

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет: комп'ютерних наук,  
менеджменту та публічного  
адміністрування  
Кафедра: інформаційних технологій

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему: Розробка інтегрованої системи для оптимізації управління  
будівельними матеріалами

Виконав магістрант групи МІС-22  
Спеціальності 122 «Комп'ютерні науки»  
САПКО Олександр Сергійович

Керівник роботи док-р PhD, доц.  
БУЧИНСЬКА Ірина Вікторівна

Рецензент д.ф.-м.н., професор  
КОВАЛЬЧУК Володимир Володимирович

## АНОТАЦІЯ

на магістерську кваліфікаційну роботу

«Розробка інтегрованої системи для оптимізації управління будівельними матеріалами»

студента Сапко Олександра Сергійовича

Інтегровані системи відіграють важливу роль на сучасних підприємствах надаючи своєчасно інформацію не тільки робітникам підприємства, а також його контрагентам. Проте нераціональне представлення даних в них та проблеми їх передачі між різними закритими системами призводять до погіршення та уповільнення бізнес-процесів. Тому актуальним є питання вивчення методів виводу даних та пошук шляхів щодо найбільш зручного їхнього представлення для користувачів.

Актуальність розробки інтегрованої системи для оптимізації управління будівельними матеріалами обумовлена такими факторами:

- зростання вартості будівельних матеріалів (у зв'язку з інфляцією та іншими факторами вартість будівельних матеріалів постійно зростає, а оптимізація управління будівельними матеріалами дозволяє компаніям заощадити кошти на закупівлі матеріалів);
- необхідність підвищення ефективності використання будівельних матеріалів (будівельним компаніям необхідно підвищувати ефективність використання будівельних матеріалів, для скорочення витрат та збереження навколишнього середовища);
- необхідність забезпечення контролю за використанням будівельних матеріалів (необхідність забезпечення контролю за використанням будівельних матеріалів).

Розробка інтегрованої системи для оптимізації управління будівельними матеріалами дозволяє вирішити ці проблеми. Система розроблена для:

- оптимізації витрат на закупівлі будівельних матеріалів;

- покращення контролю будівельних матеріалів;
- зниження ризиків.

Таким чином, розробка інтегрованої системи для оптимізації управління будівельними матеріалами є актуальною та необхідною для комунальних підприємств та будівельних компаній.

Метою роботи є розробка інтегрованої системи для оптимізації управління будівельними матеріалами яка забезпечить єдине просторове сховище даних для управління проектними задачами, даними та ресурсами.

Об'єктом дослідження є інфраструктура підприємства Skanholz та процес управління проектами на підприємстві.

Предметом дослідження є програма виводу даних для клієнтів підприємства Skanholz.

Методом дослідження є аналіз структури інформаційної системи, аналіз ринку програмного забезпечення для створення системи та застосування методу експертних оцінок та синтезу оптимальної структури системи.

Результатом роботи є розроблена інтегрована система для оптимізації управління будівельними матеріалами.

Робота складається зі вступу, 4-х розділів, висновків та переліку посилань.

*Ключові слова:* Інтегровані системи, Бази даних, СУБД, SQL, дані, модель даних, вилучення даних.

## SUMMARY

for a master's thesis « Development of an integrated system to optimize the management of building materials »

by student Oleksandr Serhiyovych Sapko

Integrated systems play an important role in modern enterprises, providing timely information not only to the workers of the enterprise, but also to its counterparties. However, the irrational presentation of data in them and the problems of their transfer between different closed systems lead to the deterioration and slowdown of business processes. Therefore, the issue of studying methods of data output and finding ways to present them most conveniently for users is relevant.

The urgency of developing an integrated system to optimize the management of building materials is due to the following factors:

- increase in the cost of construction materials (due to inflation and other factors, the cost of construction materials is constantly increasing, and optimization of construction materials management allows companies to save money on the purchase of materials);
- the need to increase the efficiency of the use of construction materials (construction companies need to increase the efficiency of the use of construction materials to reduce costs and preserve the environment);
- the need to ensure control over the use of construction materials (the need to ensure control over the use of construction materials).

The development of an integrated system for optimizing the management of building materials allows solving these problems. The system is designed for:

- optimization of costs for the purchase of building materials;
- improvement of control of construction materials;
- risk reduction.

Thus, the development of an integrated system to optimize the management of building materials is relevant and necessary for utility companies and

construction companies. The purpose of the work is to develop an integrated system for optimizing the management of building materials, which will provide a single spatial data repository for managing project tasks, data and resources.

The object of the study is the infrastructure of the Skanholz enterprise and the project management process at the enterprise. The subject of the study is a data output program for customers of the Skanholz enterprise.

The research method is the analysis of the structure of the information system, the analysis of the software market for the creation of the system and the application of the method of expert evaluations and synthesis of the optimal structure of the system.

The result of the work is a developed integrated system for optimizing the management of construction materials.

The work consists of an introduction, 4 chapters, conclusions and a list of references.

Keywords: Integrated systems, Databases, DBMS, SQL, data, data model, data extraction.

## ЗМІСТ

ВСТУП .....	9
1 ЗАСТОСУВАННЯ БАЗ ДАНИХ ДЛЯ ВИРОБНИЦТВА .....	11
1.1 Суть і поняття бази даних .....	11
1.2 Структури бази даних які використовують для керуванням виробництвом .....	17
1.3. Використання бази даних для управління виробництвом .....	23
2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ УПРАЛІННЯ БАЗАМИ ДАНИХ .....	28
2.1 Розвиток методів управління базами даних .....	28
2.2 Використання SQL-орієнтованих баз даних .....	34
2.3 Вилучення інформації з баз даних .....	42
3. ДОСЛІДЖЕННЯ ГРАФОВИХ МОДЕЛЕЙ ДЛЯ ОПИСУ І ОБРОБКИ ЧАСТКОВО СТРУКТУРОВАНОГО КОНТЕНТУ .....	50
3.1 Опис підприємства .....	50
3.2 Створення структури інформаційної системи підприємства .....	55
4 ЗАСТОСУВАННЯ БАЗ ДАНИХ ДЛЯ ВИРОБНИЦТВА .....	64
4.1 Розробка програми згідно з вимогами клієнта .....	64
4.2 Детальний опис програми .....	67
ВИСНОВКИ .....	74
ПЕРЕЛІК ПОСИЛАНЬ .....	76

## ВСТУП

У сучасну епоху цифрових технологій дані необхідні всюди, а для компаній вкрай важливо отримувати конкретну інформацію з різних джерел, щоб приймати обґрунтовані рішення. Для цього компанії застосовують бази даних.

Інтегровані системи використовуються для зберігання та керування великими обсягами структурованих і неструктурованих даних, і вони можуть використовуватися для підтримки широкого спектру дій, включаючи зберігання даних, аналіз даних і керування даними. Вони використовуються в різних середовищах, зокрема в бізнесі, наукових та державних організаціях.

Інтегровані системи даних відіграють життєво важливу роль у бізнес-операціях, оскільки дозволяє компаніям збирати інформацію, яка може інформувати процеси прийняття рішень. Це особливо важливо під час роботи з великими наборами даних, де пошук відповідної інформації вручну буде трудомістким і неефективним. Крім того, вилучення даних дозволяє організаціям отримати конкурентну перевагу, надаючи їм корисну інформацію про ринкові тенденції та поведінку споживачів. Аналізуючи ці дані, компанії можуть визначити сфери, де вони потребують покращення, і відповідно розробити стратегії.

Одним із прикладів підприємств, що активно використовує інтегровані системи для обміну даними з контрагентами є підприємство Skanholz. Тому для них актуальним є питання покращення доступності та простоти виводу даних.

У роботі пропонується розробка інтегрованої системи для оптимізації управління будівельними матеріалами. Система базується на використанні сучасних інформаційних технологій та має основні функції: ведення обліку будівельних матеріалів, планування закупівель будівельних матеріалів, контроль використання будівельних матеріалів.

Система дозволяє будівельним компаніям оптимізувати витрати на закупівлю будівельних матеріалів, покращити контроль за використанням будівельних матеріалів, знизити ризик втрат будівельних матеріалів.

Для оптимізації управління будівельними матеріалами можна використовувати сучасні інформаційні технології. У статті пропонується розробка інтегрованої системи для оптимізації управління будівельними матеріалами, яка базується на використанні таких сучасних інформаційних технологій, як веб-геоінформатика, доповнена реальність та машинне навчання.

Метою роботи є розробка інтегрованої системи для оптимізації управління будівельними матеріалами яка забезпечить єдине просторове сховище даних для управління проектними задачами, даними та ресурсами.

Задачі роботи:

- аналіз вимог замовника;
- аналіз аналогічних додатків та систем;
- створення моделі бази даних підприємства;
- створення структурної схеми;
- створення інформаційної системи згідно з потребами клієнтів підприємства.

Об'єктом дослідження є інфраструктура підприємства Skanholz та процес управління проектами на підприємстві.

Предметом дослідження є програма виводу даних для клієнтів підприємства Skanholz.

Методом дослідження є аналіз структури інформаційної системи, аналіз ринку програмного забезпечення для створення системи та застосування методу експертних оцінок та синтезу оптимальної структури системи.

Результатом роботи є розроблена інтегрована система для оптимізації управління будівельними матеріалами.



## 1 ЗАСТОСУВАННЯ БАЗ ДАНИХ ДЛЯ ВИРОБНИЦТВА

### 1.1 Суть і поняття бази даних

База даних – це інформація, налаштована для легкого доступу, керування та оновлення. Комп'ютерні бази даних зазвичай зберігають сукупність записів даних або файлів, які містять інформацію, таку як операції з продажу, дані про клієнтів, фінанси та інформацію про продукт. Бази даних використовуються для зберігання, підтримки та доступу до будь-яких даних. Вони збирають інформацію про людей, місця або речі. Ця інформація збирається в одному місці, щоб її можна було спостерігати та аналізувати. Бази даних можна розглядати як організований збір інформації.

Компанії використовують дані, що зберігаються в базах даних, для прийняття обґрунтованих бізнес-рішень. Деякі зі способів використання баз даних організаціями включають наступне:

- удосконалення бізнес-процесів. Компанії збирають дані про бізнес-процеси, такі як продажі, обробка замовлень і обслуговування клієнтів. Вони аналізують ці дані, щоб покращити ці процеси, розширити свій бізнес і збільшити дохід;
- слідкування за клієнтами. Бази даних часто зберігають інформацію про людей, наприклад клієнтів або користувачів. Наприклад, платформи соціальних мереж використовують бази даних для зберігання інформації про користувачів, такої як імена, адреси електронної пошти та поведінка користувачів. Дані використовуються, щоб рекомендувати вміст користувачам і покращувати взаємодію з ними;
- захист особистої інформації про здоров'я. Постачальники медичних послуг використовують бази даних для безпечного зберігання

особистих даних про здоров'я для інформування та покращення догляду за пацієнтами;

- зберігання персональних даних. Бази даних також можна використовувати для зберігання особистої інформації. Наприклад, особисте хмарне сховище доступне для окремих користувачів для зберігання мультимедійних даних, наприклад фотографій, у керованій хмарі.

Вперше бази даних були створені в 1960-х роках. Ці перші бази даних були мережевими моделями, де кожен запис пов'язаний з багатьма первинними та вторинними записами. Серед ранніх моделей також були ієрархічні бази даних. Вони мають схеми дерева з кореневим каталогом записів, пов'язаним із кількома підкаталогами. Реляційні бази даних були розроблені в 1970-х роках. Наступними у 1980-х роках з'явилися об'єктно-орієнтовані бази даних. Сьогодні ми використовуємо мову структурованих запитів (SQL), NoSQL і хмарні бази даних. Е. Ф. Кодд створив реляційну базу даних під час роботи в IBM. Він став стандартом для систем баз даних через його логічну схему або спосіб її організації. Використання логічної схеми відокремлює реляційну базу даних від фізичного сховища. Реляційна база даних у поєднанні з розвитком Інтернету в середині 1990-х років призвела до поширення баз даних. Багато ділових і споживчих програм покладаються на бази даних.

Існує багато типів баз даних. В обчислювальній техніці бази даних часто класифікують на основі організаційного підходу, який вони використовують. Деякі з основних організаційних баз даних включають наступне:

Реляційний. Цей табличний підхід визначає дані, щоб їх можна було реорганізувати та отримати до них доступ різними способами. Реляційні бази даних складаються з таблиць. Дані розміщуються в попередньо визначених категоріях у цих таблицях. Кожна таблиця має стовпці з принаймні однією категорією даних і рядки з певним екземпляром даних для категорій,

визначених у стовпцях. Інформація в реляційній базі даних про конкретного клієнта організована в рядки, стовпці та таблиці. Вони індексуються, щоб полегшити пошук за допомогою запитів SQL або NoSQL.

Реляційні бази даних використовують SQL у своїх інтерфейсах користувача та прикладних програм. Нову категорію даних можна легко додати до реляційної бази даних без необхідності змінювати існуючі програми. Система управління реляційною базою даних (RDBMS) використовується для зберігання, керування, запитів і отримання даних у реляційній базі даних. Як правило, RDBMS надає користувачам можливість контролювати доступ для читання/запису, вказувати створення звіту та аналізувати використання. Деякі бази даних пропонують атомарність, послідовність, ізоляцію та довговічність або сумісність з ACID, щоб гарантувати послідовність даних і завершеність транзакцій.

Дистрибутивні (Distributed). Ця база даних зберігає записи або файли в кількох фізичних розташуваннях. Обробка даних також розподіляється та тиражується в різних частинах мережі. Розподілені бази даних можуть бути однорідними, де всі фізичні розташування мають однакове базове обладнання та запускають однакові операційні системи та програми баз даних. Вони також можуть бути неоднорідними. У таких випадках апаратне забезпечення, ОС і програми бази даних можуть відрізнятися в різних місцях.

Хмарні (Cloud). Ці бази даних створено в загальнодоступній, приватній або гібридній хмарі для віртуалізованого середовища. З користувачів стягується плата залежно від обсягу пам'яті та пропускної здатності, яку вони використовують. Вони також мають можливість масштабування за вимогою та високу доступність. Ці бази даних можуть працювати з додатками, розгорнутими як програмне забезпечення як послуга.

NoSQL. Бази даних NoSQL добре підходять для роботи з великими колекціями розподілених даних. Вони можуть вирішувати проблеми продуктивності великих даних краще, ніж реляційні бази даних. Вони також добре аналізують великі неструктуровані набори даних і дані на віртуальних

серверах у хмарі. Ці бази даних також можна назвати нереляційними базами даних [1].

Об'єктно-орієнтовані (Object-oriented). Ці бази даних містять дані, створені за допомогою об'єктно-орієнтованих мов програмування. Вони зосереджені на організації об'єктів, а не на діях і даних, а не на логіці. Наприклад, запис даних зображення буде об'єктом даних, а не буквено-цифровим значенням.

Графічні (Graph). Ці бази даних є типом бази даних NoSQL. Вони зберігають, відображають і запитують відносини, використовуючи поняття з теорії графів. Бази даних графів складаються з вузлів і ребер. Вузли є сутностями і з'єднують вузли.

Ці бази даних часто використовуються для аналізу взаємозв'язків. Графічні бази даних часто використовуються для аналізу даних про клієнтів, які взаємодіють із компанією на веб-сторінках і в соціальних мережах.

Графічні бази даних використовують SPARQL, декларативну мову програмування та протокол для аналітики. SPARQL може виконувати всю аналітику, яку може виконувати SQL, а також може використовуватися для семантичного аналізу або перевірки зв'язків. Це робить його корисним для виконання аналітики наборів даних, які містять як структуровані, так і неструктуровані дані. SPARQL дозволяє користувачам виконувати аналітику інформації, що зберігається в реляційній базі даних, також стосунки «друг-друг», PageRank і найкоротший шлях (рис.1.1).

Хоча різні типи баз даних відрізняються за схемою, структурою даних і типами даних, які їм найбільше підходять, усі вони складаються з однакових п'яти основних компонентів:


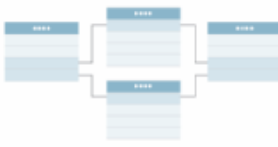
	Graph database	Relational database
FORMAT	Nodes and edges	Tables with rows and columns
RELATIONSHIPS	Considered data, represented by edges between nodes	Related across tables, established using foreign keys between tables
COMPLEX QUERIES	Run quickly and do not require joins	Require complex joins between tables
TOP USE CASES	Relationship-heavy use cases, including fraud detection and recommendation engines	Transaction-focused use cases, including online transactions and accounting
EXAMPLE		

Рисунок 1.1 - Порівняння графічних і реляційних баз даних

- Обладнання. Це фізичний пристрій, на якому працює програмне забезпечення бази даних. Апаратне забезпечення бази даних включає комп'ютери, сервери та жорсткі диски;
- Програмне забезпечення. Програмне забезпечення або додаток для роботи з базою даних надає користувачам можливість керувати базою даних. Програмне забезпечення системи керування базами даних (СУБД) використовується для управління та контролю баз даних;
- Дані. Це необроблена інформація, яку зберігає база даних. Адміністратори баз даних упорядковують дані, щоб зробити їх більш значущими;
- Мова доступу до даних. Це мова програмування, яка керує базою даних. Мова програмування і СУБД повинні працювати разом. Однією з найпоширеніших мов баз даних є SQL;
- Процедури. Ці правила визначають, як працює база даних і як вона обробляє дані.

