin надає візуальний інтерфейс для налаштування параметрів доступу до БД для користувача. В даному випадку користувач має права, які дозволяють формувати запити на вибірку даних, їх додавання, модифікацію та знищення. Окрім цього, користувач має змогу створювати, редагувати структуру таблиць БД, видаляти таблиці, тобто володіє повним інструментарієм для адміністрування.

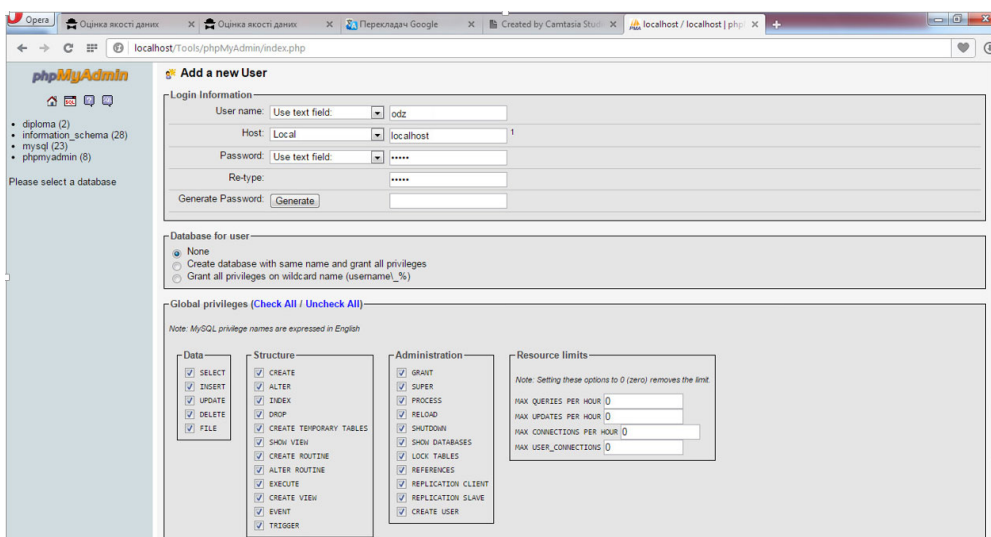


Рисунок 3.7 – Створення нового користувача та надання йому прав доступу

Створена база даних буде використовуватись як тестова при оцінюванні якості даних щодо прогнозування успішності студентів і формування їхнього рейтингу.

3.3. Реалізація системи оцінювання якості даних

Як зазначалось, система оцінювання якості даних розробляється за допомогою СКБД MySQL, мови програмування PHP та мови гіпертекстової розмітки HTML.

Основні вимоги, які висуваються до системи забезпечення та оцінювання якості даних при функціонуванні систем з БД і БЗ є автоматизація процесів передбачених розробленим методом. В загальному випадку, функціональні вимоги до системи забезпечення та оцінювання якості даних можна представити у вигляді Use Case діаграми.

При цьому користувачем системи буде проектний менеджер або керівник групи розробки систем машинного навчання. Він повинен сформувати набір критеріїв якості даних для конкретної предметної області, виходячи із структури даних і джерел їх отримання.

Після цього провести класифікацію атрибутів за характеристиками моделі якості даних і вибрати метрики для оцінювання відповідності вхідних даних до тих, які будуть одержуватись в процесі машинного навчання.

На рис. 3.8 представлено діаграму варіантів використання, на якій показано основні функціональні можливості засобу забезпечення та оцінювання якості даних при проектуванні систем машинного навчання.

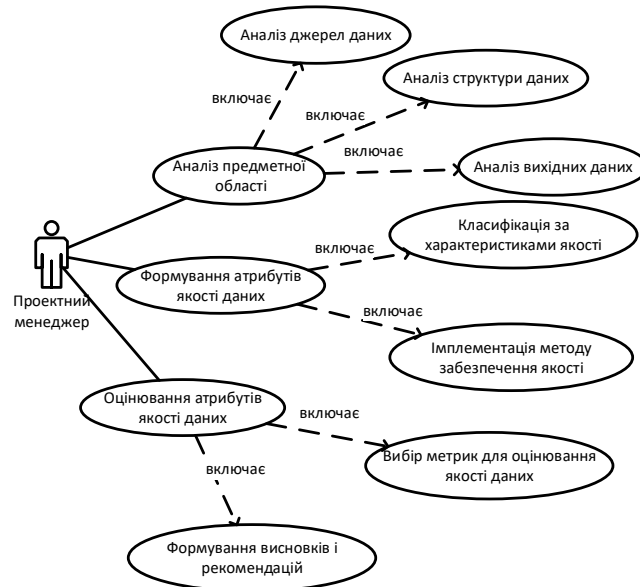


Рисунок 3.8 – Функціональні вимоги до системи забезпечення та оцінювання якості систем машинного навчання

Архітектура засобу забезпечення та оцінювання якості за шарами Фаулера наведена на рис. 3.9.

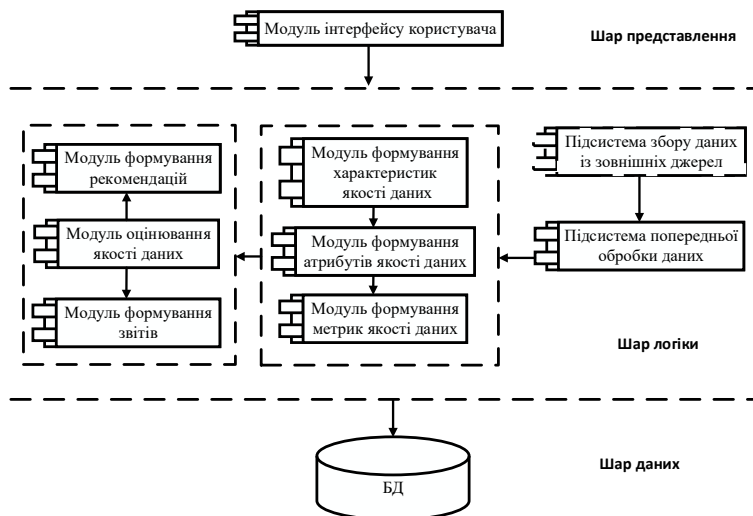


Рисунок 3.9 – Архітектура засобу забезпечення та оцінювання якості даних за шарами Фаулера

Загальну організацію засобу забезпечення та оцінювання якості даних на основі web-технологій реалізації зображено на рис. 3.10.

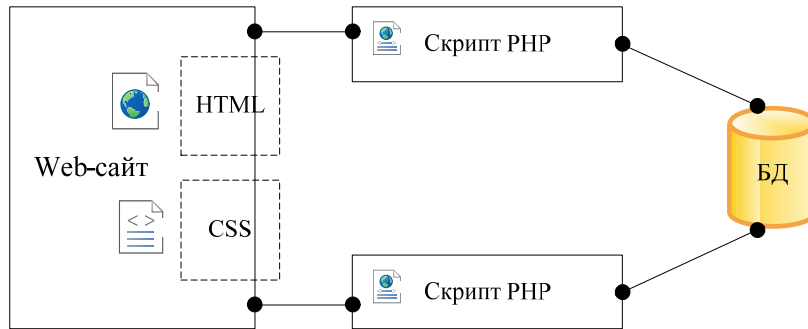


Рисунок 3.10 – Організація засобу забезпечення та оцінювання якості даних

Спочатку необхідно розробити зовнішній вигляд системи оцінки якості даних. Для цього пропонується використати мову гіпертекстової розмітки HTML та каскадні таблиці стилів CSS. У лістингу 3.3 наведено розроблену html-структуру для відображення сторінки.