Які можуть виникнути проблеми при використанні бази даних? Налаштування, експлуатація та підтримка бази даних пов'язані з деякими загальними проблемами, наприклад:

- потрібна безпека даних, оскільки дані є цінним бізнес-активом. Для захисту сховищ даних потрібен кваліфікований персонал із кібербезпеки, що може коштувати дорого;
- цілісність даних забезпечує достовірність даних. Не завжди легко досягти цілісності даних, оскільки це означає обмеження доступу до баз даних лише тим, хто кваліфікований для роботи з ними;
- продуктивність бази даних потребує регулярного оновлення та обслуговування бази даних. Без належної підтримки функціональність бази даних може погіршитися, оскільки змінюється технологія підтримки бази даних або дані, які вона містить;
- інтеграція бази даних також може бути складною. Це може передбачати інтеграцію джерел даних із різних типів баз даних і структур в єдину базу даних або в озера даних і сховища даних (рис.1.2).

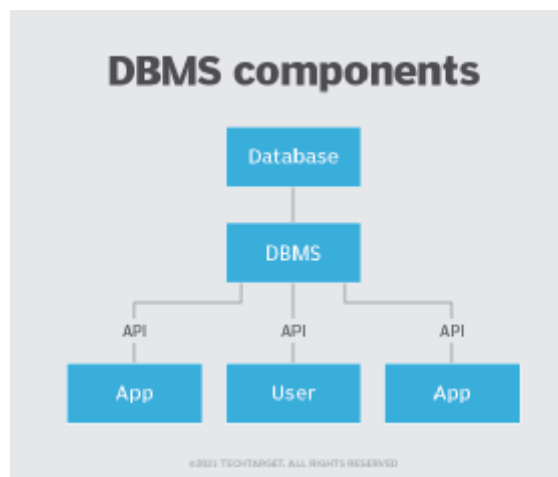


Рисунок 1.2 – Компоненти баз даних

API з'єднує користувача або програму з системою керування базою даних, що дозволяє їм взаємодіяти з базою даних [2].

СУБД дозволяє користувачам створювати базу даних і керувати нею. Він також допомагає користувачам створювати, читати, оновлювати та видаляти дані в базі даних, а також допомагає з функціями реєстрації та аудиту. СУБД забезпечує фізичну та логічну незалежність від даних. Користувачам і програмам не потрібно знати ні фізичне, ні логічне розташування даних. СУБД також може обмежувати та контролювати доступ до бази даних і надавати різні перегляди однієї схеми бази даних для кількох користувачів.

## 1.2 Структури бази даних які використовують для керування виробництвом

В даному підрозділі представлені та порівняні два підходи до моделювання структур баз даних. Перший представлений приклад - це теоретична модель, у якій кожне явище, вид дії, набір інформації, відношення та інші «речі» реального світу зі сфери процесу проектування намагаються змоделювати як об'єкти. Складна мережа зв'язків між об'єктами моделі процесу проектування (відображені в класи та об'єкти) аналізується як центральна проблема такого підходу. Такі дослідження спрямовані на моделювання багаторівневих зв'язків між інженерними структурами даних і мережею зв'язків послідовності між етапами процесу проектування, включаючи моделі ітераційних процесів. Другим представленим прикладом є структура реляційної бази даних для управління даними про продукцію та проектною документацією, розроблена для конструкторського бюро заводу електричних трансформаторів. Основними структурами бази даних є:

- шаблони структури продукту для кожної з основних функціональних збірок. Шаблон - це структура, яка використовується як відправна точка для нових проектів, включаючи посилання на документацію продукту та специфікації компонентів;

- каталог внутрішніх і стандартних компонентів, що поставляються;
- активне управління проектом (розподіл та координація проектних завдань, контроль даних та робочого процесу, процедури затвердження);
- архів попередніх проектів.

Виробнича база даних містить дані, які ви використовуєте для таких виробничих завдань, як створення та оновлення функцій. Залежно від моделі даних, яку ви використовуєте, дані у виробничій базі даних можна використовувати для створення цифрової або друкованої копії карти чи діаграми або певного типу даних [3].

Бібліотека продуктів також може містити дані для виробництва та бути виробничою базою даних. У цьому випадку база геоданих містить таблиці зі схеми бібліотеки продуктів і виробничої бази даних. Якщо виробнича база даних відокремлена від бібліотеки продуктів, дані у виробничій базі даних зазвичай відповідають моделі даних і класу продукту, що містяться в таблицях бібліотеки продуктів.

Як моделі даних, так і класи продуктів відносяться до даних або картографічного продукту, який ви створюєте за допомогою бібліотеки продуктів. Різниця полягає в тому, що модель даних включає всі набори даних і таблиці в схемі, тоді як клас продукту може містити підмножину наборів даних і таблиць у моделі даних. До класу продукту можна вибірково включати класи функцій і таблиці (рис.1.3, 1.4).

Концепція проектування бази даних - це група заходів або процедур, які покращують планування, створення, впровадження та підтримку системи управління бізнес-даними. Хороший дизайн бази даних покращує узгодженість даних, одночасно знижуючи вартість обслуговування. Це також суттєво впливає на економічні заходи щодо зберігання дисків. Тому створення бази даних вимагає інноваційної ідеї. Дотримуючись обмежень, розробник повинен вибрати, як компоненти пов'язані один з одним і який тип даних потрібно зберігати [4].



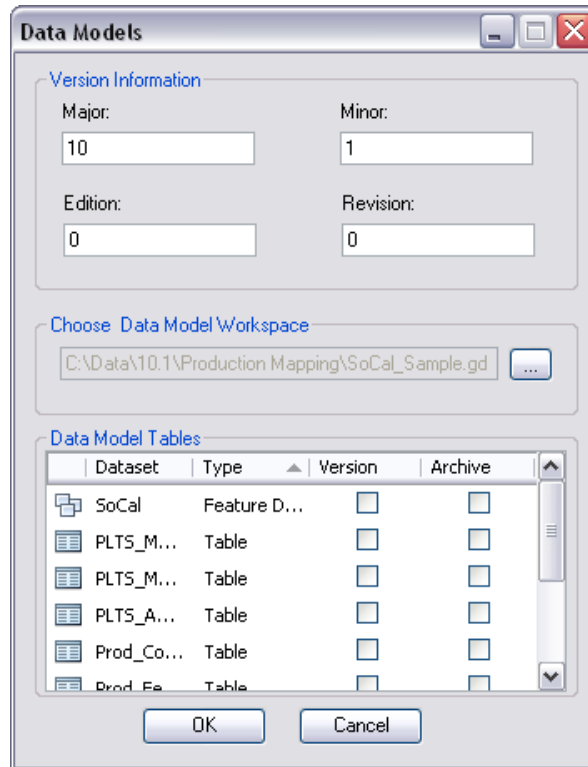


Рисунок 1.3 – Модель даних

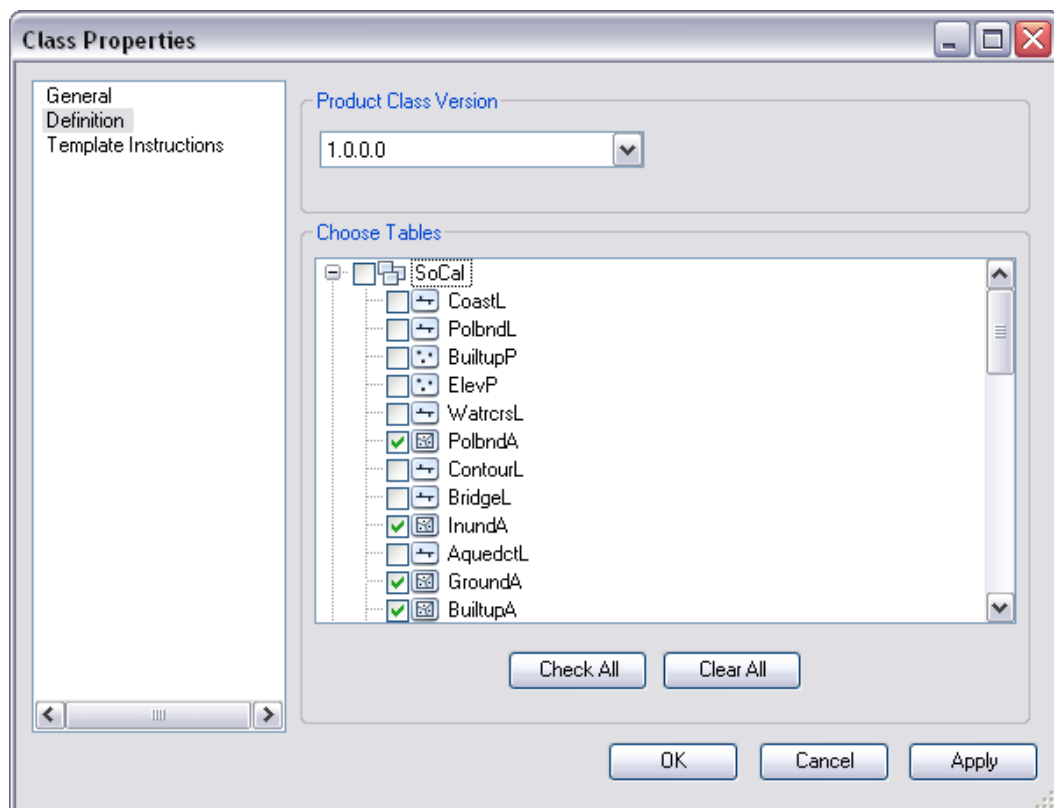


Рисунок 1.4 - Класи в базі даних

Наступним етапом є розділення потреб користувачів на різні моделі під час створення бази даних, щоб запобігти значному навантаженню або залежності від однієї функції. Як наслідок, логічні та фізичні моделі є досить важливими. Фізична модель - фізична модель стосується процедур і додатків логічної моделі.

Логічна модель - Створення моделі на основі запропонованих вимог є головною метою цього етапу. Повна модель створюється на папері без розгляду впровадження чи прийняття СУБД. База даних читає лише одну мову, і це SQL. Давайте глибше зануримося в SQL і як він приносить користь менеджерам із продуктів.

Структурована мова запитів, або SQL - це те, що має на увазі ваша технічна команда, коли вживає слово «продовження», якщо ви чули, як вони його використовують. SQL - це мова кодування, спеціально створена для баз даних. Тому він сумісний лише з базами даних. SQL використовує бази даних MySQL. Таблиці містять дані. Так само, як дані зберігаються в таблицях для ваших аркушів Excel. На основі конкретних атрибутів бази даних, відомих як ключі, ці таблиці можуть взаємодіяти одна з одною в реляційній базі даних.

Інформацію з бази даних можна створювати, отримувати, оновлювати та видаляти за допомогою SQL. Ці операції позначаються терміном CRUD. Як менеджери з продуктів, ви можете стверджувати, що ваша основна увага приділяється читабельності бази даних - отриманню даних, необхідних для ваших звітів. Інженери або менеджери баз даних визначають схему та структуру баз даних, створюючи, змінюючи та стираючи бази даних.

Переваги SQL для менеджера з продукції:

- нульова залежність від інженерів - не покладайтеся на команди даних або інженерів для створення звітів. Знайдіть для себе необхідну інформацію. Встановіть свої вимірювання та відстежуйте результати, не залучаючи інженерів. Знати це особливо корисно, якщо ви керуєте малим бізнесом або стартапом, оскільки навряд чи у вас буде спеціалізована команда даних;

- знайте, де живуть дані - ви можете краще зрозуміти свій продукт, якщо знаєте, де зберігаються дані. Ви можете переконатися, що відповідне поле включено у відповідну таблицю під час вашої наступної стратегічної зустрічі. Ви можете використовувати це для створення звітів і вимірювання ефективності вашої функції;
- швидкий доступ до даних - навіть найпростіші запити SQL дозволять отримати доступ до необхідних даних. Деякі запити можуть бути складними.

Важливість дизайну бази даних - розуміння ваших потреб клієнтів і відгуків Робота хорошого менеджера з продукту полягає в тому, щоб добре знати своїх клієнтів. Щоб визначити це, спочатку потрібно визначити профілі клієнта. Зібрати профілі разом досить складно. Але їх також потрібно осмислити. Для цього ви повинні мати знання даних. Дії ваших клієнтів впливають на прийняття ваших рішень, оскільки профілі клієнтів дають вам змогу зробити мудрий вибір на основі досвіду та вподобань користувачів. Перш ніж розробляти свій продукт, ви повинні не тільки зрозуміти свій цільовий ринок, але й зрозуміти, як вони відреагують на нього, коли він буде випущений. Ви почнете дізнаватися, де ваші користувачі відчувають біль, а де втрачають інтерес. Якщо ви знаєте дані, ви зможете точно визначити, де вони знаходяться, і вирішити, на які з них звертатися в першу чергу. Крім того, це допоможе отримати зворотний зв'язок.

Менеджери менеджерів можуть створювати чудові рішення щодо продуктів, використовуючи дані. Менеджери з продуктів виконують дві роботи, щоб визначити, чи бренд справді успішний і як модифікації впливають на продукт. Для цього потрібно поєднати відгуки та дані. Ваш продукт можна покращити, якщо ви знаєте, як оцінити дані. Постійний потік даних допоможе вам приймати рішення, засновані на фактах, а не на основі емоцій, коли ви намагаєтеся вдосконалити свій продукт. Це має вирішальне значення для розробки абсолютно нових продуктів і для менеджерів із

продуктів, які працюють у новому секторі економіки. Ви повинні мати на своєму боці якомога більше конкретних фактів, поки робите крок у невідоме.

Розуміння й обговорення даних із вашими фахівцями з обробки даних є величезною перевагою, оскільки команди часто співпрацюють між різними функціями. Це схоже на вивчення їхньої мови. Ви зможете краще сприймати їх і висловлювати їм свої думки та запитання. Великі фірми мають різні позиції, орієнтовані на дані, і якщо ви хочете просунути у своїй кар'єрі, ви повинні вміти з ними співпрацювати. Дійсно, деякі люди більше керуються даними, ніж інші. Наприклад, ви можете виявити, що ваш новий менеджер із маркетингу продуктів насолоджується даними, якщо ви підете в нову організацію [11].

Вивчення даних допоможе вам максимально використати свій час, відокремлюючи важливі від неважливих нагальних справ (рис.1.5). Тут знадобляться компроміси. Ви повинні розуміти, що зосередження на одному предметі зараз може негативно вплинути на інше пізніше. Але дані повинні досить добре розповісти вам, якою може бути ця зміна та який вплив вона матиме.

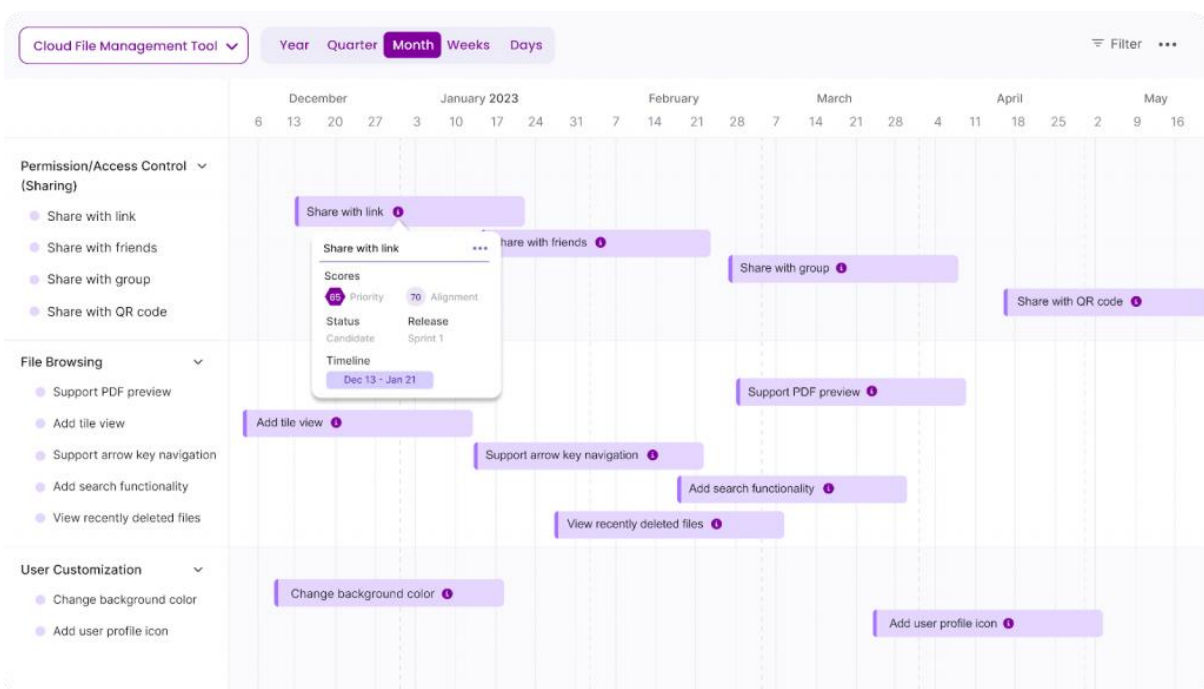


Рисунок 1.5 – Планування часу в базі даних

Дані важливі, щоб переконати членів команди та змусити їх узгодити єдину ціль продукту. Дані підтверджують ваші погляди, оскільки вони об'єктивні та не містять помилок. Це принесе користь як зацікавленим сторонам, так і членам команди. Крім того, дані дозволять вам надавати змістовні оновлення зацікавленим сторонам, орієнтованим на дані, яким потрібна додаткова інформація.

### 1.3 Використання бази даних для управління виробництвом

Підтримка цілісності, безпеки та продуктивності вашої програми залежить від ефективного керування робочою базою даних. Технології дорогі, і витрати завжди важливі. Чітке розуміння ваших інвестицій у дані може допомогти підготуватися до максимізації рентабельності інвестицій.

Озера даних знаходяться на нижній частині інвестиційної шкали. Озеро даних надає вам великий обсяг даних відносно дешево; ви отримуєте великий обсяг за нижчу початкову вартість. Якщо ви хочете фіксувати кожен частину даних, які виробляє ваша компанія, озера даних можуть підійти. Недоліком озер даних є те, що вони не призначені для швидкого збирання інформації. Відсутність структури озер даних робить отримання даних повільним і громіздким. Те, що ви заощадите на початковій ціні, може бути витрачено на час або інші системи, коли ви почнете намагатися отримати дані зі свого озера. Часто компанії використовують озера даних як зону зберігання перед тим, як перемістити дані в іншу більш структуровану систему або в групу спеціалістів із обробки даних для їх класифікації [5].

Бази даних є найдоступнішими в ціновій шкалі, і вони дуже хороші для зберігання великих обсягів даних. Витрати на гнучке зберігання зазвичай зменшуються разом із потребами. Їх проста структура (рядки та стовпці) робить пошук і використання інформації відносно швидким і легким. Кожна технологічна інфраструктура має компонент бази даних, а деякі мають більше одного. Існують додаткові приховані витрати, пов'язані з базами даних,

особливо щодо якості даних. Оскільки ваші бази даних використовуються, якість даних зазвичай погіршується. Користувачі не дотримуються протоколів введення даних, додаються налаштування та обхідні шляхи, а також створюються надмірності. Ці проблеми зазвичай виявляються під час оновлення. З часом це створює проблеми з продуктивністю та затримкою, вирішення яких вимагає великих витрат.

Сховища даних знаходяться на вищому кінці шкали витрат. Витрати визначаються обсягом даних, що швидко підвищує ціни. Однак ця вартість дає командам високо структуроване середовище даних, оскільки ви платите за сортування історичних даних під час їх зберігання. Пізніше це може призвести до економії коштів з точки зору простоти використання та доступності (хоча це все ще може мати деяку затримку) [12].

Здатність «бачити», що відбувається на машині чи виробничій лінії в даний момент, є ключовою для ефективності, якості продукції, працездатності машини та економії коштів. Оскільки це дуже специфічні системи, вони знаходяться на вершині цінової шкали інструментів керування даними виробництва. Однак, враховуючи знання та економію, яку вони приносять у виробництво та бізнес-операції, вони більше ніж заробляють на утримання та рентабельність інвестицій.

Системи керування даними призначені не лише для зберігання даних. Ви хочете отримувати дані для аналізу та тенденцій, які можуть підтримувати конкурентоспроможність вашого бізнесу. Тому важливо розуміти гнучкість і доступність кожного варіанту. Наскільки легко отримати доступ до даних і маніпулювати ними? Який рівень знань потрібен для виконання цієї роботи? Озера даних надзвичайно гнучкі, але мають обмежений доступ. Їх плоска архітектура та неструктуровані дані спрощують налаштування та переналаштування даних для моделей даних, запитів і програм. Однак, оскільки дані є неструктурованими, для їх справжнього використання потрібні технічні знання. Це означає, що дані не є легкодоступними для всіх. Натомість доступ до даних обмежується вченими та розробниками даних (або

IT-командами). Це, у свою чергу, створює вузьке місце для інших користувачів у вашій організації, включно з операторами, інженерами, керівниками заводів і C-Suite. Для порівняння, бази даних і сховища даних мають обмежену гнучкість, але високу доступність. Налагодження їх структур вимагає технічних знань і може зайняти значний час. Це повільний, делікатний процес. Однак їхні протоколи структурованих даних роблять дані доступними для більш широкого кола користувачів. У цю еру вчених, що займаються даними громадян, доступність є ключем до успіху. Працівники будь-якого рівня залежать від даних, щоб бути ефективними на робочому місці. Використання технології керування даними, яка сприяє демократизації даних, дозволяє виробникам використовувати колективні знання та досвід усього свого персоналу, а не звужувати його лише технічними командами. Це спосіб оснастити експертів процесів аналітикою для самообслуговування та штучним інтелектом, який працює на основі людських ресурсів, щоб стимулювати відкриття без необхідності писати код.

Що стосується систем керування даними, швидкість обробки різна. З огляду на те, що промисловість розвивається блискавично, розуміння змінних швидкості доступу та використання даних допомагає керувати очікуваннями. Це також може допомогти вам визначити, яка система відповідає вашим потребам і цілям. Озера даних – це великі сховища даних. Їхня здатність приймати дані будь-якого формату є зручною. Як зазначено вище, плоска архітектура та неструктурована схема даних дають змогу використовувати інформацію різноманітними корисними способами, але для цього знадобиться час. Коли виконується запит, система повинна переглядати кожен частину даних по черзі, щоб розібрати запитані дані. Скільки часу це займе, залежить від розміру озера даних. Якщо ваше озеро даних стає болотом даних, що може статися легко, час обробки збільшується. Бази даних і сховища даних обробляють дані швидше. Їхні структуровані дані та архітектура дозволяють здійснювати більш цілеспрямований пошук; система може перейти безпосередньо до відповідної області даних. Затримка

мінімальна, а результати, у більшості випадків, миттєві. Це робить їх гарним вибором для повсякденних бізнес-функцій: звітності, аналізу, візуалізації тощо. Вони можуть не відставати від попиту.

Однак застарілі системи та якість даних можуть впливати на швидкість обробки баз даних і сховищ даних. Помітна затримка може означати, що настав час для оновлення або очищення даних. Перешкодою для швидкості обробки може бути пропускна здатність вашої компанії. Навіть із обмеженням пропускної здатності затримки обробки зазвичай становлять невелику частку інших рішень для обробки даних, перелічених тут.

Здатність масштабувати систему керування даними є життєво важливим аспектом, який слід враховувати при виборі рішення. Забезпечення сумісності рішень для обробки даних у вашій інфраструктурі є ключем до успішного масштабування. Іншими словами, архітектура системи означає, що вона може обробляти лише стільки запитів одночасно. Крім того, якщо дані зберігаються на фізичних серверах, розташованих локально, важко отримати дані з віддалених місць. Команди можуть не відчувати труднощів з доступом до даних на заводі, де вони працюють, але для створення звітів або отримання даних з інших заводів знадобиться більше часу. Це ускладнює перехресне порівняння продуктивності, співпрацю над спільними проблемами або перегляд колективних даних для ініціатив або цілей усієї компанії. В сучасну епоху роботи з дому і демократизованих даних ці обмеження можуть мати серйозні наслідки для результатів бізнесу. Іншими словами, масштабувати систему, щоб охопити ціле підприємство, надзвичайно важко. Їм не вистачає можливості легко та швидко отримувати дані з кількох установок одночасно. Системи часто добре працюють на рівні заводу, але погано служать як рішення для всього підприємства - тенденція, яку ми не бачимо, щоб зникнути найближчим часом [13].

Насправді виробничі інфраструктури використовують комбінацію різних інструментів керування даними. Сховища даних зберігають історичні дані. Бази даних виконують щоденні функції. Розуміння можливостей і



недоліків кожного інструменту керування даними гарантує, що ви використовуєте їх на свою користь і максимально використовуєте їх індивідуальні сильні сторони.

Найголовніше, чітке розуміння цих рішень для керування даними допоможе вам оптимізувати ваш найбільший виробничий актив: ваш персонал. Розширення можливостей команд за допомогою даних забезпечує роботу ваших виробничих ліній і рух вашого бізнесу. Узгодження інструментів керування даними з працівником і завданням забезпечує ефективність і ефективність.

## 2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ УПРАВЛІННЯ БАЗАМИ ДАНИХ

### 2.1 Розвиток методів управління базами даних

Система керування базами даних (СУБД) - системне програмне забезпечення для створення та керування базами даних. СУБД дозволяє кінцевим користувачам створювати, захищати, читати, оновлювати та видаляти дані в базі даних. Найпоширеніший тип платформи керування даними, СУБД, по суті, служить інтерфейсом між базами даних і користувачами або прикладними програмами, гарантуючи, що дані послідовно організовані та залишаються легкодоступними.

СУБД керує даними. Механізм бази даних дозволяє отримувати доступ до даних, блокувати їх і змінювати; а схема бази даних визначає логічну структуру бази даних. Ці три основні елементи допомагають забезпечити паралельність, безпеку, цілісність даних і уніфіковані процедури адміністрування даних. СУБД підтримує багато типових завдань адміністрування баз даних, включаючи керування змінами, моніторинг і налаштування продуктивності, безпеку, резервне копіювання та відновлення. Більшість систем керування базами даних також відповідають за автоматичний відкат і перезапуск, а також за журналювання та аудит активності в базах даних і програмах, які до них звертаються. СУБД забезпечує централізоване уявлення про дані, до яких можуть мати контрольований доступ кілька користувачів із різних місць [6].

СУБД може обмежувати, які дані бачать кінцеві користувачі та як вони переглядають дані, надаючи багато переглядів єдиної схеми бази даних. Кінцеві користувачі та програмне забезпечення вільні від необхідності розуміти, де фізично розташовані дані або на якому типі носія вони знаходяться, оскільки СУБД обробляє всі запити. СУБД може запропонувати

як логічну, так і фізичну незалежність даних, щоб захистити користувачів і програми від необхідності знати, де зберігаються дані, або від занепокоєння щодо змін у фізичній структурі даних. Поки програми використовують інтерфейс прикладного програмування (API) для бази даних, яку надає СУБД, розробникам не доведеться змінювати програми лише тому, що в базу даних були внесені зміни. У системі керування реляційною базою даних (RDBMS) – найпоширенішому типі СУБД – API є SQL, стандартною мовою програмування для визначення, захисту та доступу до даних. СУБД – це складна частина системного програмного забезпечення, що складається з кількох інтегрованих компонентів, які забезпечують узгоджене кероване середовище для створення, доступу та зміни даних у базах даних

СУБД дозволяє користувачам упорядковувати, зберігати, отримувати та оптимізувати дані за допомогою програмного забезпечення з консоллю керування. Ключові компоненти системи включають оперативні дані та метадані, документацію, мови бази даних, процесор запитів, менеджер бази даних під час виконання, механізм бази даних і звітність. Існують різні типи інструментів DMBS, хоча їх можна розділити на два основних типи (рис.2.1):

- реляційний: Дані відображаються у вигляді таблиць рядків і стовпців із строгою структурою та чіткими залежностями. Оскільки мова структурованих запитів є ядром цих систем, реляційні бази даних також називають SQL. Вони корисні для складних програмних рішень, але можуть зіткнутися з проблемами, коли їх просять масштабувати [14];
- нереляційна: неструктуровані дані збираються в одному документі замість того, щоб бути організованими в рядки та стовпці, тому це вважається документоорієнтованою системою. Нереляційні бази даних також називають NoSQL. Вони є хорошим варіантом для стартапів або невеликих організацій, які використовують Agile-розробку.

Інші типи СУБД включають ієрархічні бази даних і об'єктно-орієнтовані бази даних.

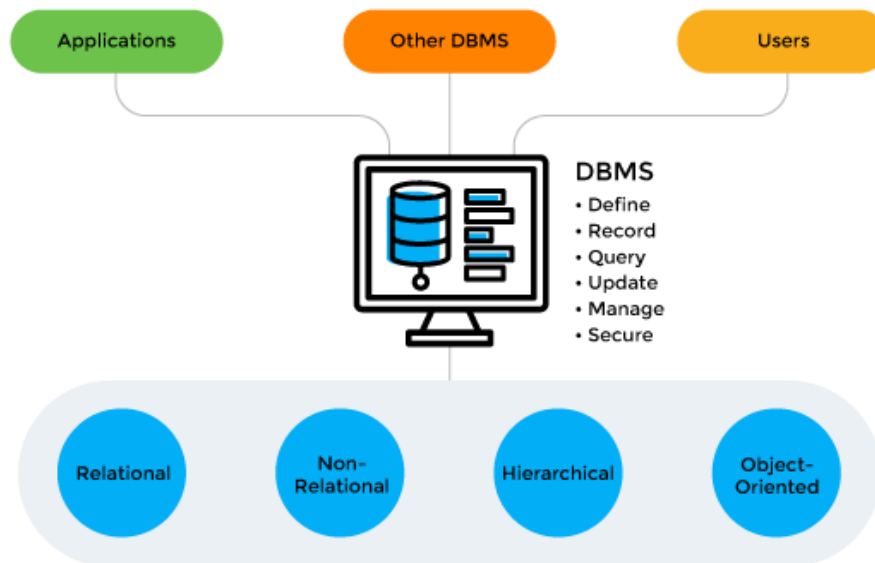


Рисунок 2.1 – Вигляд СУБД

На найпростішому рівні інструменти керування базами даних надають організаціям основу для ефективного зберігання та обробки даних. Вони спрямовані на контроль даних протягом усього життєвого циклу, забезпечуючи систематичний і безпечний спосіб створення, отримання, оновлення та керування даними. Функції інструменту керування базою даних можуть включати налаштування програми, тестування часу відгуку та тестування пропускної здатності.

Під час вибору СУБД такі елементи можуть вплинути на ваше рішення: структура даних, ліцензування, масштабованість, хмарна сумісність, безпека даних, надійність документації, крива навчання, споживання ресурсів, підтримка різних типів даних і звітність [7].

Інтеграція даних, доступ і аудит покращуються шляхом впровадження СУБД. Завдяки таким якостям СУБД, як абстракція та незалежність, організації можуть оновлювати сховище та масштабувати інфраструктуру, не впливаючи на повсякденні операції з базами даних. Наявність єдиної консолі

для виконання адміністративних завдань (замість спроб зіставляти електронні таблиці та документи з різних джерел) полегшує роботу адміністраторів баз даних.

Одна з найбільших переваг використання СУБД полягає в тому, що вона дозволяє користувачам і програмістам додатків отримувати доступ і використовувати ті самі дані одночасно, керуючи цілісністю даних. Дані краще захищені та обслуговуються, коли ними можна спільно користуватися за допомогою СУБД замість створення нових ітерацій тих самих даних, що зберігаються в нових файлах для кожної нової програми. СУБД забезпечує центральне сховище даних, до якого декілька користувачів можуть мати контрольований доступ. Централізоване зберігання та управління даними в СУБД забезпечують наступне:

- абстракція та незалежність даних;
- безпека даних;
- механізм блокування одночасного доступу;
- ефективний обробник для збалансування потреб кількох програм, які використовують ті самі дані;
- можливість швидкого відновлення після збоїв і помилок;
- сильні можливості забезпечення цілісності даних;
- протоколювання та аудит діяльності;
- простий доступ за допомогою стандартного API;
- єдині процедури адміністрування даних.