Лістинг 3.3. Лістинг відображення головної сторінки системи оцінювання якості даних

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

  <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
  <head>
    <title>Оцінка якості даних</title>
    <link rel="stylesheet" type="text/css" href="style.css">
    <link
href="http://fonts.googleapis.com/css?family=Cuprum&subset=latin,cy
rillic" media="all" rel="stylesheet">
  </head>

```

```

<body>
<div class="main" align="center">
<div class="forma">
    <p class="title">Оцінка якості даних БД по
характеристиках:</p>
    <table cellpadding="0" cellspacing="0" border="0"
width="90%" align="center">
        <tr>
            <td width="50%">
                <p class="charact">Точність</p>
            </td>
            <td width="50%">
                <p class="charact">Послідовність</p>
            </td>
        </tr>
        <tr>
            <td width="50%">
                <form action="accuracy.php" method="post">
                    <center><input type="button" name="my_button" value="Оцінити
точність" class="go_work"></center>
                </form>
            </td>
            <td width="50%">
                <form action="sequence.php" method="post">
                    <center><input type="button" name="my_button" value="Оцінити
послідовність" class="go_work"></center>
                </form>
            </td>
        </tr>
    </table>
</div>
    <p class="copyright">&copy; 2014 :Журихін Юрій Олегович, ТНТУ,
гр. СІМ-61</p>
</div>

```

У лістингу 3.4 наведені відповідні класи створеної каскадної таблиці стилів CSS для відображення сторінки.

Лістинг 3.4. Лістинг класів каскадної таблиці стилів

```
body {
background-color:#f4f4f4;
}
.main {
position:relative; margin:0 auto; overflow:hidden;
}
.forma {
position:relative; width:600px; overflow:hidden; padding:0 0
30px 0; border:2px solid #1476c4; border-radius:10px; background-
color:#fff;
}
.title {
font-family:Cuprum; color:#1476c4; font-size:28px; text-
align:center;
}
.charact {
font-family:Cuprum; color:#1476c4; font-size:22px; text-
align:center;
}
.copyright {
font-family:Cuprum; color:#1476c4; font-size:14px; text-
align:center;
}
.go_work {
background-color: transparent;cursor: pointer; width:
190px;height: 40px; border: 0px; display:inline-block; text-
decoration:none; font-family:Cuprum; font-size:18px; padding: 4px 0px
4px 0px; border-radius:5px; background: linear-gradient(to top,
#85bfee, #47a5ef); color:#fff; border:1px solid #e8e9eb;
}
```

Відображення створеної сторінки показано на рис. 3.11.

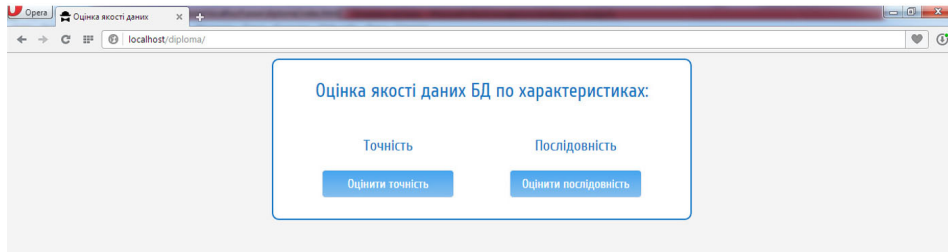


Рисунок 3.11 – Відображення головної сторінки системи

Як видно на рис. 3.11, проводити оцінювання якості даних було вирішено у відповідності до двох характеристик якості стандарту ISO/IEC 25012: точності та послідовності. В якості атрибутів було обрано виділені раніше відповідність значення даних типу поля (для характеристики точності) та порівняння датами (для характеристики послідовності).

Як раніше зазначалось, для реалізації вищезгаданих методів було обрано мову програмування PHP.

Для початку роботи з БД необхідно провести з'єднання із MySQL сервером і занесення результату у змінну, вибрати базу даних, з якою буде вестись робота:

```
<?php
$db = mysql_connect ("localhost","zhuryhin","12345");
mysql_select_db ("diploma",$db);
?>
```

Для реалізації оцінювання характеристики точності даних була здійснена перевірка на відповідність значення внесених у таблицю даних їх типу поля. Програмна реалізація даної функції наведена у лістингу 3.5.

Лістинг 3.5. Оцінка точності даних

```
<?php
$result = mysql_query ("SELECT * FROM students",$db);
$myrow = mysql_fetch_array($result);
```

```

$phone_count = 0;
$a = mysql_query("SELECT COUNT(1) FROM students",$db);
$rows = mysql_fetch_array($a);

do
{
    if (ctype_digit($myrow[0])) {
        $phone_count = $phone_count + 1;
    } else {
        $phone_count = $phone_count;
    }
}
while($myrow = mysql_fetch_array($result));

if ($phone_count = $rows) {
    echo '<p class="title">Досліджувані дані відповідають
вимогам характеристики точності.</p>';
} else {
    echo '<p class="title">Досліджувані дані не відповідають
вимогам характеристики точності.</p>';
}
?>

```

Результат виконання даної PHP-функції зображено на рис. 3.12.

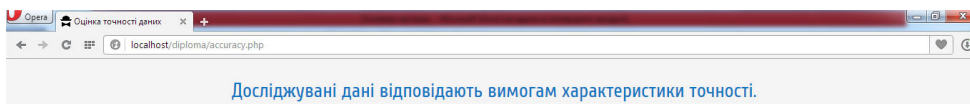


Рисунок 3.12 – Результат оцінки точності даних

Для реалізації оцінки характеристики послідовності була здійснена перевірка шляхом порівняння полів двох таблиць, які повертають дату. У таблиці students поле Dateofentry повертає дату та час зарахування студента до університету. Відповідно, у таблиці ratings поле Graduation повертає дату та час

закінчення студентом університету та отримання диплому. Програмна реалізація даної функції наведена у лістингу 3.6.

Лістинг 3.6. Оцінка послідовності даних

```
<?php
$result = mysql_query ("SELECT dateofentry FROM students WHERE
lastname='Назаров'", $db);
$myrow = mysql_fetch_array($result);

$result2 = mysql_query ("SELECT graduation FROM ratings WHERE
lastname='Назаров'", $db);
$myrow2 = mysql_fetch_array($result2);