Ще одна перевага СУБД полягає в тому, що адміністратори бази даних (DBA) можуть використовувати її, щоб накласти логічну структуровану організацію даних. СУБД забезпечує економію масштабу для обробки великих обсягів даних, оскільки вона оптимізована для таких операцій [15].

Найбільшим недоліком є вартість обладнання, програмного забезпечення та персоналу, необхідних для роботи корпоративної СУБД, такої як SQL Server, Oracle або IBM Db2. Апаратне забезпечення зазвичай являє собою сервер високого класу зі значним обсягом налаштованої пам'яті в

поєднанні з великими дисковими масивами для зберігання даних. Програмне забезпечення включає саму СУБД, яка коштує дорого, а також інструменти для програмування та тестування, а також для адміністраторів баз даних для забезпечення можливості керування, налаштування та адміністрування.

З точки зору персоналу, використання СУБД вимагає найму адміністратора баз даних, навчання розробників правильному використанню СУБД і, можливо, наймання додаткових системних програмістів для керування встановленням та інтеграцією СУБД в ІТ-інфраструктуру. Робота з додатковою складністю також є проблемою при реалізації СУБД. Програмне забезпечення СУБД є складним і вимагає глибоких знань для правильного впровадження та керування. Але СУБД взаємодіє з багатьма іншими ІТ-компонентами, такими як ОС, системи обробки транзакцій, мови програмування та мережеве програмне забезпечення. Забезпечення належної конфігурації та ефективності такого складного налаштування може бути складним і спричинити уповільнення продуктивності або навіть збої системи.

Деякі витрати та адміністративні накладні витрати на роботу корпоративних систем баз даних можна зменшити за допомогою моделі хмарних обчислень. Наприклад, постачальник хмарних послуг (CSP) встановлює та керує апаратним забезпеченням, яким можуть спільно користуватися користувачі хмари. Крім того, сховище, пам'ять та інші ресурси можна масштабувати відповідно до потреб залежно від потреб використання. А базові завдання адміністратора баз даних, такі як встановлення виправлень і просте резервне копіювання, стають обов'язком CSP. Таким чином, для деяких баз даних може бути простіше та економічно ефективніше розгорнути в хмарі, а не локально.

Підприємства, яким потрібно зберігати дані та отримувати до них пізніше для ведення бізнесу, мають життєздатний варіант використання для розгортання СУБД. Будь-яка програма, яка потребує великого обсягу даних, до яких мають отримати доступ кілька користувачів або клієнтів, є кандидатом на використання СУБД. Більшість середніх і великих організацій

можуть отримати вигоду від використання СУБД, оскільки вони мають більше потреб у обміні даними та одночасному забезпеченні та можуть легше подолати проблеми з витратами та складністю.

До 2019 року технології СУБД з відкритим вихідним кодом стрімко набирали обертів. Насправді Gartner прогнозує, що бази даних з відкритим кодом становитимуть 10% загальних витрат на програмне забезпечення для баз даних у цьому році завдяки зростанню впровадження на підприємствах. До 2022 року три з п'яти найкращих баз даних за рейтингом DB-Engines були відкритими. Більшість основних ІТ-організацій використовують програмне забезпечення з відкритим кодом у деяких своїх критично важливих операціях. Ця тенденція доповнює дві інші: придбання більшими конкурентами постачальників баз даних з відкритим кодом і розширення ринку послуг хмарних баз даних. У 2019 році Gartner також заявив, що хмарні бази даних сприяли більшій частині зростання ринку СУБД, описуючи хмару як «платформу за замовчуванням для керування даними». У 2021 році Gartner дійшов висновку, що «до 2022 року дохід від хмарних систем управління базами даних становитиме 50% від загального доходу ринку СУБД». У зв'язку зі зростаючим переходом до хмари численні постачальники СУБД представили керовані хмарні служби баз даних, які пропонують звільнити ІТ-команди та групи керування даними від виконання багатьох завдань, необхідних для розгортання, налаштування та адміністрування систем баз даних [16].

Ще одна зростаюча тенденція - це те, що Gartner називає НТАР - використання однієї СУБД для обробки транзакцій і аналітики без потреби окремої СУБД для кожної операції. Щоб підтримати цю тенденцію, все більше постачальників СУБД створюють гібридні системи баз даних, які надають кілька механізмів баз даних в одній СУБД. Більшість гібридних СУБД забезпечують комбінацію реляційних і кількох механізмів і API NoSQL. Приклади включають Altibase, Microsoft Azure Cosmos DB і DataStax Enterprise. Перша СУБД була розроблена на початку 1960-х років, коли

Чарльз Бахман створив навігаційну СУБД, відому як інтегроване сховище даних. Невдовзі після цього IBM розробила систему управління інформацією (IMS), ієрархічну СУБД, призначену для мейнфреймів IBM, яка й сьогодні використовується багатьма великими організаціями.

## 2.2 Використання SQL-орієнтованих баз даних

У світі інформатики бази даних необхідні для зберігання різних типів даних і керування ними. Бази даних SQL, зокрема, відіграють вирішальну роль в управлінні складними структурами даних і їх організації таким чином, щоб зробити пошук і маніпулювання даними ефективними та надійними. SQL (Structured Query Language) - це доменно-спеціальна мова, яка використовується для керування реляційними базами даних і виконання таких завдань, як отримання, оновлення та видалення даних із них.

База даних SQL - це тип бази даних, яка використовує SQL для керування своїми даними та забезпечує відповідну платформу для широкого кола програм і користувачів. Вона містить таблиці, які зберігають дані в рядках і стовпцях, де кожен рядок представляє окремий запис, а стовпці представляють атрибути цього запису. Наприклад, у таблиці, що містить інформацію про студентів, кожен рядок представлятиме студента, а кожен стовпець представлятиме таку інформацію, як ім'я або дата народження студента.

Бази даних SQL відіграють життєво важливу роль у різних сферах інформатики. Вони забезпечують організований і структурований спосіб зберігання та керування величезними обсягами даних, забезпечуючи легкість пошуку, організації, аналізу та обробки інформації. Деякі з ключових причин важливості баз даних SQL в інформатиці включають [8]:

- ефективне зберігання та керування даними;
- послідовний і гнучкий пошук даних;
- цілісність і безпека даних;



- ефективна та надійна система резервного копіювання;
- паралельний контроль і управління транзакціями;
- масштабованість і підтримка розподілених систем.

Будучи універсальною та потужною мовою, SQL служить різноманітним цілям у управлінні та маніпулюванні даними в реляційних базах даних [17].

Деякі з ключових завдань аналізу даних і звітності, які полегшує SQL, включають:

- фільтрування та сортування даних на основі певних критеріїв: SQL дозволяє користувачам застосовувати умови до своїх запитів за допомогою пропозиції WHERE, дозволяючи їм отримувати лише ті дані, які відповідають їхнім вимогам. Крім того, речення ORDER BY можна використовувати для сортування результатів на основі певних стовпців або виразів;
- виконання обчислень і агрегування даних: SQL підтримує різні математичні та статистичні функції, такі як SUM, AVG, MIN і MAX, які можна використовувати для виконання обчислень і агрегацій даних для глибшого аналізу;
- об'єднання даних із кількох таблиць за допомогою об'єднань і підзапитів: SQL надає кілька методів об'єднання даних із різних таблиць, таких як INNER JOIN, OUTER JOIN і підзапити, допомагаючи користувачам отримувати доступ до пов'язаних даних і отримувати інформацію, яка охоплює кілька таблиць;
- групування даних для генерування підсумків і статистики. Речення GROUP BY у SQL дозволяє користувачам групувати дані на основі одного або кількох стовпців, уможливаючи генерацію підсумків і статистики для кожної групи. Це може бути особливо корисним для виявлення тенденцій або закономірностей у даних;
- створення аналізу на основі часу та тенденцій: SQL пропонує різні функції дати та часу, такі як DATE\_ADD, DATE\_SUB і

DATE\_FORMAT, які можна використовувати для аналізу даних на основі конкретних періодів, визначення сезонних моделей або прогнозування майбутніх тенденцій;

- створення спеціальних звітів для прийняття рішень: SQL дозволяє користувачам створювати спеціальні звіти на льоту, пристосовуючи результати до конкретних бізнес-потреб і процесів прийняття рішень.

SQL відіграє вирішальну роль у бізнес-аналітиці та сховищах даних, де великі обсяги даних консолідуються, перетворюються та зберігаються для подальшого аналізу. Його функції масштабованості та оптимізації дозволяють виконувати складні завдання, такі як процеси видобування, перетворення, завантаження (ETL) і аналіз даних.

Ключові завдання бізнес-аналітики та сховищ даних, які підтримує SQL, включають:

- вилучення даних із багатьох джерел: SQL можна використовувати для вилучення даних із різних джерел, таких як реляційні бази даних, плоских файлів або навіть API, що полегшує консолідацію даних у центральному сховищі;
- перетворення та очищення даних: SQL надає широкий набір функцій і операцій, які можна використовувати для перетворення та очищення даних, таких як маніпулювання рядками, форматування дати та умовні вирази;
- завантаження даних у сховище даних: можливості масового завантаження SQL дозволяють ефективно вставляти великі обсяги даних у сховище даних, мінімізуючи час і ресурси, необхідні для процесу завантаження;
- управління схемою сховища даних і метаданими: SQL можна використовувати для визначення та підтримки структури сховища даних, включаючи таблиці, подання, індекси та обмеження, забезпечуючи організованість і ефективне зберігання даних;

- оптимізація продуктивності запитів: SQL пропонує різні методи оптимізації, такі як індексування, розділення та матеріалізовані представлення, які можна використовувати для покращення продуктивності запитів до великих наборів даних у сховищі даних;
- впровадження безпеки даних і контролю доступу: компонент SQL Data Control Language (DCL) дозволяє адміністраторам баз даних керувати доступом користувачів і дозволами в сховищі даних, забезпечуючи безпеку даних і відповідність.

SQL широко використовується в розробці веб-додатків, оскільки він сприяє ефективному зберіганню та пошуку даних для створення динамічного вмісту. Веб-розробники часто інтегрують SQL із популярними мовами програмування, такими як Python, PHP і JavaScript, для створення веб-сайтів і програм, керованих даними.

Основні завдання розробки веб-додатків, які підтримує SQL, включають:

- створення та керування базовою базою даних: SQL можна використовувати для проектування та підтримки структур бази даних для веб-додатків, забезпечуючи організованість та ефективне зберігання даних;
- реалізація автентифікації та авторизації користувачів: SQL дозволяє розробникам зберігати та керувати обліковими даними та ролями користувачів, дозволяючи створювати безпечні веб-додатки з механізмами контролю доступу;
- створення динамічних веб-сторінок і вмісту: надаючи запити та маніпулюючи даними за допомогою SQL, розробники можуть створювати веб-сторінки, які відображають динамічний вміст на основі введення користувача, уподобань або інших критеріїв;
- реалізація функцій пошуку: потужні можливості SQL для створення запитів можна використовувати для створення функцій пошуку в веб-

додатках, дозволяючи користувачам швидко та легко знаходити відповідну інформацію;

- керування контентом, створеним користувачами: SQL дозволяє розробникам зберігати, керувати та модерувати контент, створений користувачами, такий як коментарі, оцінки та огляди, сприяючи залученню користувачів і взаємодії з веб-додатками;
- реєстрація та моніторинг активності користувачів: SQL можна використовувати для реєстрації та аналізу активності користувачів у веб-додатках, надаючи розуміння поведінки користувачів і допомагаючи розробникам оптимізувати взаємодію з ними.

SQL можна легко інтегрувати з іншим програмним забезпеченням і системами, такими як системи управління взаємовідносинами з клієнтами (CRM), системи планування ресурсів підприємства (ERP) і системи географічної інформації (GIS). Ця інтеграція дозволяє організаціям використовувати потужність SQL для керування та аналізу даних у різних бізнес-функціях і доменах [18].

Основні завдання програмної та системної інтеграції, які підтримує SQL, включають:

- синхронізація даних між системами: SQL можна використовувати для вилучення, перетворення та завантаження даних між різними системами, забезпечуючи послідовність і актуальність даних у всій організації;
- об'єднання даних з різних джерел: SQL дозволяє користувачам запитувати та об'єднувати дані з кількох джерел, дозволяючи їм виконувати комплексний аналіз, який охоплює декілька систем і доменів;
- автоматизація процесів, керованих даними: SQL можна використовувати для автоматизації різних процесів, керованих даними, в інтегрованих системах, таких як оновлення рівня запасів у системі ERP на основі даних про продажі з системи CRM;

- розширення функціональності системи за допомогою користувацьких запитів SQL: розширюваність SQL дозволяє користувачам створювати користувацькі запити та функції, які розширюють можливості інтегрованих систем, дозволяючи їм вирішувати конкретні бізнес-потреби чи завдання;
- полегшення прийняття рішень на основі даних у всій організації: інтегруючи SQL з іншими системами, організації можуть отримувати доступ і аналізувати дані з різних джерел, що дає їм змогу приймати обґрунтовані рішення, які стимулюють зростання та інновації.

SQL відіграє вирішальну роль в управлінні доступом користувачів і дозволами в реляційних базах даних, забезпечуючи безпеку даних і відповідність політикам організації. SQL має компонент, який дозволяє адміністраторам баз даних контролювати привілеї користувачів і підтримувати безпечне середовище бази даних.

Основні завдання керування користувачами та доступом, які підтримує SQL, включають:

- створення та керування обліковими записами користувачів: SQL можна використовувати для створення та керування обліковими записами користувачів у базі даних, дозволяючи адміністраторам визначати конкретні ролі та обов'язки для кожного користувача;
- визначення рівнів доступу користувачів: SQL дозволяє адміністраторам визначати та запроваджувати рівні доступу для користувачів, гарантуючи, що вони можуть отримати доступ лише до даних і функцій, які стосуються їх ролі;
- надання та скасування дозволів: за допомогою команд GRANT і REVOKE адміністратори бази даних можуть призначати та скасовувати дозволи для користувачів, контролюючи їхній доступ до певних об'єктів бази даних, таких як таблиці, представлення та збережені процедури;

- впровадження контролю доступу на основі ролей: SQL підтримує концепцію ролей, які можна використовувати для групування користувачів із подібними вимогами до доступу, що полегшує керування та підтримку дозволів у всій організації;
- аудит активності користувачів: SQL можна використовувати для реєстрації та моніторингу активності користувачів у базі даних, надаючи розуміння поведінки користувачів і допомагаючи адміністраторам виявляти неавторизований доступ або потенційні порушення безпеки;
- забезпечення конфіденційності даних і відповідності вимогам. Ефективно керуючи доступом і дозволами користувачів, SQL допомагає організаціям підтримувати конфіденційність даних і відповідати відповідним нормам, таким як Загальний регламент захисту даних (GDPR) або Закон про переносимість і підзвітність медичного страхування (HIPAA).

У динамічному бізнес-середовищі ефективне керування та резервне копіювання баз даних SQL є вирішальними для забезпечення безпеки та цілісності даних. Він передбачає реалізацію стратегій резервного копіювання, розуміння варіантів відновлення бази даних, а також вирішення типових проблем із базою даних SQL.

Резервне копіювання бази даних SQL має важливе значення для підтримки цілісності даних і забезпечення швидкого відновлення бази даних у разі втрати даних або відновлення після будь-яких неочікуваних проблем. Існує кілька способів зробити це, залежно від використовуваної системи бази даних і вимог до резервного копіювання.

Studio SQL Server Management Studio (SSMS) - це інструмент графічного інтерфейсу користувача, розроблений Microsoft, який можна використовувати для керування та резервного копіювання баз даних SQL.

Інший підхід до резервного копіювання баз даних SQL полягає у використанні інструментів командного рядка та сценаріїв для автоматизації та планування.

Відновлення бази даних SQL передбачає отримання бази даних із резервної копії та вирішення будь-яких типових проблем із базою даних, які могли виникнути. Залежно від характеру проблеми та доступних резервних копій можна застосувати різні методи відновлення для відновлення початкового стану бази даних.

Перш ніж відновлювати базу даних SQL, важливо визначити та діагностувати основні проблеми. Кілька поширених проблем можуть впливати на бази даних і вимагати відновлення:

- випадкове видалення або модифікація даних;
- збій або пошкодження обладнання;
- помилки програмного забезпечення;
- зловмисне програмне забезпечення або порушення безпеки;
- пошкодження бази даних через системні збої або збої живлення.

Виявлення цих проблем може вимагати аналізу журналів бази даних, перевірки журналів системних подій або використання спеціальних інструментів і методів усунення несправностей бази даних [9].