if ($myrow[0] < $myrow2[0])
{
    echo '<p class="title">Досліджувані дані відповідають вимогам
характеристики послідовності.</p>';
}
else {
    echo '<p class="title">Досліджувані дані не відповідають вимогам
характеристики послідовності.</p>';
}
?>
```

Результат виконання даної PHP-функції зображено на рис. 3.13.

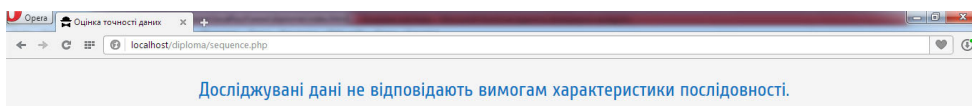


Рисунок 3.13 – Результат оцінки послідовності даних

Як бачимо, досліджувані дані не є послідовними, а значить не відповідають вимогам характеристики послідовності. Причиною цього є помилки в заповненні

полів однієї або іншої таблиці БД. Цілком зрозуміло, що студент не може закінчити університет і отримати диплом швидше, аніж бути зачисленим в цей університет.

Після виправлення помилок у таблицях проводимо повторну оцінку даних на предмет послідовності. Результат виконання зображено на рис. 3.14.

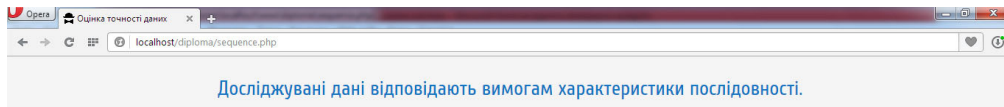


Рисунок 3.14 – Результат повторної оцінки послідовності даних після виправлення помилок у БД

Основні результати даного розділу полягають у виборі технології та засобів реалізації системи оцінювання якості даних, проектуванні та реалізації БД і архітектури системи забезпечення та оцінювання якості даних, проведенні експериментальних досліджень та одержанні результатів роботи системи, що дало змогу покращити першочергові дані і підвищити їх якість.

Основні результати даного розділу полягають в наступному:

1. Обґрунтовано застосування мови програмування PHP і стеку технологій HTML/CSS, СКБД MySQL для програмної імплементації системи аналізу якості даних і визначено їх основні переваги над іншими технологіями.
2. Спроектовано архітектуру програмної системи і схему бази даних, що враховує вимоги предметної області та особливості моделі і методу аналізу якості даних і дає змогу оцінити характеристики точності і послідовності стандарту ISO 25012.
3. Експериментально підтверджено дієвість та ефективність розробленої системи аналізу якості даних при функціонуванні систем з базами даних і знань, яка враховує особливості предметної області та структуру моделі ISO 25012.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Метою дипломної роботи магістра є дослідження методів і засобів аналізу якості даних при функціонуванні систем з базами даних і знань. Застосування електронних та електронно-обчислювальних пристроїв вимагає дотримання відповідних правил, норм з охорони і гігієни праці, а також правил техніки безпеки.

Для ефективної і безпечної роботи колективів працівників з розробки програмного забезпечення, у тому числі і фахівців з аналізу якості вимог, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці [25, 26].

Окрім цього, робоче місце програміста чи фахівця з комп'ютерних систем, повинно бути обладнаним з дотримання вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Згідно НПАОП 0.00-7.15-18 приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до вимог:

- переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації у відповідності до НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні»;

- Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом Держбуду України, з димовими пожежними сповісниками та переносними вуглекислотними вогнегасниками та ДСТУ Б.В.1.1-36:2016 „Визначення

категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою”.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами Типових норм належності вогнегасників, затверджених наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи, і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог ДБН В.2.5-56:2014.

Приміщення, в яких розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.1.1-7-2016 „Пожежна безпека об’єктів будівництва”, ДБН В.2.5-56:2014, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними. Лінія електромережі для живлення комп’ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Комп’ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з’єднань і електророзеток заводського виготовлення.

У штепсельних з’єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж

Примечание [1]: не діючий

присєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним.

Не допускається підключати комп'ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення комп'ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При експлуатації програмної системи підтримки процесу аналізу якості даних, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18.

Організація робочого місця фахівця із запровадження або аналізу якості даних при функціонування систем з базами даних і знань повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам.

Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами НПАОП 0.00-7.15-18.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп'ютера, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців при аналізі якості даних при функціонуванні систем з базами даних і знань.

4.2. Фактори, що впливають на функціональний стан користувачів комп'ютера

Технологічний процес обробки та зберігання інформації вимагає високої культури виробництва, особливої чистоти повітря виробничого середовища. При цьому запиленість і загазованість негативно впливає на точність і надійність роботи електронного устаткування, а також на якість технічних пристроїв.

На підприємствах інформаційної обробки даних потрібний захист від пилоутворення, надійна звукоізоляція між виробничими приміщеннями, а також оптимальні санітарно-гігієнічні умови (мікроклімат, освітлення, опалювання, вентиляція та ін.).

Джерелами шуму на підприємствах такого типу є самі обчислювальні машини (вбудовані в стійки ЕОМ вентилятори, принтери і так далі), центральна система вентиляції і кондиціонування повітря й інше устаткування.

У виробничих приміщеннях, в яких робота на відеодисплейних терміналах (ВДТ) і персональних комп'ютерах (ПК) є допоміжною, рівні шуму на робочих місцях не повинні перевищувати значень, встановлених для цих видів робіт санітарними нормами допустимих рівнів шуму на робочих місцях.

При виконанні основної роботи на ВДТ і ПК (диспетчерські, операторські, розрахункові кабінки і пости управління, зали обчислювальної техніки та ін.) рівень шуму на робочому місці не повинен перевищувати 50 дБА.

У приміщеннях, де працюють інженерно-технічні працівники, що здійснюють лабораторний, аналітичний або вимірювальний контроль, рівень шуму не повинен перевищувати 60 дБА.

У приміщеннях операторів ПК (без дисплеїв) рівень шуму не повинен перевищувати 65 дБА. На робочих місцях в приміщеннях, де розміщені шумні агрегати обчислювальних машин, рівень шуму згідно СанПіН 2.2.2. 542-96 не повинен перевищувати 75 дБА.

Ще однією групою факторів, які негативно можуть впливати на людину є мікрокліматичні параметри. Мікрокліматичні параметри виробничого середовища - це поєднання температури, відносної вологості і швидкості руху повітря.

Ці параметри значною мірою впливають на функціональну діяльність людини, її самопочуття, здоров'я, а також і на надійність роботи обчислювальної техніки. Причому у виробничих умовах характерна сумарна дія мікрокліматичних параметрів.

Великий вплив на мікроклімат в приміщеннях підприємства здійснюють джерела тепла. До них відноситься: обчислювальне устаткування, прилади освітлення, обслуговуючий персонал, а також сонячна радіація. Причому найбільші сумарні тепловиділення серед приміщень підприємства мають зали з великою кількістю ПК, а в них основним тепловиділяючим устаткуванням є ПК, які дають в середньому до 80% сумарних тепловиділень. Від приладів освітлення тепловиділення складають в середньому 12%, від обслуговуючого персоналу - 1%, від сонячної радіації - 6%. Приплив тепла через непрозорі конструкції, що захищають, складає 1%.

На організм людини і роботу устаткування великий вплив чинить відносна вологість повітря. При вологості повітря до 40% стає крихкою основа магнітної стрічки, підвищується знос магнітних голівок, виходить з ладу ізоляція дрітків, а також виникає статична напруга при русі носіїв інформації в ПК.

Для зниження концентрації пилу в приміщеннях з електронним устаткуванням потрібно, щоб обслуговуючий персонал працював в халатах і легкому змінному взутті. Запиленість в приміщеннях з ПК не повинна перевищувати $0,5 \text{ мг/м}^3$. Забороняється палити, оскільки частки попелу, осідаючи на поверхнях магнітних носіїв, викликають збій в роботі ПК.

Устаткування, рівні шуму якого перевищують нормовані, повинно знаходитися поза приміщенням з ВДТ і ПК. Шум в машинних залах знижують, послабляючи шуми самих джерел і спеціальними архітектурно-будівельними рішеннями.

Додатковими заходами щодо шумопогашення в машинних приміщеннях можуть бути:

- пристрій підвісної стелі, яка служить звукопоглинальним екраном;
- використання звукопоглинальних матеріалів з максимальними коефіцієнтами звукопоглинання в області частот 63-8000 Гц для обробки приміщень;
- зменшення площі скляних огорожень і віконних отворів;
- встановлення особливо шумлячих пристроїв на пружні подушки;
- застосування на робочих місцях звукопоглинаючих екранів;
- використання однотонних завіс з щільної тканини, підвішених в складку на відстані 15-20 см від огороження. Ширина завіси має бути в 2 рази більше ширини вікна.

У виробничих приміщеннях, в яких робота на ВДТ і ПК є допоміжною, температура, відносна вологість і швидкість руху повітря на робочих місцях повинні відповідати діючим санітарним нормам мікроклімату виробничих приміщень.

Для підтримки відповідних мікрокліматичних параметрів на підприємствах використовуються системи опалювання і вентиляції, а також проводиться кондиціонування повітря в приміщеннях.

Система опалювання забезпечує достатньо постійне і рівномірне нагрівання повітря в приміщеннях в холодний період року, а також пожежо- і вибухобезпечність. При цьому коливання температури протягом доби не повинні перевищувати 2-3 С°; у горизонтальному напрямі - 2 С° на кожен метр довжини, у вертикальному – 1 С° на кожен метр висоти приміщення.

Систему опалювання розраховують на відшкодування втрат теплоти через конструкції будівлі, на нагрів проникаючого в приміщення холодного повітря і матеріалів, що надходять ззовні, устаткування.

Для забезпечення встановлених норм мікрокліматичних параметрів і чистоти повітря в приміщеннях з ПК застосовують вентиляцію. Проектування системи вентиляції припускає визначення витрати повітря для вентиляції

приміщення і охолодження ПК, складання принципової схеми вентиляції приміщення та аеродинамічного розрахунку повітровідводів, вибір повітрозабірних і повітророзподільних пристроїв.

Мінімальна витрата повітря визначається з розрахунку 50-60 м³/год на одного працівника.

Умови по повітрообміну (за 1 годину) наступні:

- двох-трьохкратний - в приміщенні з ПК, в приміщеннях підготовки технічних носіїв, рахункового контролю, в групах прийому і випуску продукції, службах технічного обслуговування устаткування, в архівах;
- п'ятикратний - в приміщеннях копіювання і оформлення документації;
- півторакратний - в інших приміщеннях.

Кондиціонування повітря потрібне в приміщеннях з ПК, приміщеннях для зберігання носіїв інформації, приміщеннях для розміщення сервісної і периферійної апаратури, приміщеннях прийому і видачі інформації.

Отже, на підприємствах інформаційної обробки даних потрібно впроваджувати заходи захисту від таких факторів впливу виробничого середовища, як пил, шум, електромагнітне випромінювання, дотримуватись і забезпечувати відповідність мікроклімату, освітлення та інших показників діючим санітарно-гігієнічним умовам (опалювання, вентиляція та ін.).

ВИСНОВКИ

У дипломній роботі магістра одержано наступні наукові і практичні результати.

1. Проведено аналітичний огляд наукових праць і практичних порад щодо проектування та функціонування програмних систем з базами даних і знань, у результаті якого визначено роль і місце баз даних в інформаційній інфраструктурі організацій, що дало змогу обґрунтувати актуальність задач аналізу якості даних при прийнятті управлінських рішень.

2. Проаналізовано характеристики та особливості форматів представлення даних при використанні NoSQL та реляційних баз даних і знань, що дало змогу виявити фактори впливу на якість даних та в подальшому врахувати їх при побудові моделі аналізу та оцінювання якості.

3. Проведено порівняльний аналіз реляційних СКБД та NoSQL баз даних на предмет якості виконання операцій запису, зчитування та оновлення даних, а також відповідності показникам точності і продуктивності, у результаті чого встановлено, що ефективність їхнього використання залежить від кількості даних, цілей і задач, які передбачено програмною системою.

4. Обґрунтовано необхідність використання моделі стандарту ISO 25012 для аналізу та оцінювання якості даних при функціонуванні систем з базами даних і знань, що дало змогу структурувати показники оцінювання та забезпечити повноту і ефективність результатів у визначеній прикладній предметній області.

5. Запропоновано математичне представлення структури моделі ISO 25012 у вигляді комплексу показників якості даних, що включає в себе три ієрархічні блоки і дає змогу оперувати якістю даних на різних рівнях, а також спрощує реалізацію процесу автоматизації аналізу даних при функціонуванні систем з базами даних і знань.

6. Визначено та запропоновано показники другого блоку моделі якості даних у вигляді атрибутів, що дало змогу забезпечити ефективність процесів

аналізу та управління якістю даних при функціонування програмних систем з БД і БЗ.

7. Створено метод аналізу та оцінювання якості даних при функціонуванні програмних систем з базами даних і знань, що забезпечує можливість представлення якості даних у кількісному вигляді та оперувати такими показниками як точність, ефективність, актуальність даних тощо.

8. Обґрунтовано застосування мови програмування PHP і стеку технологій HTML/CSS, СКБД MySQL для програмної імплементації системи аналізу якості даних і визначено їх основні переваги над іншими технологіями.

9. Спроектовано архітектуру програмної системи і схему бази даних, що враховує вимоги предметної області та особливості моделі і методу аналізу якості даних і дає змогу оцінити характеристики точності і послідовності стандарту ISO 25012.

10. Експериментально підтверджено дієвість та ефективність розробленої програмної системи аналізу якості даних при функціонуванні систем з базами даних і знань, яка враховує особливості предметної області та структуру моделі ISO 25012.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ISO/IEC 25012. Software engineering – Data quality model
2. ДСТУ 2844 -94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. Чинний від 01.01.96. К. Держстандарт України. 1995. - 15 с.
3. ДСТУ ISO 9001 – 2001. Системи управління якістю. Вимоги. Чинний від 27.06.2001. К. Держстандарт України. 2001.23 с.
4. Харченко О., Яцишин В. Розробка та керування вимогами до програмного забезпечення на основі моделі якості. Том 14. 2009. №1. с. 201-207
5. Бабенко Л., Лаврищева К. Основи програмної інженерії: Навч. посіб./ Л. Бабенко К.: Т-во "Знання", КОО, 2001 – с. 269.
6. Бозм Б. Инженерное проектирование информационных систем. М.: Радио и связь. 1985. 512 с.
7. Грищенко В., Лаврищева Е. Методы и средства компонентного программирования. Кибернетика и системный анализ. №1. 2003. с.39–55.
8. Аткинсон Л. PHP 5. Библиотека профессионала. Изд-во «Вильямс». 2006 . 944 с.
9. Конверс Т. PHP 5 и MySQL. Библия пользователя. Изд-во «Вильямс» – 2009. 1216 с.
10. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными.
11. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
12. ГОСТ Р 51170-98 Качество служебной информации. Термины и определения
13. ДСТУ 2844 -94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. Чинний від 01.01.96. К. Держстандарт України. 1995. 15 с.

14. Коваль Г., Коротун Т., Яблокова Т., Кузаченко Л. Планирование обеспечения надежности информационных систем. Проблемы программирования. 2001. №3-4. с. 40 – 47.
15. Барсегян А., Куприянов М., Степаненко В., Холод И. Технологии анализа данных /– СПб. : Изд-во " БХВ-Петербург".2008. 384 с.
16. Акулов О. Информатика. Базовый курс.2-е изд. М.: 2005. 552 с
17. ГОСТ Р 51275-2006 Защита информации. Объект информации. Общие положения
18. Вендров А. CASE-технологии: Современные методы и средства проектирования информационных систем. М. : Изд-во "Финансы и статистика", 1998. 176 с.
19. Замковий В., Райчев І., Харченко О. Принципи проектування відкритих розподілених систем. «НАУ-друк». 2010. 240 с.
20. Саати Т. Принятие решений. Метод анализа иерархий. М.: Радио и связь. 1993. 315 с.
21. Кузнецов М.В. MySQL 5. Наиболее полное руководство. БХВ-Петербург. 2017. 1007 с.
22. НАПБ А.01.001-2004 «Правила пожежної безпеки в Україні».
23. Стеблюк М.І. Цивільна оборона та цивільний захист: Підручник. — 2-ге вид., переробл. — К.: Знання, 2010. — 487 с.
24. Алферов П. Роль бизнес-заказчика в ИТ-проекте. Управление проектами. №4. 2008. URL: http://www.pmmagazine.ru/document.asp?ob_no=771 (дата звернення 15.11.2021 р.).
25. Фатрелл Р., Шафер Д., Шафер Л. Управление программными проектами: достижение оптимального качества при минимуме затрат. М.: Издательский дом «Вильямс». 2013. 1136 с.
26. Соммервилл И. Инженерия программного обеспечения. Издательский дом “Вильямс”. 2015. 624 с.

27. Кантор М. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения. М.: Вильямс. 2012. 176 с.
28. Симонов С. Технологии и инструментарий для управления рисками. Jet Info. № 2. 2003. 34 с.
29. Fenton N., Neil M. Decision Support Software for Probabilistic Risk Assessment Using Bayesian Networks. IEEE Software. 2014. № 31 (2). P. 21-26.
30. Britkin A.I. Model estimate the duration of the iterative software development process. Open education. 2009. №4. P. 75.
31. Желібо Є., Заверуха Н., Зацарний В. Безпека життєдіяльності. К.: 2001. 483 с.
32. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Київ. 2018. URL: http://sop.zp.ua/norm_праор_0_00-7_15-18_01_ua.php (дата звернення: 10.12.2019).
33. Бедрій Я. Основи охорони праці користувачів персональних комп'ютерів: навчальний посібник для студентів ВНЗ та інженерів-практиків. Навчальна книга-Богдан. 2014. 144 с.

ДОДАТОК А
Апробація результатів роботи

ДОДАТОК Б

Фрагменти програмного коду програмного засобу аналізу якості даних

Б.1 – Вміст файлу package.json

```
{
  "name": "Quolitizer",
  "version": "1.0.0",
  "license": "MIT",
  "author": "Naida Volodymyr",
  "description": "Quolitizer project built using Angular 2+,
Express, Mongoose and Node.",
  "angular-cli": {},
  "engines": {
    "node": "6.11.4",
    "npm": "3.10.10"
  },
  "scripts": {
    "ng": "ng",
    "build": "ng build",
    "start": "node dist/server/app.js",
    "predev": "tsc -p server",
    "dev": "concurrently \"mongod\" \"ng serve -pc proxy.conf.json -
-open\" \"tsc -w -p server\" \"nodemon dist/server/app.js\"",
    "prod": "concurrently \"mongod\" \"ng build -aot -prod && tsc -p
server && node dist/server/app.js\"",
    "test": "ng test",
    "testbe": "tsc -p server && mocha dist/server/test --exit",
    "lint": "ng lint",
    "lintbe": "tslint server/**/*.ts{,x}",
    "e2e": "ng e2e"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^5.0.3",
    "@angular/common": "^5.0.3",
```

```

"@angular/compiler": "^5.0.3",
"@angular/core": "^5.0.3",
"@angular/forms": "^5.0.3",
"@angular/http": "^5.0.3",
"@angular/platform-browser": "^5.0.3",
"@angular/platform-browser-dynamic": "^5.0.3",
"@angular/router": "^5.0.3",
"angular2-jwt": "^0.2.3",
"bcryptjs": "^2.4.3",
"body-parser": "^1.18.2",
"bootstrap": "4.0.0-alpha.5",
"core-js": "^2.5.1",
"dotenv": "^4.0.0",
"express": "^4.16.2",
"font-awesome": "^4.7.0",
"jquery": "^3.2.1",
"jsonwebtoken": "^8.1.0",
"mongoose": "^4.13.5",
"morgan": "^1.9.0",
"rxjs": "^5.5.2",
"tether": "1.4.0",
"zone.js": "^0.8.18"
},
"devDependencies": {
  "@angular/cli": "1.5.4",
  "@angular/compiler-cli": "^5.0.3",
  "@angular/language-service": "^5.0.3",
  "@types/express": "^4.0.37",
  "@types/jasmine": "~2.5.54",
  "@types/jasminewd2": "~2.0.3",
  "@types/node": "~6.0.92",
  "chai": "^4.1.2",
  "chai-http": "^3.0.0",
  "codelyzer": "~3.2.2",
  "concurrently": "^3.5.1",
  "jasmine-core": "~2.6.2",

```