Після виявлення проблем наступним кроком буде відновлення бази даних SQL за допомогою доступних резервних копій. Залежно від типу резервної копії та серйозності проблеми можна застосувати різні методи відновлення:

- повне відновлення бази даних: передбачає відновлення всієї бази даних із повної резервної копії та застосування подальших диференціальних резервних копій або резервних копій журналу транзакцій, якщо вони доступні;
- часткове відновлення бази даних: відновлює певні компоненти або об'єкти бази даних, наприклад таблиці чи індекси, з резервної копії.

Цей метод корисний, коли вражено лише частину бази даних і повне відновлення не потрібне;

- відновлення на певний момент часу: відновлює базу даних до певного моменту часу за допомогою повної, диференціальної резервної копії та резервної копії журналу транзакцій. Цей метод корисний у разі втрати даних через випадкову модифікацію або зараження шкідливим програмним забезпеченням.

Процес відновлення бази даних SQL буде відрізнятися залежно від використовуваної системи керування базою даних. Наприклад, SQL Server пропонує такі графічні інструменти, як SQL Server Management Studio, для відновлення баз даних, тоді як MySQL і PostgreSQL покладаються на утиліти командного рядка для виконання відновлення бази даних.

### 2.3 Вилучення інформації з баз даних

Вилучення даних – це процес отримання або отримання даних із різних джерел і перетворення їх у придатний для використання формат для подальшого аналізу, звітування чи зберігання. Це один із найважливіших кроків в управлінні даними, який дає змогу передавати дані в додатки або аналітику. Дані можуть надходити з різних джерел, включаючи бази даних, електронні таблиці, веб-сайти, інтерфейси прикладного програмування (API), файли журналів, дані датчиків тощо. Ці джерела можуть бути структурованими (організованими в таблиці або записи) або неструктурованими (текстові або нетабличні дані).

Видобуток даних відносно простий, коли ви маєте справу зі структурованими даними, такими як табличні дані у файлах Excel або реляційних базах даних. Однак краще використовувати програмне забезпечення для вилучення даних, якщо мова йде про неструктуровані джерела даних, такі як PDF-файли, електронні листи, зображення та відео [19].



Важлива проблема, яку вирішує вилучення даних, полягає в тому, що воно покращує доступність даних. Уявіть собі бізнес із різними джерелами даних, і всі ці дані мають різні формати, і кожен відділ намагається використовувати ці дані відповідно до своїх потреб; скільки безладу це створить! Вилучення даних збирає всі дані разом, дозволяючи конвертувати їх у стандартизований формат, а потім помістити в централізоване джерело, щоб кожен міг використовувати їх за потреби. Результатом стає більше можливостей користувачів, які можуть використовувати дані, не покладаючись на ІТ-ресурси.

Витяг даних і аналіз даних часто плутають одне з одним. Однак між ними є різниця. Як ми пояснювали раніше, вилучення даних – це збір даних із різних джерел і підготовка їх для аналізу чи зберігання в структурованій базі даних. Інтелектуальний аналіз даних, з іншого боку, - це процес виявлення закономірностей, тенденцій, ідей або цінних знань із набору даних. Йдеться про застосування різноманітних методів статистики, машинного навчання та аналізу даних для отримання корисної інформації з даних. Основна мета інтелектуального аналізу даних полягає в тому, щоб виявити приховані шаблони або зв'язки в даних, а потім використовувати їх для прийняття рішень або прогнозного моделювання (табл. 2.1) [10].

Процес вилучення даних починається з ідентифікації джерел даних. Ви повинні чітко визначити, які дані вам потрібні та де вони розташовані. Це може бути в документах, базах даних або програмах соціальних мереж. Визначивши джерела даних, потрібно вибрати відповідний метод для кожного джерела. Для зображень вам може знадобитися OCR; для веб-сайтів вам може знадобитися програмне забезпечення для сканування веб-сторінок і так далі і так далі.

Потім вам потрібно встановити з'єднання з вибраними джерелами даних. Спосіб підключення може відрізнитися залежно від типу джерела. Ви можете використовувати рядок підключення до бази даних, ім'я користувача та пароль для баз даних. Вам може знадобитися використовувати API для веб-

джерел. Деяке програмне забезпечення для вилучення даних пропонує комплексне рішення з різноманітними вбудованими з'єднувачами, щоб можна було підключитися до всіх джерел одночасно.

Таблиця 2.1 - Порівняння видобутку та витягу даних

	Видобуток даних	Витяг даних
Мета	У центрі уваги інтелектуального аналізу даних – отримати з даних корисну інформацію. Ви можете використовувати інтелектуальний аналіз даних, щоб виявити зв'язки, зробити прогнози, виявити тенденції або знайти аномалії в даних.	Вилучення даних спрямоване на збір, очищення та перетворення даних у послідовний і структурований формат, щоб користувачі мали надійний набір даних для запитів або аналізу.
Спосіб	Інтелектуальний аналіз даних часто вимагає глибокого розуміння статистичного аналізу та машинного навчання. Він використовує різні методи та алгоритми, включаючи кластеризацію, класифікацію, регресію, аналіз правил асоціації та виявлення аномалій.	Вилучення даних зазвичай включає прийом даних, синтаксичний аналіз і методи перетворення. Інструменти та методи, які зазвичай використовуються для вилучення даних, включають веб-збирання, розбір документів, вилучення тексту та вилучення даних на основі API.
Результат	Вихідним результатом інтелектуального аналізу даних є практична інформація або шаблони, які можна використовувати для прийняття обґрунтованих рішень або створення прогнозних моделей. Ці статистичні дані можуть включати тенденції, кореляції, кластери подібних точок даних або правила, які описують асоціації в даних.	Результатом вилучення даних є структурований набір даних, готовий для аналізу. Це може передбачати очищення даних для видалення невідповідностей, відсутніх значень або помилок. Видобуті дані зазвичай зберігаються у форматі, придатному для запитів або аналізу, наприклад у реляційній базі даних.
Час	Інтелектуальний аналіз даних виконується після того, як дані видобуто, очищено, перетворено та перевірено.	Вилучення даних зазвичай є початковим етапом аналізу, який виконується перед будь-яким поглибленим дослідженням або моделюванням.

Ви можете використовувати запити SQL для отримання певних даних із таблиць для баз даних. Для документів може знадобитися вилучення тексту за допомогою оптичного розпізнавання символів (або спеціальних аналізаторів документів). Більшість інструментів вилучення даних тепер не містять коду, що означає, що все, що вам потрібно зробити, це просто

перетягнути конектор і підключитися до будь-якого джерела даних, не вивчаючи розширених запитів SQL або програмування мови.

Після вилучення даних вони часто не відповідають формату, який вимагається для кінцевого призначення або навіть для аналізу. Наприклад, у вас можуть бути дані у форматі XML або JSON, і вам може знадобитися конвертувати їх у Excel для аналізу. Може бути кілька сценаріїв, тому трансформація даних є важливою. Деякі поширені завдання трансформації включають:

- очищення даних для видалення дублікатів, обробки відсутніх значень і виправлення помилок;
- нормалізація даних шляхом перетворення форматів дати або стандартизації одиниць вимірювання;
- збагачення даних шляхом додавання зовнішньої інформації або обчислюваних полів.

Потім перетворені дані надсилаються в пункт призначення, який змінюється залежно від мети даних. Ви можете зберігати дані в плоских файлах, таких як файли CSV, JSON або Parquet, або помістити їх у реляційну базу даних (наприклад, MySQL, PostgreSQL) або базу даних NoSQL (наприклад, MongoDB).

Багато веб-служб надають API, які дозволяють розробникам отримувати дані з програм у структурованому форматі. Вилучення на основі API передбачає надсилання запитів HTTP до цих API, а потім отримання даних. Це надійний і структурований спосіб отримання даних з онлайн-джерел, таких як соціальні медіа-платформи, погодні служби або постачальники фінансових даних. Вилучення тексту (обробка природної мови – NLP) Методи виділення тексту часто використовують обробку природної мови (NLP) для вилучення інформації з неструктурованих текстових даних, таких як документи, електронні листи чи публікації в соціальних мережах. Техніки НЛП включають розпізнавання іменованих

сутностей (NER) для вилучення сутностей, таких як імена, дати та місця, аналіз настроїв і класифікацію тексту для вилучення ідей із тексту [20].

Оптичне розпізнавання символів (OCR) перетворює друкований або рукописний текст із документів, зображень або відсканованих сторінок у машиночитані й редаговані текстові дані. Програмне забезпечення OCR аналізує оброблені зображення, щоб розпізнавати та перетворювати текстовий зміст у машинозчитувані символи. Механізми оптичного розпізнавання символів використовують різні методи для визначення почуттів, зокрема розпізнавання образів, виділення ознак і алгоритми машинного навчання.

Синтаксичний аналіз документа - це коли комп'ютерна програма або система витягує структуровану інформацію з неструктурованих або напівструктурованих документів. Ці документи можуть мати різні формати, наприклад PDF-файли, файли Word, HTML-сторінки, електронні листи або рукописні нотатки. Система аналізу визначає структуру документа. Потім він витягує відповідні елементи даних, включаючи імена, адреси, дати, номери рахунків-фактур і описи продуктів, на основі конкретних ключових слів, регулярних виразів або інших методів зіставлення шаблонів.

Після того як у вас є джерела даних і ви вирішите, яка методика чи методи працюють, вам потрібно налаштувати систему для роботи вилучення даних. Ви можете вибрати ручне вилучення даних, повне вилучення даних або поступове вилучення даних. Давайте подивимося на плюси та мінуси кожного типу вилучення даних:

Повне вилучення, повне завантаження чи оновлення витягує всі дані з вихідної системи за одну операцію. Цю техніку можна використовувати, коли вихідні дані не змінюються часто, а повна та актуальна копія даних є важливою. Однак повне вилучення даних може бути ресурсомістким, особливо для великих наборів даних, оскільки воно отримує всі дані незалежно від того, чи змінилися вони після попереднього вилучення. Часто

це найкращий вибір як початковий крок у проектах зі сховищ даних або міграції даних.

Інкрементне вилучення, яке також називається дельта-вилученням або зміною даних із записом (CDC), використовується для вилучення лише даних, які змінилися з часу останнього вилучення. Це найкращий вибір при роботі з джерелами даних, які часто змінюються, наприклад транзакційними базами даних. Крім того, це ефективніше, ніж повне вилучення, оскільки зменшує обсяг переданих і оброблених даних. Загальні методи інкрементного вилучення включають відстеження на основі часових позначок, номерів версій або використання прапорців для позначення оновлених записів.

У минулому більшість організацій видобували дані вручну. Деякі все ще копіюють і вставляють дані з документів, електронних таблиць або вебсторінок в іншу програму або базу даних. Однак ручне вилучення займає багато часу, може викликати помилки та неминуче не підходить для великомасштабних завдань вилучення даних. Тим не менш, це може бути корисним для випадкового або тимчасового отримання даних, коли автоматизація неможлива.

Однак підприємства все ще потребують допомоги з проблемами вилучення даних. Бізнес отримує дані в середньому з 400 джерел. Усі ці джерела мають різний формат, структуру та метод доступу, що ускладнює вилучення даних, до того ж своєчасне. Згідно з опитуванням, проведеним IDG, цей вибух у джерелах даних створює складне середовище, яке зупиняє проекти; фактично 32% опитаних людей зазначили, що їм потрібна допомога у підключенні до джерел даних.

Існує близько 4,95 мільярдів користувачів Інтернету, які щодня генерують близько 2,5 квінтільйонів байтів даних. Отже, проблемою є не лише різноманітність джерел даних, але й обсяг даних.

Переміщення великих обсягів даних із вихідних систем до центрального сховища може зайняти час, головним чином, якщо пропускна

здатність мережі організації обмежена. Крім того, керування великими обсягами даних також означає потенційні проблеми з керуванням даними.

Великі обсяги даних і різноманітність джерел даних це проблема, але це не закінчується - сьогодні дані складніші, ніж будь-коли. Прошли ті часи, коли він зберігався просто у двох таблицях у Excel. Сьогодні ви знайдете ієрархічні дані, файли JSON, зображення, PDF-файли тощо. Крім того, усі ці дані взаємопов'язані. Наприклад, у даних соціальних мереж люди пов'язані різними типами стосунків, такими як дружба, підписки, лайки та коментарі. Ці зв'язки створюють мережу взаємопов'язаних точок даних. А тепер уявіть, що ви отримуєте ці точки даних, а потім вписуєте їх у схему.

Обробка помилок і моніторинг є ключовими аспектами вилучення даних, оскільки вони забезпечують надійність і якість витягнутих даних. Це ще важливіше при вилученні даних у режимі реального часу, коли дані вимагають негайного виявлення та обробки помилок.

Багатьом організаціям потрібне вилучення та аналіз даних у режимі реального або майже в реальному часі. Оскільки потоки даних безперервно надходять, системи повинні встигати за темпом прийому даних, тому масштабованість є важливою. Налаштовуючи свою інфраструктуру, ви повинні переконатися, що вона може впоратися з будь-яким зростанням обсягу даних.

Зважаючи на те, що дані стали складнішими, єдиним рішенням для вирішення проблем із вилученням даних є використання інструменту вилучення даних, який може автоматизувати більшість завдань. Ось деякі з переваг використання інструменту вилучення даних порівняно з ручним вилученням даних:

- працюйте з кількома джерелами даних. Інструменти вилучення даних мають вбудовані конектори, які спрощують підключення до всіх джерел даних одночасно, включаючи веб-сайти, бази даних, електронні таблиці, PDF-файли, електронні листи та API. Крім того, інструменти вилучення даних сьогодні оснащені можливостями ШІ,

які можуть витягувати дані з неструктурованих документів за допомогою потужних алгоритмів ШІ;

- масштабованість: найкраща частина інструментів вилучення даних полягає в тому, що вони можуть масштабуватися для ефективної обробки великих обсягів даних. Вони можуть видобувати та обробляти дані пакетами або безперервно, щоб задовольнити потреби компаній із зростаючими вимогами до даних;
- якість даних: багато інструментів вилучення даних містять функції перевірки та очищення даних, які допомагають виявити та виправити помилки чи невідповідності в витягнутих даних;
- автоматизація. Інструменти вилучення даних можна запланувати для запуску через певні проміжки часу або за певними подіями, що зменшує потребу в ручному втручанні та гарантує постійне оновлення даних.

Інтернет речей (IoT) також трансформує ландшафт даних. Окрім стільникових телефонів, планшетів і комп'ютерів, дані тепер генеруються такими переносними пристроями, як FitBit, автомобілями, побутовою технікою та навіть медичними приладами. Результатом є постійно зростаючий обсяг даних, які можуть бути використані для підвищення конкурентоспроможності компанії після того, як дані були вилучені та перетворені

### 3 ДОСЛІДЖЕННЯ ГРАФОВИХ МОДЕЛЕЙ ДЛЯ ОПИСУ І ОБРОБКИ ЧАСТКОВО СТРУКТУРОВАНОГО КОНТЕНТУ

#### 3.1 Опис підприємства

Skandholz – це німецька фірма, що виробляє різноманітні товари з дерева. У перелік товарів входять такі вироби як двері, вікна, бесідки, меблі та інше. Усі товари можуть бути вироблені з різних порід дерев та/або бути пофарбовані в потрібний колір. Також, товари можуть мати різні розмірні характеристики.

Як підприємство воно є відокремленим цілісним виробничо-господарським організмом, у якому передбачається соціальна, виробничо-технічна, організаційно-адміністративна єдність, а також фінансово-економічна самостійність.

Отже, виробничо-технічна єдність передбачає:

- відповідність наявного технологічного обладнання та виробничих площ характеру виробництва;
- послідовний зв'язок та завершеність всіх технологічних процесів;
- єдність технічного та виробничого керівництва.

Організаційно-адміністративна єдність передбачає, що апарат управління організує виробничо-господарську діяльність трудового колективу на основі плану, веде облік та звітність підприємства.

Структура підприємства - це його внутрішня будова, сукупність складових елементів і систему зв'язків між ними, об'єднаних в єдиний господарський організм.

Задля більшої життєздатної діяльності підприємства необхідно створення виробничої та організаційної структур управління, тобто. утворення функціональних областей та виробничих підрозділів, кожен з яких матиме своє певне призначення та стійкі зв'язки з іншими підрозділами



внаслідок горизонтального поділу праці, та формування рівнів управління по вертикалі.

Основою для вибору організаційних структур на підприємствах є зазвичай перспективні плани їх розвитку, обсяги виробництва продукції, нормативи чисельності та співвідношень різних категорій персоналу та інші фактори.

Удосконалення виробничої інфраструктури одна із чинників поліпшення діяльності підприємства. Невиробнича інфраструктура підприємства створюється соціального обслуговування працівників підприємства. Вона включає житлово-комунальні структури, дитячі садки, лікарні, санаторії, оздоровчі комплекси, навчальні заклади та інші необхідні служби (рис. 3.1).