```

    "jasmine-spec-reporter": "~4.1.0",
    "karma": "~1.7.1",
    "karma-chrome-launcher": "~2.1.1",
    "karma-cli": "~1.0.1",
    "karma-coverage-istanbul-reporter": "^1.2.1",
    "karma-jasmine": "~1.1.0",
    "karma-jasmine-html-reporter": "^0.2.2",
    "mocha": "^4.0.1",
    "nodemon": "^1.12.1",
    "protractor": "~5.1.2",
    "ts-node": "~3.2.0",
    "tslint": "~5.7.0",
    "typescript": "~2.4.2"
  }
}

```

Б.2 – Код сервера

```

import * as bodyParser from 'body-parser';
import * as dotenv from 'dotenv';
import * as express from 'express';
import * as morgan from 'morgan';
import * as mongoose from 'mongoose';
import * as path from 'path';

import setRoutes from './routes';

const app = express();
dotenv.load({ path: '.env' });
app.set('port', (process.env.PORT || 3000));

app.use('/', express.static(path.join(__dirname, '../public')));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));

let mongodbURI;
if (process.env.NODE_ENV === 'test') {

```

```

    mongodbURI = process.env.MONGODB_TEST_URI;
  } else {
    mongodbURI = process.env.MONGODB_URI;
    app.use(morgan('dev'));
  }

mongoose.Promise = global.Promise;
const mongodb = mongoose.connect(mongodbURI, { useMongoClient: true
});

mongodb
  .then((db) => {
    console.log('Connected to MongoDB on', db.host + ':' + db.port);

    setRoutes(app);

    app.get('/*', function(req, res) {
      res.sendFile(path.join(__dirname, '../public/index.html'));
    });

    if (!module.parent) {
      app.listen(app.get('port'), () => {
        console.log('Angular Full Stack listening on port ' +
app.get('port'));
      });
    }

  })
  .catch((err) => {
    console.error(err);
  });

export { app };

```

Б.3 – Сервіс для керування користувачами

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs/Observable';

import { User } from '../shared/models/user.model';

@Injectable()
export class UserService {

  constructor(private http: HttpClient) { }

  register(user: User): Observable<User> {
    return this.http.post<User>('/api/user', user);
  }

  login(credentials): Observable<any> {
    return this.http.post<any>('/api/login', credentials);
  }

  getUsers(): Observable<User[]> {
    return this.http.get<User[]>('/api/users');
  }

  countUsers(): Observable<number> {
    return this.http.get<number>('/api/users/count');
  }

  addUser(user: User): Observable<User> {
    return this.http.post<User>('/api/user', user);
  }

  getUser(user: User): Observable<User> {
    return this.http.get<User>(`/api/user/${user._id}`);
  }
}
```

```

    editUser(user: User): Observable<string> {
        return this.http.put(`/api/user/${user._id}`, user, {
responseType: 'text' });
    }

    deleteUser(user: User): Observable<string> {
        return this.http.delete(`/api/user/${user._id}`, { responseType:
'text' });
    }

}

```

Б.4 – Код компоненту admin.component

```

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormControl, FormGroup, Validators } from
 '@angular/forms';

import { ToastComponent } from '../shared/toast/toast.component';
import { AuthService } from '../services/auth.service';
import { UserService } from '../services/user.service';
import { User } from '../shared/models/user.model';
import { QuolityService } from '../services/quolity.service';
import { QuolityParameter } from '../shared/models/quolity-
parameter.model';

@Component({
    selector: 'app-admin',
    templateUrl: './admin.component.html',
    styleUrls: ['./admin.component.scss']
})
export class AdminComponent implements OnInit {
    quolityParameter = new QuolityParameter();
    quolityParameters: QuolityParameter[] = [];

    users: User[] = [];

```

```

isLoading = true;
isAddParameters = false;
isEditing = false;

addParameterForm: FormGroup;
name = new FormControl('', Validators.required);
value = new FormControl('', Validators.required);
weight = new FormControl('');

constructor(public auth: AuthService,
             public toast: ToastComponent,
             private userService: UserService,
             private quolityService: QuolityService,
             private formBuilder: FormBuilder,) { }

ngOnInit() {
  this.getUsers();
  this.getParameters();
  this.addParameterForm = this.formBuilder.group({
    name: this.name,
    value: this.value,
    weight: this.weight
  });
}

getUsers() {
  this.userService.getUsers().subscribe(
    data => this.users = data,
    error => console.log(error),
    () => this.isLoading = false
  );
}

deleteUser(user: User) {
  if (window.confirm('Are you sure you want to delete ' +
    user.username + '?')) {

```

```

        this.userService.deleteUser(user).subscribe(
            data => this.toast.setMessage('user deleted successfully.',
'success'),
            error => console.log(error),
            () => this.getUsers()
        );
    }
}

toggleAddParameters() {
    this.isAddParameters = !this.isAddParameters;
}

getParameters() {
    this.quolityService.getParameters().subscribe(
        data => this.quolityParameters = data,
        error => console.log(error),
        () => this.isLoading = false
    );
}

addParameter() {

this.quolityService.addParameter(this.addParameterForm.value).subscr
ibe(
    res => {
        this.quolityParameters.push(res);
        this.addParameterForm.reset();
        this.toast.setMessage('item added successfully.',
'success');
    },
    error => console.log(error)
);
}

```

```

enableEditing(qParam: QuolityParameter) {
  this.isEditing = true;
  this.quolityParameter = qParam;
}

cancelEditing() {
  this.isEditing = false;
  this.quolityParameter = new QuolityParameter();
  this.toast.setMessage('item editing cancelled.', 'warning');
  // reload the quolity to reset the editing
  this.getParameters();
}

editParameter(qParam: QuolityParameter) {
  this.quolityService.editParameter(qParam).subscribe(
    () => {
      this.isEditing = false;
      this.quolityParameter = qParam;
      this.toast.setMessage('item edited successfully.',
'success');
    },
    error => console.log(error)
  );
}

deleteParameter(qParam: QuolityParameter) {
  if (window.confirm('Are you sure you want to permanently delete
this item?')) {
    this.quolityService.deleteParameter(qParam).subscribe(
      () => {
        const pos = this.quolityParameters.map(elem =>
elem._id).indexOf(qParam._id);
        this.quolityParameters.splice(pos, 1);
        this.toast.setMessage('item deleted successfully.',
'success');
      },

```

```

        error => console.log(error)
    );
}
}
}

```

Б.5 – Код конфігураційного файлу app.module.ts

```

import { NgModule, CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { RouterModule } from './routing.module';
import { SharedModule } from './shared/shared.module';
import { QuolityService } from './services/quolity.service';
import { UserService } from './services/user.service';
import { AuthService } from './services/auth.service';
import { AuthGuardLogin } from './services/auth-guard-
login.service';
import { AuthGuardAdmin } from './services/auth-guard-
admin.service';
import { AppComponent } from './app.component';
import { QuolityComponent } from './quolity/quolity.component';
import { AboutComponent } from './about/about.component';
import { RegisterComponent } from './register/register.component';
import { LoginComponent } from './login/login.component';
import { LogoutComponent } from './logout/logout.component';
import { AccountComponent } from './account/account.component';
import { AdminComponent } from './admin/admin.component';
import { NotFoundComponent } from './not-found/not-found.component';

@NgModule({
  declarations: [
    AppComponent,
    QuolityComponent,
    AboutComponent,
    RegisterComponent,
    LoginComponent,
    LogoutComponent,
    AccountComponent,

```

```
        AdminComponent,  
        NotFoundComponent  
    ],  
    imports: [  
        RouterModule,  
        SharedModule  
    ],  
    providers: [  
        AuthService,  
        AuthGuardLogin,  
        AuthGuardAdmin,  
        QuolityService,  
        UserService  
    ],  
    schemas: [CUSTOM_ELEMENTS_SCHEMA],  
    bootstrap: [AppComponent]  
}))  
  
export class AppModule { }
```

ДОДАТОК В

Слайди презентації

**Методи і програмний засіб аналізу якості
даних при функціонуванні програмного
забезпечення з базами даних і знань на
основі стеку технологій HTML/CSS, PHP та
MySQL**

Дипломна робота магістра

Керівник дипломної роботи:

доктор технічних наук,
професор кафедри програмної інженерії
Тернопільського національного технічного
університету імені Івана Пулюя
Пастух Олег Анатолійович

Магістрантка:

Деміда Христина Миколаївна

АКТУАЛЬНІСТЬ ТЕМИ

2

Інформаційний контекст сучасного бізнесу стрімко розвивається в умовах інтенсивної глобальної конкуренції. Прийняття рішень керівниками в даний час значною мірою залежить від своєчасної, доступної та актуальної інформації у контексті вирішення поставлених проблем

Постійний розвиток інформаційно-комунікаційних технологій породив нові системи та платформи, які щосекунди генерують велику кількість даних. Дані потенційно можуть бути носієм важливої інформації, необхідної для створення подальших знань, пов'язаних з конкретною проблемою.

Великі обсяги неструктурованих даних мають характеристики, відмінні від даних, згенерованих власними системами підтримки бізнес-процесів та прийняття рішень. Реляційні структури баз даних (наприклад, сховищ даних) та відповідні інструменти не можуть ефективно опрацьовувати та аналізувати значні обсяги інформації з точки зору розумного використання часу та ресурсних витрат. Це стало основною причиною появи технології BigData.

Оскільки, на сьогодні для підтримки бізнес-процесів 80% ринку займають інформаційні системи з реляційними базами даних, а становлення сфери BigData та NoSQL баз даних все ще інтенсивно розвивається, то актуальною задачею в індустрії програмування є забезпечення якості даних, що є складним комплексним показником і вимагає розробки та впровадження методів і засобів для оцінювання їх якості.

МЕТА, ОБ'ЄКТ І ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** полягає у дослідження методів та інструментальних програмних засобів аналізу та оцінювання якості даних при функціонуванні програмного забезпечення з базами даних і знань
- **Об'єктом дослідження** є процеси аналізу та оцінювання якості даних при функціонування систем з базами даних і знань
- **Предметом дослідження** є технології аналізу якості даних у базах даних і знань.



ЗАДАЧІ ДОСЛІДЖЕННЯ

Для досягнення мети у дипломній роботі магістра були поставлені наступні задачі:

- аналіз наукових публікацій, стандартів і технологій проектування програмних систем з базами даних і знань;
- обґрунтування та математична формалізація моделі аналізу якості даних;
- розробка методу аналізу даних щодо відповідності критеріям якості даних;
- проектування структури системи автоматизації процесу аналізу даних у системах з базами даних і базами знань;
- програмна реалізація прототипу системи аналізу та оцінювання даних.

НАУКОВА НОВИЗНА ОДЕРЖАНИХ РЕЗУЛЬТАТІВ

5

Наукова новизна одержаних результатів полягає в наступному:

- уперше запропоновано математичне представлення структури моделі ISO 25012 у вигляді комплексу показників якості даних, що включає в себе три ієрархічні блоки і дає змогу оперувати якістю даних на різних рівнях, а також спрощує реалізацію процесу автоматизації аналізу даних при функціонуванні систем з базами даних і знань.
- уперше створено метод аналізу та оцінювання якості даних при функціонуванні програмних систем з базами даних і знань, що забезпечує можливість представлення якості даних у кількісному вигляді та оперувати такими показниками як точність, ефективність, актуальність даних тощо.

КЛАСИФІКАЦІЯ МОДЕЛЕЙ ДАНИХ

Під час дослідження та аналізу різних типів даних, а також методів їх представлення у комп'ютерних системах, визначено та проаналізовано основні моделі їх представлення, які показано на даному слайді.



ПОРІВНЯННЯ ОПЕРАЦІЙ У РЕЛЯЦІЙНИХ ТА NO SQL БАЗАХ ДАНИХ

На сучасному етапі розвитку інженерії програмного забезпечення на практиці використовується дав тип баз даних: реляційні, в основі яких лежить використання табличного представлення даних і теорія реляційної алгебри, та NoSQL, що залежить від конкретної мови програмування за допомогою якої реалізовано СКБД.

База даних	Операція читання	Операція запису	Операція видалення
My SQL	Insert into journal values ('scope',234, both open and close)	Update set of Journal Id=345 where 357	Delete from list of journal where Name='pharmacy'
MongoDB	Db.journal.insert ({'name':'scope', id:235,type:'both open and closed'})	Db.journal.update({'_id':'id'})	Db.Journal.delete()
Cassandra	Insert into journal values('scope', '234','both open and close');	Update Set of journal Id=234 where id=350;	Delete set of journal id=234 where id=350
Hbase	Assign 'journal','row2', 'id:a','scope',	Similar to insert operation	Disable 'set of journal'

ХАРАКТЕРИСТИКИ СУЧАСНИХ NOSQL БАЗ ДАНИХ

Особливості і порівняння NoSQL баз даних, а також їх властивості наведено на даному слайді.

Властивості БД	MongoDB	Cassandra	Accumulo	Couch DB	Hbase	Redis	Riak
Мова програмування	C++	Java	Java	Erlang	Java	C,C++	Erlang
Модель даних	BSON	Big Table	Big Table	JSON	Big Table та Dynamo	Структура даних	Структура даних
Відмовостійкість	Реплікація	Розподіленість і реплікація	Реплікація	Реплікація	Розподіленість і реплікація	Реплікація	Реплікація
Носій даних	Пам'ять, файлова платформа	Dynamo для зберігання даних	HDFS	Пам'ять, файлова платформа	HDFS	Файлова система	Пам'ять
Ком'юніті	AGPL	Facebook	Apache	Apache	Apache	BSD	Apache
MapReduce	+	+	+	+	+	-	+
Мова запитів	-	API виклики	Java API, Thrift API	-	XML, Thrift API	API виклики	Javascript
Моделі реплікації	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація	Master-Slave реплікація
Протокол	TCP/IP	Thrift	Thrift	HTTP/ REST	Thrift, API	Бінарний протокол, подібний до telnet	REST

ПОРІВНЯННЯ ОПЕРАЦІЙ У РЕЛЯЦІЙНИХ ТА NO SQL БАЗАХ ДАНИХ

9

Переваги і недоліки використання NoSQL баз даних показано на даному слайді. Виходячи з результатів їх аналізу, можна ще раз підтвердити актуальність досліджень щодо забезпечення якості даних.

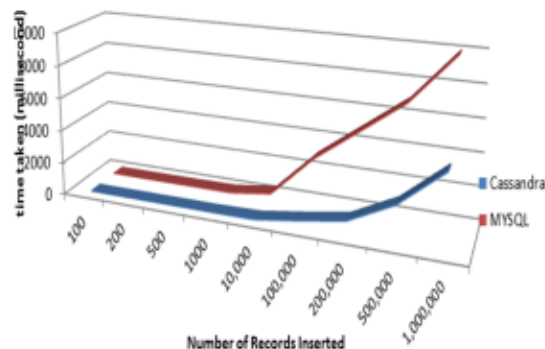
Переваги	Недоліки
Підтримка і простота масштабованості рішень	Не зрілість
Не потребують адміністратора	Не достатня швидкість і гнучкість при навантаженнях
Працює з більшим простором даних	Складність при модернізації
Величезний вибір моделей даних	Відсутня стандартизована мова запитів
NoSQL, DBaaS, Riak, Cassandra запрограмовані для роботи при збоях апаратного забезпечення.	Лише кілька NoSQL баз даних взагалі не мають негативних відгуків

ПОРІВНЯННЯ ПРОДУКТИВНОСТІ РЕЛЯЦІЙНИХ ТА NO SQL БАЗ ДАНИХ

10

Результати порівняння продуктивності реляційних та нереляційних баз даних на прикладі Cassandra та MySQL щодо запису даних показано на даному слайді. Як видно з результатів, кращі показники у NoSQL базах даних.

Кількість записів	Cassandra(MS)	MySQL
100	1	5
200	2	9
500	4	19
1000	8	43
10,000	60	400
100,000	456	3,000
200,000	918	6,000
500,000	2280	15,000
100,0000	4560	30,000

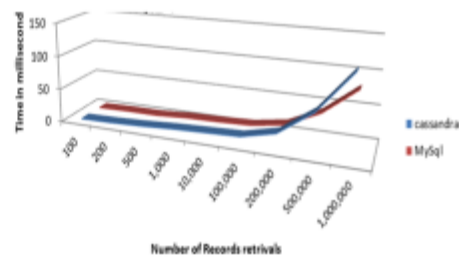


ПОРІВНЯННЯ ПРОДУКТИВНОСТІ РЕЛЯЦІЙНИХ ТА NO SQL БАЗ ДАНИХ

11

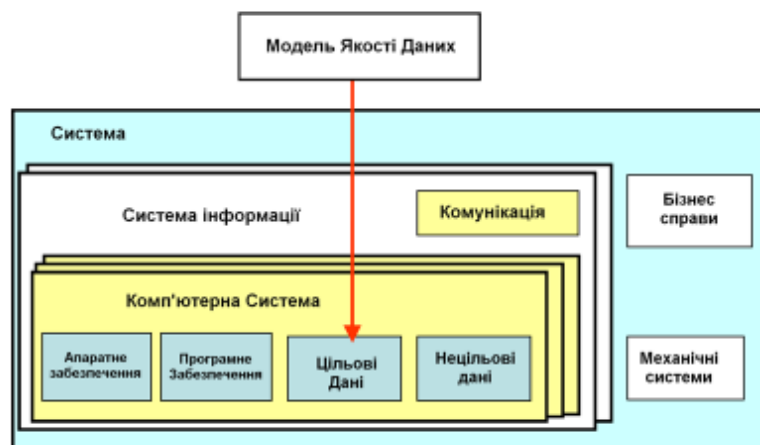
Результати порівняння продуктивності реляційних та нереляційних баз даних на прикладі Cassandra та MySQL щодо читання даних показано на даному слайді. Як видно з результатів, кращі показники у MySQL.