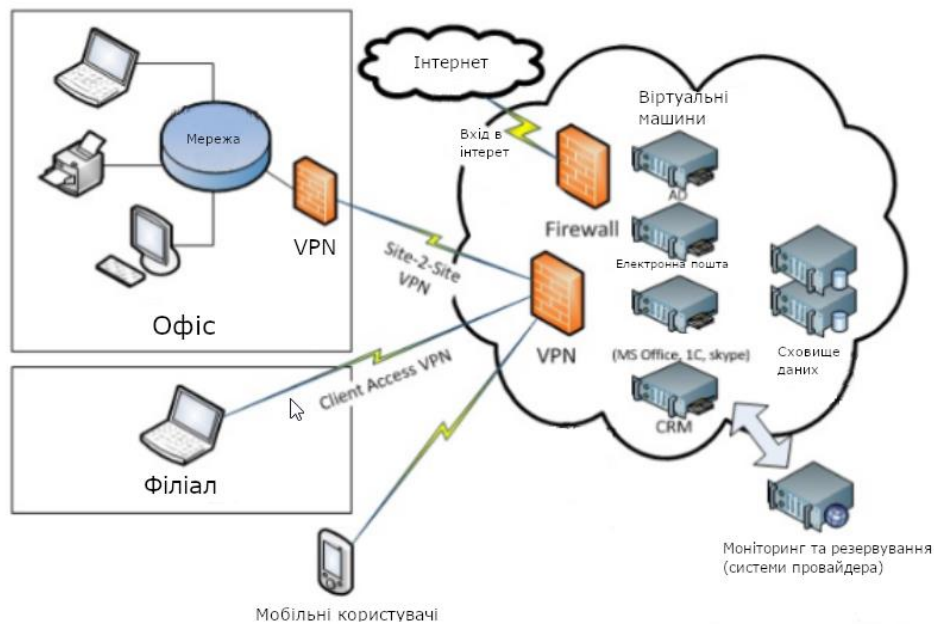


Рисунок 3.1 - IT-інфраструктура підприємства

Етапи створення IT-інфраструктури компанії:

- розробка та затвердження технічного завдання. Технічне завдання є документом, який включає всі вимоги замовника до створюваної інформаційної системи.

- розробка проекту. Після затвердження технічного завдання розробляється робочий проект – документ, що містить технічний опис реалізації вимог, зазначених у технічному завданні.
- використання. На етапі застосування відбувається фізична реалізація розробленого проекту.
- створення виконавчої документації. Завершальним етапом створення ІТ-інфраструктури є створення виконавчої документації. Виконавча документація містить докладний опис ІТ-інфраструктури, необхідний для використання та супроводження створеної корпоративної інформаційної системи.

Побудова спроектованої грамотно ІТ-інфраструктури дає низку переваг:

- оптимізація поточних витрат;
- мінімізація витрат під час масштабування ІТ-інфраструктури;
- мінімізація ризиків, пов'язаних з інтеграцією, оновленням або заміною компонентів ІТ-інфраструктури;
- підвищення продуктивності, захищеності та керованості ІТ-сервісів;
- чітке визначення завдань ІТ-служби у компанії, прогнозованість її завантаженості та результативності, оптимізація складу, планування підвищення кваліфікації ІТ-фахівців;
- можливість чіткої постановки завдань, розв'язуваних з допомогою ІТ, визначення критеріїв успішного вирішення основних завдань ІТ і їх реалізації.

Узгодження між окремими вузлами та блоками виконують за допомогою перехідних апаратно-логічних пристроїв, які називають апаратними інтерфейсами. Стандарти на апаратні інтерфейси називають протоколами. Таким чином, протокол - це сукупність технічних умов, які мають бути забезпечені розробниками пристроїв для успішного узгодження їх роботи з іншими пристроями.

Програмне забезпечення - сукупність програм, системи обробки інформації та програмних документів, необхідні експлуатації цих програм. Кінцева мета будь-якої комп'ютерної програми - управління апаратними засобами.

Між програмами, як і між фізичними вузлами та блоками, існує взаємозв'язок - багато програм працюють, спираючись на інші програми нижчого рівня, тобто ми можемо говорити про міжпрограмний інтерфейс. Можливість існування такого інтерфейсу теж заснована на існуванні технічних умов і протоколів взаємодії, а на практиці він забезпечується розподілом програмного забезпечення на кілька рівнів, що взаємодіють між собою.

Як виробник, компанія продає товари сама на своєму сайті а також продає свою продукцію різним магазинам-ретейлерам.

На рис. 3.2 можна побачити схему роботи підприємства з даними, що в результаті будуть надані магазинам-ретейлерам.

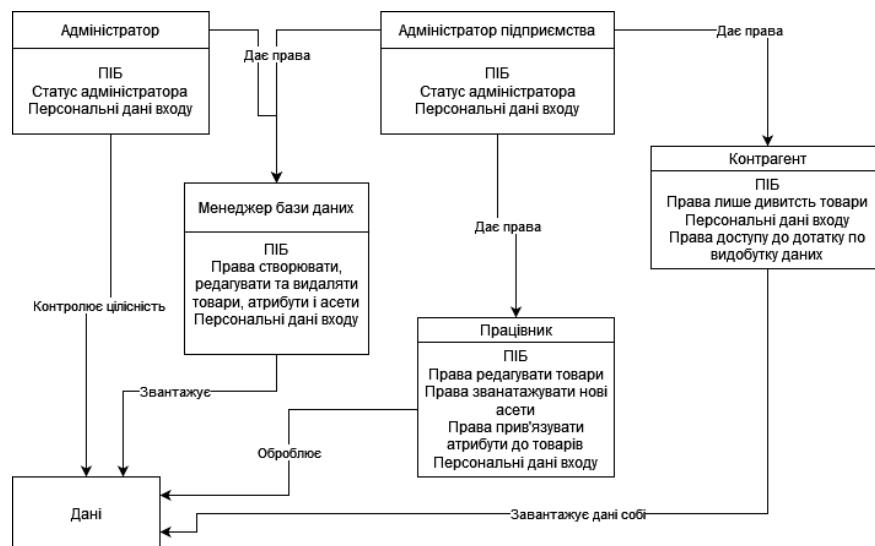


Рисунок 3.2 - схема роботи підприємства з даними

Як виглядає аккаунт в інформаційній системі можна побачити в табл. 3.1:

Таблиця 3.1 Основна структура аккаунту

Назва	Тип	Опис
Id	Int	Унікальний ідентифікатор користувача
First_name	Varchar	Ім'я
Last_name	Varchar	Прізвище
Login	Varchar	Логін
Password	Varchar	Пароль
Email	Varchar	Пошта
Phone	Varchar	Телефон
Company	Varchar	Представником якої компанії є
Position	Varchar	Посада в компанії (якщо є компанія)

```

CREATE TABLE users (
  user_id INT NOT NULL AUTO_INCREMENT,
  first_name VARCHAR(255) NOT NULL,
  address VARCHAR(255) NOT NULL,
  phone_number VARCHAR(255) NOT NULL,
  ...
  PRIMARY KEY (user_id)
);

```

Цей код створює таблицю користувачів інформаційної системи (рис. 3.3).

Адміністратори є як з боку підприємства так і в вигляді підтримки. Але, підтримка лише налаштовує дані спочатку а потім лише контролює цілісність даних. Адміністратори зі сторони підприємства навпаки вирішують хто буде мати які права та допуски до різних видів даних. Саме тому вони в першу чергу видають права на редагування та/або перегляд чи редагування даних для представників всіх інших ролей.

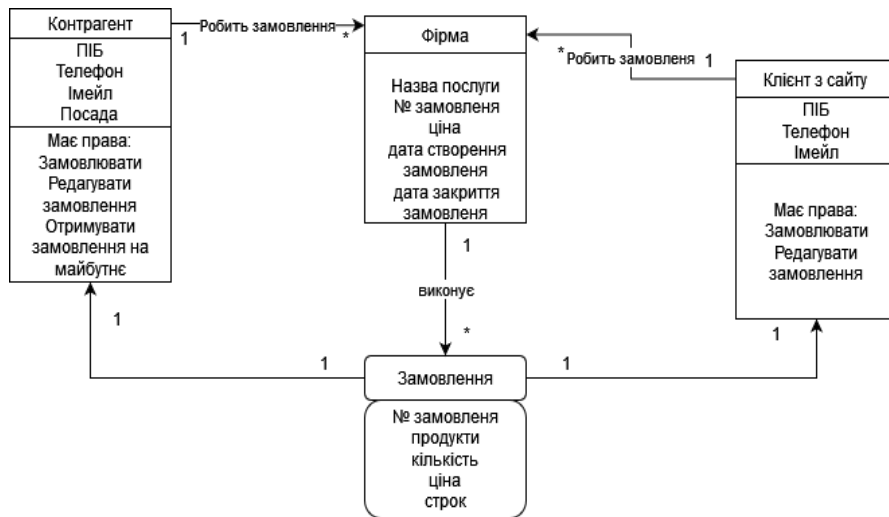


Рисунок 3.3 - Схема замовлень в інформаційній системі підприємства

Менеджери бази даних виконують основну роботу по наповненню її змістом. Вони створюють та редагують більшість товарів, слідкують щоб інформація зазначена в базі даних відповідала дійсності та щоб увесь асортимент був представленим.

Працівники підприємства небагато використовують базу даних щоденно, але під час діяльності можуть виникнути якісь зміни, тому працівники повинні оновити інформацію в базі даних згідно зі своїми обов'язками. Однак, на відміну від менеджерів бази даних, їх права обмежені лише тими сферами до яких вони безпосередньо відносяться згідно своїм основним обов'язкам.

Контрагенти теж мають свої акаунти у базі, але їхні права обмежені лише тим що їм необхідно бачити. Вони не можуть якимось впливати на будь-що в базі, лише дивитись ті дані що їм можна. Також їм надано доступ до веб-додатку для виводу даних у вигляді архіву.

### 3.2 Створення структури інформаційної системи підприємства

Замовником було сформульовано набір вимог до інформаційної системи за такими критеріями: функціональні можливості, технічна реалізація, захист інформації, режими функціонування, ергономіка. Розробка системи ґрунтувалася на сформульованих вимогах.

Під час розробки інформаційної системи повинні враховуватися такі вимоги до функціональних можливостей:

- додавання продуктів, їх опис;
- призначення відповідального за продукт працівника;
- надання інформації про прострочені продукти;
- створення зручного виводу даних для контрагентів.

Під час розробки повинні враховуватися такі вимоги до технічної реалізації:

- реалізація у вигляді веб-системи;
- створення єдиної інформаційної бази у межах всієї організації;
- використання реляційної моделі представлення даних;
- функціонування програми в операційних системах Microsoft Windows.

Під час розробки повинні враховуватися такі вимоги щодо захисту інформації:

- надання доступу до системи управління проектами лише зареєстрованим працівникам та довіреним контрагентам;
- захист інформації слід забезпечити на основі регулювання прав доступу до інформаційних ресурсів під час їх читання, запису, створення або знищення інформації з використанням штатних засобів СУБД.

Система управління проектами має функціонувати у таких режимах: адміністратор, працівник, контрагент (рис. 3.2).

- У режимі «адміністратор» допускаються будь-які операції.
- У режимі «працівник» допускаються будь-які операції:
- редагування продукту; створення, редагування асетів;

- можливість зміни статусу продуктів та асетів;
- призначення відповідального виконання завдання;
- доступ до візуалізації прогресу виконання проекту;
- перегляд сторонніх продуктів.

У режимі «контрагент» допускаються такі операції:

- перегляд всіх доступних до продажу продуктів;
- перегляд візуалізації прогресу продуктів;
- завантаження даних про продукти у вигляді архіву.

При розробці повинні враховуватися такі вимоги до ергономіки та технічної естетики:

- однозначність значень користувачем пунктів меню;
- підбір кольорової гами, що забезпечує чіткість сприйняття символів на екрані поряд із мінімальною втомою зору;
- супровід помилкових дій індикацією на екрані та підказкою про подальші дії.

Для розробки інформаційної системи управління проектами було використано ASP.NET - технологію створення веб-додатків та веб-сервісів від компанії Microsoft. Більшість веб-проектів у світі зараз реалізується саме за допомогою цієї технології.

Переваги ASP.NET:

- код, що компілюється, виконується швидше, більшість помилок відловлюється ще на стадії розробки;
- розширюваний набір елементів управління та бібліотек класів дозволяє швидше розробляти програми;
- спирається на багатомовні можливості .NET, що дозволяє писати код сторінок, на VB.NET, Delphi.NET, Visual C#, J#;
- можливість поділу візуальної частини та бізнес-логіки по різних файлах;

- наявність тазег-сторінок для встановлення шаблонів оформлення сторінок. Мовою для розробки Web-програми обрано C.

C# - об'єктно-орієнтована мова програмування від компанії Microsoft. Вона була розроблена наприкінці 90-х років як частина загальної стратегії і вперше випущена у вигляді альфа-версії у середині 2000 року.

Як середовище розробки було використано Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів.

Microsoft SQL Server - система управління реляційними базами даних (СУРБД), розроблена корпорацією Microsoft. Використовується до роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства.

Переваги використання СУБД MS SQL:

- надійність;
- висока швидкість роботи;
- резервне копіювання;
- масштабованість;
- можливість доступу необмеженої кількості користувачів до інформаційної системи;
- можливість сортування, пошуку даних.

Система складається з таких компонентів як база даних, веб-інтерфейс, додаток.

База даних містить інформацію про всі будівельні матеріали, які використовуються на об'єкті. Веб-інтерфейс дозволяє користувачам вводити та отримувати інформацію з бази даних. Додаток дозволяє користувачам отримувати інформацію про будівельні матеріали в режимі реального часу.

Система має основні функції ведення обліку будівельних матеріалів (характеристика товарів), планування закупівель будівельних матеріалів, контроль використання будівельних матеріалів



Інформація у інформаційній системі зберігається як таблиці. Такий спосіб подання інформації називається реляційною моделлю даних. Таблиці складаються з рядків та стовпців (полів). Окремий рядок таблиці складається з полів та містить інформацію про один конкретний об'єкт.

На рис. 3.4 можна побачити спрощений вигляд схеми даних інтегрованої системи, що стосується роботи з контрагентами. Вона складається з кількох таблиць:

- таблиця «Attribute» - інформація про атрибути продуктів;
- таблиця "Channel" - інформація про канали;
- таблиця "Product" - інформація про товари;
- таблиця «Asset» - інформація асети товарів;
- таблиця "Price" - інформація про ціни товарів.

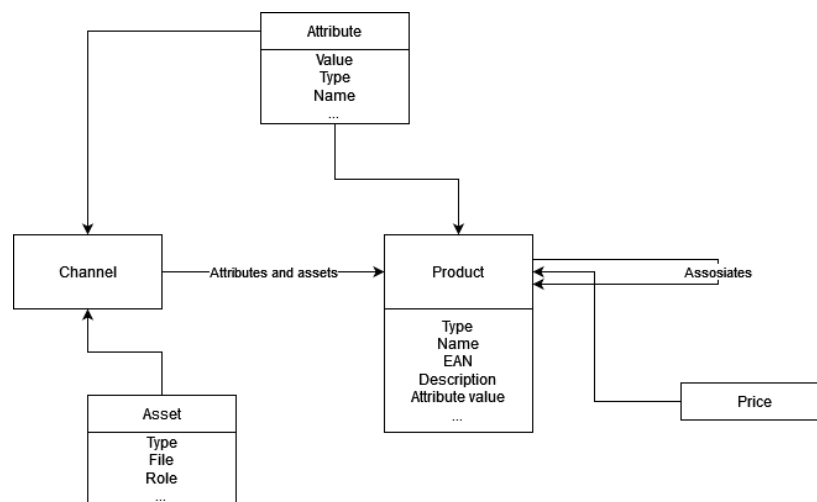


Рисунок 3.4 - Схема даних в базі даних підприємства

Основою даних є таблиця продукту (табл. 3.2). Саме навколо неї консолідуються інші таблиці. Вони типізують декі види даних що можуть стосуватись продукту але є не у всіх продуктів. Також продукти можуть мати асоційовані продукти - це ті продукти, що за якимись параметрами відносяться до даного продукту. Наприклад, це продукти, що часто купують разом з даним продуктом.

```

CREATE TABLE products (
  product_id INT NOT NULL AUTO_INCREMENT,
  product_name VARCHAR(255),
  description VARCHAR(255),
  product_type LIST(product_type,
  product_EAN VARCHAR(255),
  REFERENCES attributes (attribute_value _id), assets
(asset_id), channels (attribute_value _id, asset_id)
);

```

Цей код створює таблицю атрибутів у інформаційній системі.