Кількість записів	Cassandra(MS)	MySQL(MS)
100	2	1
200	3	2
500	5	2
1,000	8	5
10,000	10	6
100,000	12	8
200,000	24	16
500,000	60	40
100,0000	120	80



МІСЦЕ МОДЕЛІ ЯКОСТІ ДАНИХ В КОМП'ЮТЕРНІЙ СИСТЕМІ

Для оцінювання якості даних розроблено міжнародний стандарт ISO 25012, застосовування якого при дослідженнях комп'ютерних систем показано на даному.



МІСЦЕ МОДЕЛІ ЯКОСТІ ДАНИХ В КОМП'ЮТЕРНІЙ СИСТЕМІ

Стандарт ISO 25012 стосується процесів управління якістю програмного забезпечення, а роль моделей серії ISO 2500n показано на даному слайді



ЗАГАЛЬНА СТРУКТУРА МОДЕЛІ ЯКОСТІ ДАНИХ

Модель якості даних, що визначена у стандарті ISO/IEC 25012, описує п'ятнадцять характеристик якості відповідно з невід'ємної і залежної від системи точок зору.

Деякі характеристики є релевантними з обох точок зору. Характеристики якості даних будуть мати різну важливість і пріоритет для різних зацікавлених сторін.

Характеристики	Якість даних	
	Невід'ємна	Залежна від системи
Чіткість	+	
Повнота	+	
Послідовність	+	
Надійність	+	
Актуальність	+	
Доступність	+	+
Узгодженість	+	+
Конфіденційність	+	+
Ефективність	+	+
Точність	+	+
Відстежуваність	+	+
Зрозумілість	+	+
Готовність		+
Портативність		+
Відновлюваність		+

АТРИБУТИ МОДЕЛІ ЯКОСТІ ДАНИХ

Для програмних систем, які функціонують з використанням баз даних і знань у дипломній роботі до визначеної моделі ISO 25012 атрибутами, які показано на даному і наступному слайдах

Характеристика	Атрибут
Повнота	Розмір файлу
	Розмір в контексті слів та речень
Послідовність	Наявність довідників та словників
	Порівняння датамп
Надійність	Джерело одержання даних
	Експертна оцінка
Актуальність	Частота оновлення даних
	Частота перевірки та зміни даних
	Стабільність оновлення даних
Узгодженість	Відповідність стандартам конфіденційності
	Залежність від зовнішнього середовища
Конфіденційність	Наявність алгоритмів шифрування
	Наявність алгоритмів дешифрування
	Надійність методу шифрування
	Вразливість

АТРИБУТИ МОДЕЛІ ЯКОСТІ ДАНИХ

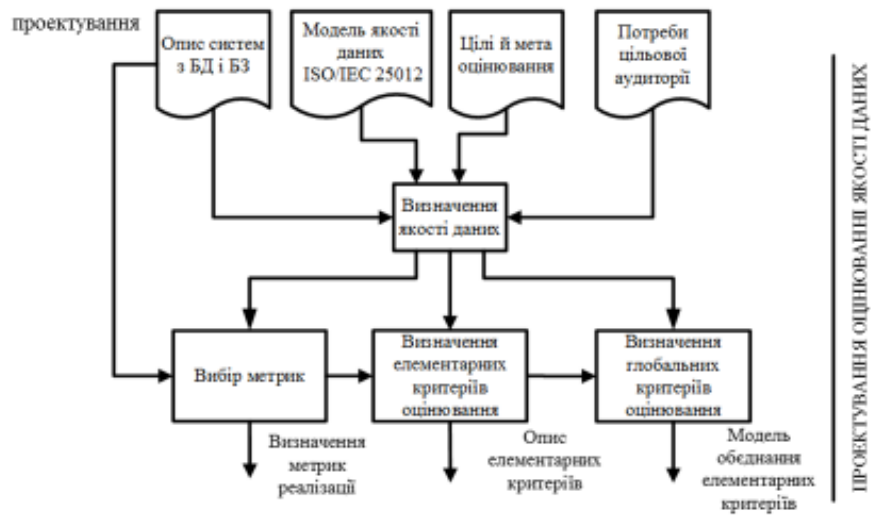
16

Характеристика	Атрибут
Ефективність	Відповідність розміру даних до виділеного для них місця у сховищі
	Відповідність типів даних значенням у файлі
	Наявність засобів підвищення пошуку
Точність	Розрядність числових типів
	Відповідність значення даних їх типу (розміру)
	Достовірність зв'язків між сутностями (атрибутами)
	Точність обчислень
Відстежуваність	Логи
Зрозумілість	Інтервал фіксації зміни даних
	Опис метаданими
Готовність	Можливість одночасного читання
	Можливість одночасного запису
	Заборона одночасного читання
	Заборона одночасного запису
	Здатність до резервування
Портативність	Можливість міграції даних
	Кількість даних, які однозначно інтерпретуються після міграції
	Кількість даних, які неоднозначно інтерпретуються після міграції
Відновлюваність	Резервування копій
	Швидка відміна внесених змін
	Синхронізація із зовнішніми засобами

МЕТОД АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

17

У роботі запропоновано метод для аналізу та оцінювання якості даних при функціонуванні систем з базами даних і знань, який передбачає виконання двох стадій: проектування оцінювання якості та безпосереднє вимірювання значень показників якості. На цьому слайді показано процес проектування



МЕТОД АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

18

На даному слайді представлено процес з відповідними етапами безпосереднього вимірювання показників якості даних



МЕТОД АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

19

Наприклад, нехай маємо атрибут, що інтерпретує кількість зв'язків ізольованих чи відсутніх даних у полях БД. В даному випадку, можна скористатися такою непрямою метрикою:

$$X = \frac{\text{Кількість пошкоджених записів}}{\text{Загальна кількість записів в БД}} \quad (2.21)$$

У результаті обрахунку одержимо конкретне кількісне значення. Для того, щоб інтерпретувати якість атрибута даних пропонується використати деяку елементарну функцію:

$$g(X) = \begin{cases} 1, & \text{якщо } X = 0; \\ \frac{X_{\max} - X}{X_{\max}}, & \text{якщо } 0 \leq X \leq X_{\max}; \\ 0, & \text{якщо } X \geq X_{\max}. \end{cases} \quad (2.22)$$

де $g(X)$ – елементарна функція для визначення оцінки;

X – метрика атрибута якості даних;

$X_{\max}$ – погоджена верхня границя, що характеризує допустимість задоволеності якості даних.

МЕТОД АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

20

Крім того, при визначенні якості атрибута можна використовувати коефіцієнт прийнятності, тоді оцінку можна обчислити за формулою

$$q(X) = g(X) \cdot A \quad (2.23)$$

де $q(X)$ – локальний показник якості;

A – коефіцієнт прийнятності, визначений для деякого атрибута в результаті експертного аналізу.

МЕТОД АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

21

Коефіцієнт прийнятності A виражає якість, доцільність, зручність досліджуваного атрибута з точки зору учасників експертного оцінювання.

Припустимо, що процедура комплексного оцінювання якості, використовує лінійну адитивну модель. Тоді визначити частинну або загальну якість даних можна, використовуючи інтегральний показник. Інтегральний показник якості обчислюємо за формулою:

$$Q = \sum_{i=1}^N q_i(X) \cdot k_i \quad (2.24)$$

де Q – інтегральна оцінка якості даних;

k_i – ваговий коефіцієнт атрибута якості даних;

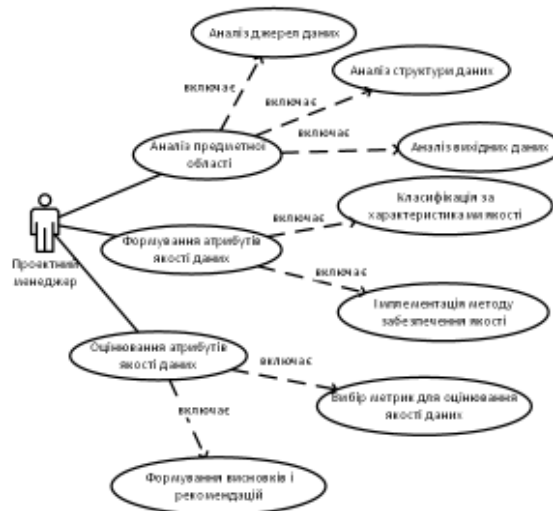
N – кількість атрибутів.