Таблиця 3.2 - Основна структура продукту

Назва	Тип	Опис
Id	int	Унікальний ідентифікатор продукту
Name	varchar	Ім'я продукту
Type	list	Тип продукту
EAN	int	Унікальний ідентифікатор продукту за стандартом ЄС
Description	varchar	Опис товару
Attribute value	link	Значення атрибуту

Кожен продукт може мати декілька різних цін, тому це теж окрема таблиця.

До кожного продукту прив'язні канали (табл. 3.3). Канали – це «лінзи» за допомогою яких може фільтруватись інформація специфічна для різних країн, наприклад дані на їх мові. Хоча компанія та її контрагенті і німецькомовні, але покупці є мешканцями всіх країн європейської спільноти.

```

CREATE TABLE channels (
  channel_id INT NOT NULL AUTO_INCREMENT,
  channel_name VARCHAR(255) NOT NULL,
  description VARCHAR(255),
  channel_language LIST(channel_language) NOT NULL,
  FOREIGN KEY (product_id) REFERENCES attributes
(attribute_value _id), assets (asset_id)
);

```

Цей код створює таблицю атрибутів у інформаційній системі.

Таблиця 3.3 - Основна структура каналу

Назва	Тип	Опис
Id	int	Унікальний ідентифікатор каналу
Name	nvarchar	Назва каналу
Language	list	Мова каналу

Саме тому асети (табл. 3.4) (тобто картинки та пдф файли) з'єднані з товарами через канали. Однак, одна картинка чи пдф може бути приєднана до багатьох товарів одночасно. Так, наприклад, для дверей різного кольору одна й та сама інструкція для збирання абсолютно підходить, але для однієї двері вона може бути на різних мовах.

```
CREATE TABLE assets (
  asset_id INT NOT NULL AUTO_INCREMENT,
  asset_file LIST NOT NULL (255),
  asset_type LIST(asset_type) NOT NULL,
  FOREIGN KEY (product_id, channel_id)
);
```

Цей код створює таблицю асетів у інформаційній системі.

Таблиця 3.4 - Основна структура асету

Назва	Тип	Опис
Id	int	Унікальний ідентифікатор асету
Type	list	Вид асету
File	link	Посилання на файл
Role	list	Роль асету (важлива для контрагента)

Атрибути (табл. 3.5) (тобто властивості товару) можуть бути специфічними для деяких країн але є і універсальні. Значення специфічних атрибутів також передається через призму каналу.

```
CREATE TABLE attributes (
  attribute_id INT NOT NULL AUTO_INCREMENT,
  attribute_name VARCHAR(255) NOT NULL,
  description VARCHAR(255),
  attribute_type LIST(attribute_type) NOT NULL,
  FOREIGN KEY (product_id, channel_id) REFERENCES attribute_value
(attribute_value_id)
);
```

Цей код створює таблицю атрибутів у інформаційній системі.

Таблиця 3.5 - Основна структура атрибуту

Назва	Тип	Опис
Id	int	Унікальний ідентифікатор атрибуту
Name	nvarchar	Назва атрибуту
Value	nvarchar/list/int/... (depends on Type)	Значення атрибуту
Type	list	Вид атрибуту (задає параметри для значення)

Для аналізу того, як ефективно функціонує магазин будівельних матеріалів і для подальшого вивчення організації закупівельних процесів з погляду надійності постачальників, розумно здійснити моделювання відповідних бізнес-процесів за допомогою створення діаграм IDEF.

Діаграма розкладу процесу "Робота відділу з закупівлею" включає кілька підпроцесів (рис. 3.5): прийняття рішення щодо закупівлі товару; оцінка характеристик постачальників; проведення закупівлі товарів – формування замовлення у вибраних постачальників; інформування клієнтів; взаємодія з клієнтами.

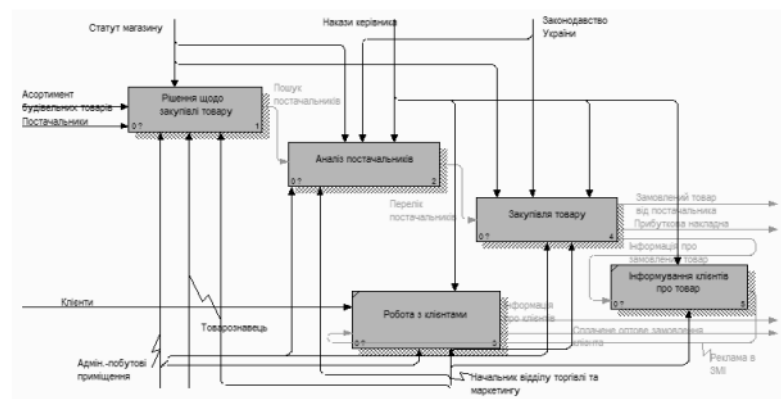


Рисунок 3.5 – Діаграма роботи відділу з закупівель

У цьому дослідженні було вирішено зосередити увагу на підпроцесах, пов'язаних із постачальниками, а не з клієнтами. Розглянемо розбиття на етапи робіт у підпроцесах "Процес закупівлі товару" та "Оцінка характеристик постачальників".

На рис. 3.6 представлено діаграму «Процесу закупівлі товару».

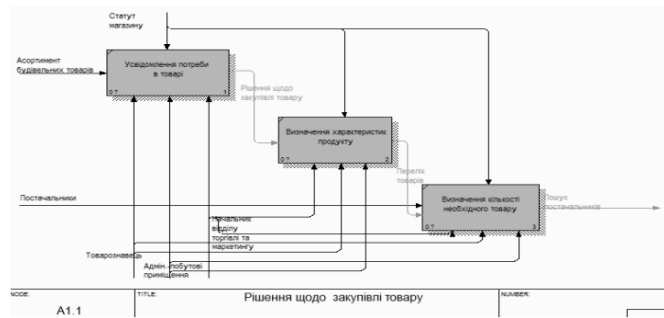


Рисунок 3 – Діаграма «Рішення щодо закупівлі товару»

### Рисунок 3.6 – Діаграма процесу закупівлі товару

На рис. 3.7 представлена діаграма, що відображає процес "Характеристика постачальників". У результаті попереднього етапу було прийнято рішення щодо закупівлі будівельних матеріалів.

Начальник відділу торгівлі формує запит на конкретні товарні позиції у визначеній кількості. Ці запити надсилаються постачальникам, які включені у визначений перелік. Отримані відповіді від постачальників аналізуються, приділяючи особливу увагу цінам товарів, вказаним у їх прайс-листах. Після оцінки постачальників начальник відділу торгівлі та маркетингу приймає рішення стосовно вибору відповідного постачальника і вносить його до списку узгоджених постачальників.

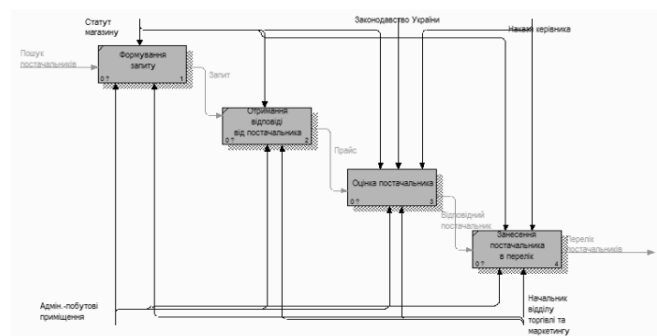


Рисунок 3.7 – Характеристика постачальника

Після аналізу діяльності відділу закупівель було виявлено, що процес "Характеристка постачальника" не має наукового обґрунтування, при цьому основним критерієм є лише суб'єктивно визначена надійність поставок та ціни на будівельні товари.

## 4 ЗАСТОСУВАННЯ БАЗ ДАНИХ ДЛЯ ВИРОБНИЦТВА

### 4.1 Розробка програми згідно з вимогами клієнта

Спочатку було проаналізовано звернення клієнта з проханням створити сторінку, де представники його контрагентів могли б завантажувати всю необхідну інформацію про вироблені клієнтом товари. За основу необхідно було взяти стару версію додатку для видобутку інформації, що була розроблена для попередньої бази даних та впровадити запропоновані покращення. Базовий функціонал мав залишитись колишнім, проте необхідно брати дані з іншої бази даних.

Зовнішній вигляд попередньої версії сторінки можна побачити на рис.

4.1:

Händlerauswahl:

Produktauswahl:

Bitte nun semikolon-basiert die gewünschten EAN-Codes in das unten folgende Textfeld einfügen:

Bilder nur einmal exportieren (ArticleImages.csv)  
(Ignoriert das Namensschema für alle Bilder)

Namensschema:

Platzhalter	Bedeutung
{EAN}	EAN
{LAN}	Artikelnummer
{code}	Merchant-Artikelnummer
{name}	Bezeichnung des Bildes bspw. "Bild_1" oder "Aufbauanleitung"
{original}	Originalname der Datei

**Dokumente**

Datenblatt:

Aufbauanleitung:

**360°-Bilder**

Im Stammverzeichnis

In zusätzlichem Unterordner mit dem Namen:

**Bildtypen**

<input checked="" type="checkbox"/> Bild 1	<input checked="" type="checkbox"/> Animation mit Hintergrund	<input checked="" type="checkbox"/> Grundriss	<input checked="" type="checkbox"/> Detail 1	<input checked="" type="checkbox"/> Detail 2
<input checked="" type="checkbox"/> Bild 2	<input checked="" type="checkbox"/> Animation ohne Hintergrund	<input checked="" type="checkbox"/> Schnitt	<input checked="" type="checkbox"/> Detail 3	<input checked="" type="checkbox"/> Detail 4
<input checked="" type="checkbox"/> Bild 3		<input checked="" type="checkbox"/> Ansicht vorne	<input checked="" type="checkbox"/> Detail 5	<input checked="" type="checkbox"/> Detail 6
<input checked="" type="checkbox"/> Bild 4		<input checked="" type="checkbox"/> Ansicht hinten	<input checked="" type="checkbox"/> Detail 7	<input checked="" type="checkbox"/> Detail 8
<input checked="" type="checkbox"/> Bild 5		<input checked="" type="checkbox"/> Ansicht links	<input checked="" type="checkbox"/> Detail 9	<input checked="" type="checkbox"/> Detail 10
		<input checked="" type="checkbox"/> Ansicht rechts	<input checked="" type="checkbox"/> Detail 11	<input checked="" type="checkbox"/> Detail 12

Рисунок 4.1 – Зовнішній вигляд попередньої версії сторінки.

Також необхідно зазначити, що попередня сторінка не мала всього зазначеного в ній функціоналу. Деякі функції в ній були присутні лише в інтерфейсі, але реальних дій при натисканні цих кнопок не виникало. Це, зокрема, такі елементи як ПДФ документ (Datenblatt), інструкція зі збирання (Aufbauanleitung) та картинки з видом 360 ° градусів (360 °-Bilder).

Перед початком роботи необхідно було проаналізувати роботу попередньої сторінки, а також з нуля розробити як вимоги так і роботу тих елементів, які існували у попередника лише у вигляді кнопок вибору на інтерфейсі і не використовувалися при підсумковому експорті, оскільки дані для них у минулій БД були відсутні. Для цього був використаний досвід використання аналогічних програм, побажання замовника та його фінальних користувачів (клієнтів).

Під час роботи клієнт кілька разів просив змінити проект у зв'язку з:

- Змінами або уточненнями вимог самого замовника;
- Новими вимогами (або уточненнями чи змінами старих вимог), що були надані його клієнтами;
- Додаванням або вирізанням функціоналу, що був признаний зайвим у зв'язку з переходом на нову базу даних.

Так, наприклад, документи стали не багатомовними, а фільтрація на вигляд продукту була прибрана. На жаль, картинки з виглядом 360 ° не були надані клієнтом, тому вони так і залишилися заглушкою.

Фінальний вигляд нової сторінки виводу можна побачити на рис. 4.2. На жаль, матеріалу щодо проміжного вигляду не залишилось, тому у всій роботі буде представлено лише фінальний варіант, що був прийнятий замовником.

Для того, щоб побачити цю таблицю, необхідно авторизуватись у СУБД замовника. Для цього кожен контрагент має власний акаунт.

## Händlerauswahl:

Bauhaus

Bitte nun semikolon-basiert die gewünschten EAN-Codes in das unten folgende Textfeld einfügen:

4018211760855

## Dokumente

Datenblatt: Deutsch

Aufbauanleitung: Deutsch

## Bildtypen

- Ambientebild
- Freisteller
- Detail 1
- Detail 2
- Detail 3
- Detail 4

- Detail 5
- Detail 6
- Detail 7
- Skizze 1
- Skizze 2
- Skizze 3

- Bilder nur einmal exportieren (Articleimages.csv)  
(Ignoriert das Namensschema für alle Bilder)

Namensschema: {EAN}\_{LAN}\_{name}

Platzhalter	Bedeutung
{EAN}	EAN
{LAN}	Artikelnummer
{code}	Merchant-Artikelnummer
{name}	Bezeichnung des Bildes bspw. "Bild_1" oder "Aufbauanleitung"
{original}	Originalname der Datei

## 360°-Bilder

- Im Stammverzeichnis

In zusätzlichem Unterordner mit dem Namen:

- Skizze 4
- Skizze 5
- Skizze 6
- Ansicht Vorne
- Ansicht Links
- Ansicht Hinten

- Ansicht Rechts
- Grundriss
- Schnitt

Alle abwählen

Alle auswählen

Export starten

Рисунок 4.2 – Зовнішній вигляд остаточної версії сторінки.



## 4.2 Детальний опис програми

Для того, щоб зрозуміти як працює додаток для видобутку інформації, необхідно розібратись які в нього присутні елементи. Отже, в даній роботі буде описано меню даного додатку послідовно та згідно з його складовими частинами.

Весь додаток можна умовно поділити на шість складових частин. Всі вони за своєю суттю є фільтрами для пошуку інформації у таблиці продукт та пов'язаних з нею таблицях. Усі картинки та назви як в даній роботі, так і у самому додатку на німецькій мові, оскільки замовник потребує лише її для роботи.

- Вибір контрагента (Händlerauswahl) (Рис.4.3).

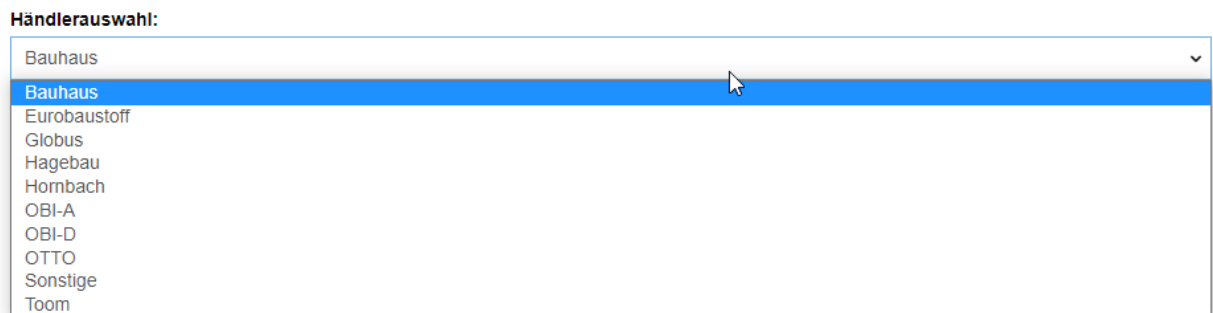


Рисунок 4.3 – Зовнішній вигляд меню вибору контрагента

Це найперше і найважливіше меню вибору в даному додатку. Воно підвантажує список схвалених контрагентів із бази даних, після чого фільтрує інформацію про товар або товари, що необхідна конкретно обраному контрагенту.

- Вибір коду товару (EAN) (Рис.4.4).

**Händlerauswahl:**

Bauhaus

**Bitte nun semikolon-basiert die gewünschten EAN-Codes in das unten folgende Textfeld einfügen:**

4018211760855

Рисунок 4.4 - Зовнішній вигляд меню вибору коду товару

EAN - це унікальний код товару. Він відомий як виробнику, і контрагенту. Саме він є ключовим у виборі товарів як на старій, так і на новій сторінці.

Розділювачем для EAN на старій сторінці та на початкових етапах нової було»; »(EAN1; EAN2; EAN3), проте не всім контрагентам це було зручно. За словами клієнта, багато хто з них просто копіював номери з документа або блокнота, тому в результаті пропозицій контрагентів було вирішено додати роздільник у вигляді абзацу. На даний момент обидва варіанти роздільників доступні.