Отже, інтегральна оцінка якості даних представляє собою суму частинних показників якості окремих атрибутів якості визначених у моделі якості даних, помножених відповідно на вагові коефіцієнти атрибутів.

ЗАСІБ АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

22

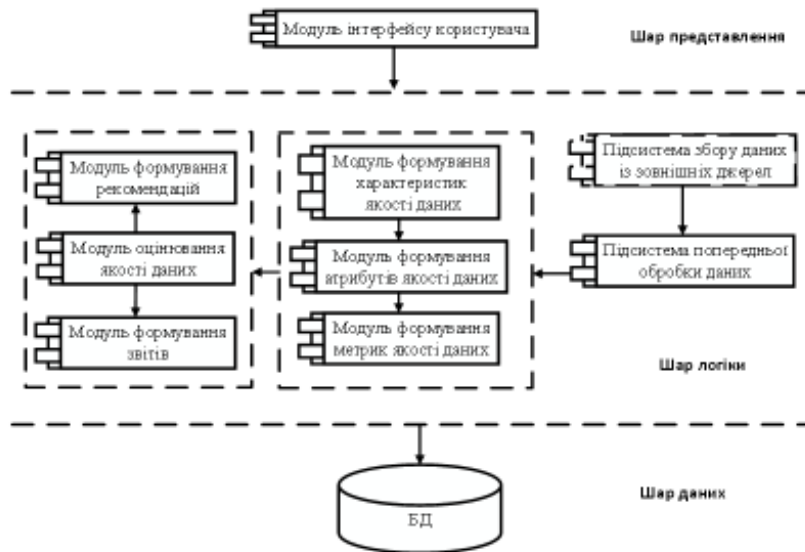
Вимоги до проєктування засобу аналізу та управління якістю даних систем при функціонування з базами даних і знань наведено на даному слайді з використанням use case діаграми



ЗАСІБ АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

23

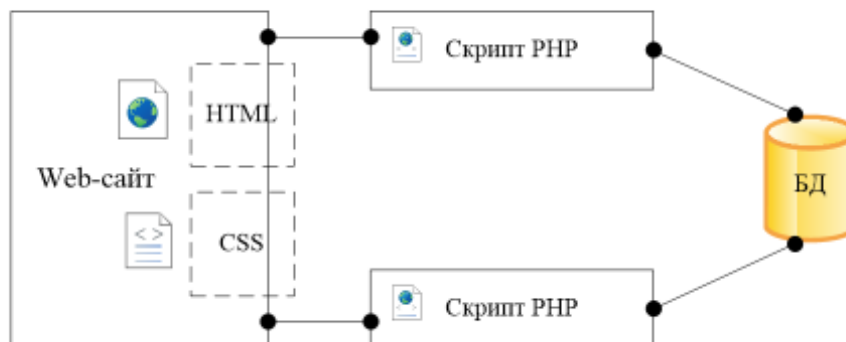
Архітектуру програмного засобу аналізу та оцінювання якості даних показано на даному слайді у вигляді шарів Фаулера.



ЗАСІБ АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

24

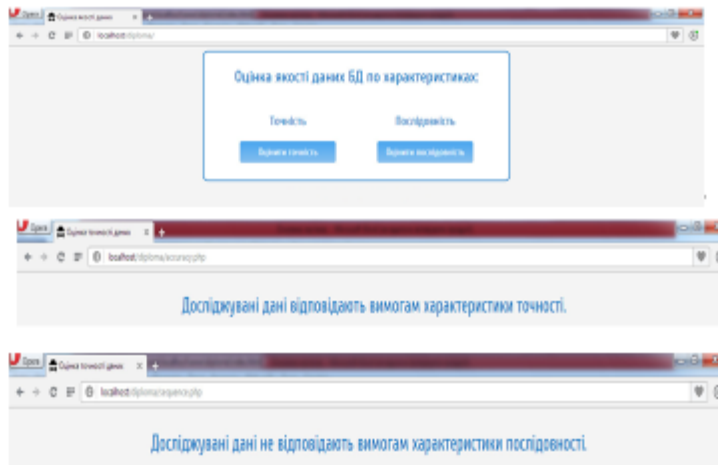
Структуру програмного засобу управління та аналізу якістю із застосуванням стек технологій HTML/CSS, мови програмування PHP та СКБД MySQL показано нижче.



ЗАСІБ АНАЛІЗУ ТА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ

25

Інтерфейс та результати оцінювання якості тестової БД показано на даному слайді і стосується двох характеристик – точності та послідовності.



ВИСНОВКИ

26

У дипломній роботі магістра одержано наступні наукові і практичні результати.

1. Проведено аналітичний огляд наукових праць і практичних порад щодо проектування та функціонування програмних систем з базами даних і знань, у результаті якого визначено роль і місце баз даних в інформаційній інфраструктурі організацій, що дало змогу обґрунтувати актуальність задач аналізу якості даних при прийнятті управлінських рішень.
2. Проаналізовано характеристики та особливості форматів представлення даних при використанні NoSQL та реляційних баз даних і знань, що дало змогу виявити фактори впливу на якість даних та в подальшому врахувати їх при побудові моделі аналізу та оцінювання якості.
3. Проведено порівняльний аналіз реляційних СКБД та NoSQL баз даних на предмет якості виконання операцій запису, зчитування та оновлення даних, а також відповідності показникам точності і продуктивності, у результаті чого встановлено, що ефективність їхнього використання залежить від кількості даних, цілей і задач, які передбачено програмною системою.

ВИСНОВКИ

27

4. Обґрунтовано необхідність використання моделі стандарту ISO 25012 для аналізу та оцінювання якості даних при функціонуванні систем з базами даних і знань, що дало змогу структурувати показники оцінювання та забезпечити повноту і ефективність результатів у визначеній прикладній предметній області.
5. Запропоновано математичне представлення структури моделі ISO 25012 у вигляді комплексу показників якості даних, що включає в себе три ієрархічні блоки і дає змогу оперувати якістю даних на різних рівнях, а також спрощує реалізацію процесу автоматизації аналізу даних при функціонуванні систем з базами даних і знань.
6. Визначено та запропоновано показники другого блоку моделі якості даних у вигляді атрибутів, що дало змогу забезпечити ефективність процесів аналізу та управління якістю даних при функціонуванні програмних систем з БД і БЗ.
7. Створено метод аналізу та оцінювання якості даних при функціонуванні програмних систем з базами даних і знань, що забезпечує можливість представлення якості даних у кількісному вигляді та оперувати такими показниками як точність, ефективність, актуальність даних тощо.

ВИСНОВКИ

28

4. Обґрунтовано необхідність використання моделі стандарту ISO 25012 для аналізу та оцінювання якості даних при функціонуванні систем з базами даних і знань, що дало змогу структурувати показники оцінювання та забезпечити повноту і ефективність результатів у визначеній прикладній предметній області.
5. Запропоновано математичне представлення структури моделі ISO 25012 у вигляді комплексу показників якості даних, що включає в себе три ієрархічні блоки і дає змогу оперувати якістю даних на різних рівнях, а також спрощує реалізацію процесу автоматизації аналізу даних при функціонуванні систем з базами даних і знань.
6. Визначено та запропоновано показники другого блоку моделі якості даних у вигляді атрибутів, що дало змогу забезпечити ефективність процесів аналізу та управління якістю даних при функціонуванні програмних систем з БД і БЗ.
7. Створено метод аналізу та оцінювання якості даних при функціонуванні програмних систем з базами даних і знань, що забезпечує можливість представлення якості даних у кількісному вигляді та оперувати такими показниками як точність, ефективність, актуальність даних тощо.

ВИСНОВКИ

29

8. Обґрунтовано застосування мови програмування PHP і стеку технологій HTML/CSS, СКБД MySQL для програмної імплементації системи аналізу якості даних і визначено їх основні переваги над іншими технологіями.
9. Спроектовано архітектуру програмної системи і схему бази даних, що враховує вимоги предметної області та особливості моделі і методу аналізу якості даних і дає змогу оцінити характеристики точності і послідовності стандарту ISO 25012.
10. Експериментально підтверджено дієвість та ефективність розробленої програмної системи аналізу якості даних при функціонуванні систем з базами даних і знань, яка враховує особливості предметної області та структуру моделі ISO 25012.

30

ДЯКУЮ ЗА УВАГУ!