- Вибір імені медіа файлів (Namensschema) (рис.4.5).

Bilder nur einmal exportieren (Articleimages.csv)  
(Ignoriert das Namensschema für alle Bilder)

**Namensschema:** {EAN}\_{LAN}\_{name}

Platzhalter	Bedeutung
{EAN}	EAN
{LAN}	Artikelnummer
{code}	Merchant-Artikelnummer
{name}	Bezeichnung des Bildes bspw. "Bild_1" oder "Aufbauanleitung"
{original}	Originalname der Datei

Рисунок 4.5 – Зовнішній вигляд меню імені медіа файлів

Контрагент може побажати назвати медіа, що експортуються, по-своєму, тому сторінка дає йому такий вибір. Оскільки одна медіа (наприклад інструкція зі складання) може бути релевантною для кількох схожих товарів

(наприклад шафа бежевого або чорного кольору цілком можуть мати одну і ту ж інструкцію), то контрагенту надано можливість враховувати коди товару при виборі імені медіа файлу.

Таблиця під блоком вибору імені надає 5 кодових виразів, які можуть з'явитися в імені медіа файлу. Все інше буде передано один-на-один. Розширення медіа файлу буде таким самим, як у базі даних. Код за замовченням можна спостерігати на картинці. Він заповнюється так навіть якщо оновити сторінку (оскільки більшість контрагентів взагалі не використовують цю функцію). Однак, якщо контрагент не бажає створювати безліч дублікатів, він може вибрати експортувати медіа лише раз (Bilder nur einmal exportieren (ArticleImages.csv)). У такому випадку схема ігнорується і застосовується лише код оригінального імені медіа файлу, що міститься в базі даних (original).

Звичайно ж, в обох випадках, в окремому файлі буде записано який медіа-файл (з його поточним ім'ям при експорті) прив'язаний до кожного вибраного продукту.

– Вибір ПДФ файлу (рис.4.6).



Рисунок 4.6 – Зовнішній вигляд меню вибору ПДФ файлу

У виробника є два види ПДФ файлів, прив'язаних до товару – рекламна брошура (Datenblatt) та інструкція зі збирання (Aufbauanleitung). Рекламна брошура має варіації кількома мовами, а інструкція зі збирання одномовна.

Побудова імен обох цих видів ПДФ файлів підпорядковується вибору імені медіа файлів, проте рекламна брошура завжди є унікальною для кожного товару.

Якщо контрагент не бажає експортувати будь-який з цих файлів, він завжди може вибрати опцію «не генерувати». Ця опція є опцією за промовчанням.

- Зображення з видом 360° (рис.4.7).

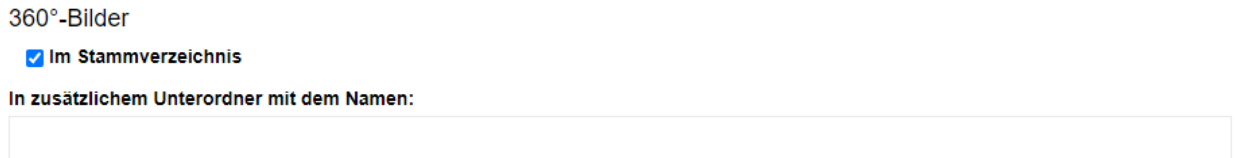


Рисунок 4.7 – Зовнішній вигляд меню вибору зображення з видом 360°

Хоча ця опція технічно і є «заглушкою», проте це лише тому, що через відсутність таких картинок у базі даних фільтрація завжди видає порожній результат. Якщо клієнт все ж таки вирішить додати такі зображення як було зазначено дана опція буде видавати всі необхідні результати. Ще одна технічно реалізована, але не працююча з-за відсутності даних особливість цієї опції - це можливість винесення всіх таких картинок в окрему суб-папку з ім'ям, що вирішить контрагент. Дане ім'я задається у меню «У додатковій вкладеній папці з назвою:» (In zusätzlichem Unterordner mit dem Namen:). Якщо назва не задана, то і винесення не відбудеться.

Побудова імен картинок з видом 360° також підпорядковується вибору імені медіа файлів.

- Всі інші картинки (рис.4.8).

Усі картинки в базі даних мають задану роль (Bildtypen). Саме вона і є фільтром, яким контрагент вибирає, чи експортуватиме її чи ні. Для вибору достатньо «проклікати» необхідні галочки. Стандартний функціонал з опціями «вибрати все» та «скинути все» також присутній.

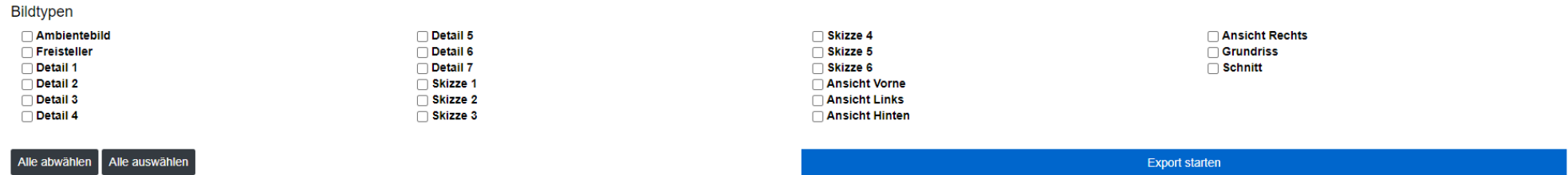


Рисунок 4.8 – Зовнішній вигляд меню вибору всіх інших видів картинок

		Папка с файлами			
..					
ArticleAddons.csv	215	125	Файл Microsoft Of...	29.05.2023 9:26	B9F23093
ArticleImages.csv	637	148	Файл Microsoft Of...	29.05.2023 9:26	9EA10A6A
28207487_Grundriss_206117-01-31.png	5 398	5 117	Файл "PNG"	03.02.2023 17:01	3DE8A853
28207487_Ansicht Vorne_206117-01-31.png	7 280	6 911	Файл "PNG"	03.02.2023 17:01	81769F97
28207487_Schnitt_206117-01-31.png	8 486	8 188	Файл "PNG"	03.02.2023 17:01	3B88706B
28207487_Skizze 1_206117-01-31.jpg	138 196	55 373	Файл "JPG"	03.02.2023 17:01	C0231A2A
28207487_Skizze 2_206117-01-31.jpg	140 476	65 160	Файл "JPG"	03.02.2023 17:01	467F7039
28207487_Freisteller_206117-01-31.jpg	145 796	116 033	Файл "JPG"	03.02.2023 17:01	4CA119B0
28207487_Skizze 3_206117-01-31.jpg	149 300	49 461	Файл "JPG"	03.02.2023 17:01	C47C9C1B

Рисунок 4.9 – Зовнішній вигляд архіву, що був утворений в результаті експорту

Побудова імен зображень підпорядковується вибору імені медіа файлів.

В результаті експорту створюється архів і після цього він на стандартний спосіб завантажується через меню браузеру. Зміст архіву видно на зображенні вище (рис.4.9).

В архіві присутні дві таблиці. Це опис продукту та його асоціатів (ArticleAddons.csv) (рис.4.10) (продуктів що, наприклад, часто продаються разом) заданого для вибраного контрагента, а також таблиця всіх вибраних продуктів та прив'язаних до них медіа файлів (ArticleImages.csv) (рис.4.11).

В таблиці, що описує продукт та його асоціати (ArticleAddons.csv) вказуються коди (EAN, LAN та KundenArtNr) та назва (Titel) продукту та кожного з продуктів, що асоційовані з ним (за аналогією, тільки додано префікс «AddOn-»).

	A	B	C	D	E	F	G	H
1	EAN	LAN	KundenAr	Titel	AddOn-EAN	AddOn-LANs	AddOn-KundenArtNr	Addon-Titel
2	4,02E+12	371808-00	28203847	Pavillon Orleans 1, Leimholz, natur	'''	'''	'''	'''
3	4,02E+12	371808-01	28203856	Pavillon Orleans 1, Leimholz, eiche hell	'''	'''	'''	'''
4	4,02E+12	371808-02	28203855	Pavillon Orleans 1, Leimholz, nussbaum	'''	'''	'''	'''

Рисунок 4.10 - Зовнішній вигляд таблиці, що описує продукт та його асоціатів (ArticleAddons.csv)

	A	B	C	D	E	F	G
1	Artikelnummer	EAN	LAN	Bildtyp	Dateiname		
2	371808-01-31	4E+12	371808-0	Detail 1	28203856_Detail 1_371808-01-31.jpg		
3	371808-01-31	4E+12	371808-0	Schnitt	28203856_Schnitt_371808-01-31.png		
4	371808-01-31	4E+12	371808-0	Skizze 3	28203856_Skizze 3_371808-01-31.jpg		
5	371808-01-31	4E+12	371808-0	Grundriss	28203856_Grundriss_371808-01-31.png		
6	371808-01-31	4E+12	371808-0	Ansicht Vorne	28203856_Ansicht Vorne_371808-01-31.png		
7	371808-01-31	4E+12	371808-0	Skizze 1	28203856_Skizze 1_371808-01-31.jpg		
8	371808-01-31	4E+12	371808-0	Skizze 2	28203856_Skizze 2_371808-01-31.jpg		
9	371808-01-31	4E+12	371808-0	Ambientebild	28203856_Ambientebild_371808-01-31.jpg		

Рисунок 4.11 - Зовнішній вигляд таблиці всіх обраних продуктів та прив'язаних до них медіа файлів (ArticleImages.csv)

В таблиці всіх обраних продуктів та прив'язаних до них медіа файлів (ArticleImages.csv) наведено коди (EAN та LAN), а також назви прив'язаних до продукту медіа файлів (Dateiname) згідно з застосованою у меню «вибір

імені медіа файлів (Namensschema)» схемою і обраними ролями. Також в таблиці показано тип (роль) картинки (Bildtyp). Це ті ж самі типи картинок, що зазначені в пункті «всі інші картинки».

Якщо за вибраними фільтрами не було знайдено медіа файлів для продукту, він виключається з таблиці всіх обраних продуктів і прив'язаних до них медіа файлів (ArticleImages.csv). А якщо медіа файлів взагалі немає у вибірці, то таблиця всіх обраних продуктів та прив'язаних до них медіа файлів (ArticleImages.csv) не створюється. Таблиця, що описує продукт та його асоціати (ArticleAddons.csv) виводиться в будь-якому разі. В разі відсутності будь-якого продукту в обраних, експорт не відбувається.

## ВИСНОВКИ

У сучасному цифровізованому світі інтегровані системи відіграють важливу роль не тільки у повсякденному житті або під час продаж але і у виробництві. Підприємства, що застосовують дані інновації, отримують в результаті значну кількість конкурентних переваг.

Одним з таких підприємств є німецька фірма Skanholz. Вона виробляє різноманітні товари з дерева. У перелік товарів входять такі вироби як двері, вікна, бесідки, меблі та інше. Саме великий об'єм даних, пов'язаний з роботою з різноманітними товарами та багатьма клієнтами вимусив дану фірму вступити на шлях використання інтегрованої системи для оптимізації управління будівельними матеріалами. Даний крок забезпечить підприємству єдине просторове сховище даних для управління проектними задачами, даними та ресурсами.

У роботі був проведений аналіз існуючих моделей репрезентації частково структурованих даних у базах даних, зокрема, ієрархічної, мережевої, інвертованої файлової, реляційної, пост-реляційної, графової, багатозначної, об'єктно-орієнтованої тощо.

В результаті роботи була розроблена та запропонована для використання підприємством Skanholz інтегрована система для оптимізації управління будівельними матеріалами. Систему створено за технологією ASP.NET, що допомагає реалізує всі побажання замовника. Даний метод побудови інтегрованої системи базується на використанні сучасних інформаційних технологій та має основні функції: ведення обліку будівельних матеріалів, планування закупівель будівельних матеріалів, контроль використання будівельних матеріалів.

Пропонована система дозволяє будівельним компаніям оптимізувати витрати на закупівлю будівельних матеріалів, покращити контроль за



використанням будівельних матеріалів, знизити ризик втрат будівельних матеріалів.

База даних системи містить інформацію про всі будівельні матеріали, які використовуються на об'єкті. Веб-інтерфейс дозволяє користувачам вводити та отримувати інформацію з бази даних.

Пропонована інтегрована система для оптимізації управління будівельними матеріалами має ряд переваг:

- використання сучасних інформаційних технологій дозволяє оптимізувати витрати на закупівлю будівельних матеріалів, покращити контроль за використанням будівельних матеріалів та знизити ризик втрат будівельних матеріалів.
- система є інтегрованою, що дозволяє користувачам отримувати всю необхідну інформацію в одному місці.
- система є гнучкою і може бути налаштована відповідно до потреб конкретного об'єкта.

Пропонована система може бути використана комунальними підприємствами та будівельними компаніями.

На даний момент дана інтегрована система використовується підприємством у їхній повсякденній роботі.

Таким чином, усі поставлені завдання даної магістерської роботи виконані, а поставлена мета - досягнута. Використання запропонованої системи структуризації даних підвищує продуктивність управління підприємством.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Two Examples of Database Structures in Management of Engineering Data Neven Pavković та Mario Štorga University of Zagreb. URL: [https://www.researchgate.net/publication/268270067\\_Two\\_Examples\\_of\\_Database\\_Structures\\_in\\_Management\\_of\\_Engineering\\_Data](https://www.researchgate.net/publication/268270067_Two_Examples_of_Database_Structures_in_Management_of_Engineering_Data) (дата звернення 10.11.2023).
2. Production database. URL: <https://desktop.arcgis.com/ru/arcmap/latest/extensions/production-mapping/production-database.html> (дата звернення 10.11.2023).
3. What Do Product Managers Need To Know About Database Design? URL: <https://chisellabs.com/blog/what-do-product-managers-need-to-know-about-database-design/> (дата звернення 11.11.2023).
4. Data Management for Manufacturing: from databases to Digital Twins database management system (DBMS) Craig S. Mullins, Mullins Consulting. URL: <https://www.techtarget.com/searchdatamanagement/definition/database-management-system> (дата звернення 11.11.2023).
5. Database Management Tools (With Features and Benefits) Updated June 25, 2022. URL: <https://www.indeed.com/career-advice/career-development/database-management-tools> (дата звернення 11.11.2023).
6. Database management tools. URL: <https://www.plutora.com/ci-cd-tools/database-management-tools> (дата звернення 12.11.2023).
7. SQL Database. URL: <https://www.studysmarter.co.uk/explanations/computer-science/databases/sql-database/> (дата звернення 12.11.2023).
8. What Is SQL Used For? 7 Top Uses by Enterprise DNA Experts | 3:10 pm EDT | April 10, 2023. SQL. URL: <https://blog.enterprisedna.co/what-is-sql-used-for/> (дата звернення 12.11.2023).

9. What is Data Extraction: Techniques, Tools, and Best Practices by Astera Analytics Team Updated on: September 27th, 2023. URL: <https://www.astera.com/type/blog/what-is-data-extraction-a-brief-guide/> (дата звернення 13.11.2023)

10. What is Data Extraction? Definition and Examples. URL: <https://ua.talend.com/resources/data-extraction-defined/> (дата звернення 13.11.2023).

11. Daniel R. M., M. G. Kenward, S. N. Cousens, and B. L. De Stavola Using causal diagrams to guide analysis in missing data problems. *Statistical methods in medical research*. 2012 p.

12. Darwiche A Modeling and reasoning with Bayesian networks. Cambridge University Press. Cambridge, 2009.

13. GUTIERREZ RENZO ANGLES and CLAUDIO Survey of Graph Database Models [Article] // *ACM Computing Surveys*, Vol. 40, No. 1, Article 1. February 2008. 1.

14. Haitovsky Y. Missing data in regression analysis. [s.l.]: Journal of the Royal Statistical Society. Series B (Methodological), 1968. pp. 67-82.

15. Koller D. and N. Friedman Probabilistic graphical models: principles and techniques. 2009.

16. Little R. and D. Rubin Statistical analysis with missing data. [s.l.]: Wiley, 2002.

17. Scheffer J. Dealing with missing data. *Research Letters in the Information and Mathematical Sciences*. 2002 p. cc. 153-160.

18. Thoemmes F. and K. Mohan Graphical representation of missing data problems. *Structural Equation Modeling: A Multidisciplinary Journal*. 2015.

19. Thoemmes F. and N. Rose Selection of auxiliary variables in missing data problems: Not all auxiliary variables are created equal. Technical Report R-002: Cornell University, 2013.

20. Chazelle B. The discrepancy method: randomness and complexity [Книга]. Cambridge: Cambridge University Press, 2000.