**Міністерство освіти і науки україни**

**Національний авіаційний університет**

**Кафедра комп’ютеризованих систем управління**

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_ Литвиненко О.Є.

“\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 р.

**ДИПЛОМНА РОБОТА**

**(пояснювальна записка)**

**випускника освітньоГО СТУПЕНЯ**

**“магістр”**

**Тема:** Система збереження та швидкого доступу до електронних документів

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Виконавець:**                                                                           Марола О. В.

**Керівник:**                                                                               Росінська Г. П.

**Нормоконтролер:**Тупота Є. В.

**Київ 2020**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

Навчально-науковий інститут Комп’ютерних інформаційних технологій\_\_\_\_\_\_\_\_\_\_

Кафедра комп’ютеризованих систем управління\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність (спеціалізація) 6.050102 «Комп’ютера інженерія». (шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_ Литвиненко О.Є.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2020 р.

**ЗАВДАННЯ**

**на виконання дипломної роботи**

Мароли Олеся Віталійовича      

(прізвище, ім’я, по батькові випускника в родовому відмінку)

1.Тема дипломної роботи «Система збереження та швидкого доступу до\_\_\_\_\_\_\_\_\_\_\_ електронних документів»

затверджена наказом ректора від «27» серпня 2019 р. № 1808/ст

2.Термін виконання роботи: з 14 жовтня 2019 року по 9 лютого 2020 року

3.Вихідні дані до роботи: Дослідити способи організації процесу збереження корпоративних документів. Дослідити існуючі програмні продукти та сформувати вимоги до власного програмного продукту.

4.Зміст пояснювальної записки: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) Аналіз проблематики області дослідження

2) Дослідження технологій створення файлових сховищ

3) Проектування програмної системи

5.Перелік обов’язкового графічного (ілюстративного) матеріалу:

1) Приклад структури хмарного сховища

2) Загальна архітектура системи «*Book Store*»

3) Приклад роботи програми «*Book Store*»

6.Календарний план-графік

|  |  |  |  |
| --- | --- | --- | --- |
| №  пор. | Завдання | Термін  виконання | Відмітка про виконання |
| 1 | Ознайомлення з постановкою завдання дипломної роботи | 14.10.2019 – 16.10.19 |  |
| 2 | Аналіз літературних джерел та інтернет-ресурсів | 17.10.19 – 18.10.19 |  |
| 3 | Дослідження та аналіз запитів користувача на веб-сайтах | 19.10.19 – 16.11.19 |  |
| 4 | Проектування архітектури модуля та інтерфейсу користувача | 17.11.19 – 20.11.19 |  |
| 5 | Розробка програмного модуля | 21.11.18 – 12.01.19 |  |
| 6 | Підготовка пояснювальної записки | 13.01.19 – 01.02.19 |  |
| 7 | Підготовка графічного матеріалу | 02.02.19 –  03.02.19 |  |

7.Дата видачі завдання:  14 жовтня 2019 р.

Керівник дипломної роботи       доц. Росінська Г. П.

(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання  Марола О. В.

(підпис випускника) (П.І.Б.)

**РЕФЕРАТ**

Пояснювальна записка до дипломної роботи «Система збереження та швидкого доступу до електронних документів»: 82с.*,* 1 таблиця, 32 рис.*,* 31 літературне джерело, 1 додаток.

ДОСТУП ДО ДАНИХ, ІНФОРМАЦІЙНИЙ ПОШУК, СИСТЕМА ЗБЕРЕЖЕННЯ ДОКУМЕНТІВ, ПОШУКОВИЙ ЗАПИТ, АЛГОРИТМ ЗБЕРЕЖЕННЯ.

Об’єкт дослідження - автоматизовані засоби збереження електроних документів з використанням баз документів та комплекс засобів швидкого доступу до електронних документів.

Предмет дослідження - система збереження та швидкого доступу до електронних документів.

Метою дипломної роботи є розробити систему збереження документів користувачів для швидкого та зручного отримання доступу до електронних документів.

Методи дослідження - алгоритми збереження, методи пошуку даних у базі документів, методи об’єктно-орієнтованого програмування, технології створення та функціонування систем збереження документів, теорія баз документів.

Здійснено аналіз існуючих систем збереження документів, які реалізують подібні функції та автоматизують бізнес-процеси, здійснено їх порівняльну характеристику, виявлено переваги та недоліки. Досліджено різноманітні алгоритми та методи збереження даних. Створено та проведено тестування системи збереження та швидкого доступу до електронних документів. Шляхом використання різних видів електронних документів було отримано резульати, які задовольніють потреби користувачів.

Матеріали дипломної роботи рекомендується використовувати у навчальних та комерційних цілях, для малих підпиємств, тощо.

**ЗМІСТ**

[ВСТУП 6](#_Toc30523650)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ ОБЛАСТІ ДОСЛІДЖЕННЯ 10](#_Toc30523651)

[1.1. Особливості роботи з даними 10](#_Toc30523652)

[1.2. Способи збереження документів 12](#_Toc30523653)

[1.3. Огляд існуючих програмних рішень 18](#_Toc30523658)

[1.4. Бази документів для збереження даних локально (на сервері) 21](#_Toc30523662)

[1.5. Коротка характеристика деяких СУБД 30](#_Toc30523665)

[1.6. Порівняння найактуальніших серверних СУБД 35](#_Toc30523669)

[1.7. Висновки до розділу 37](#_Toc30523670)

[РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ СТВОРЕННЯ ФАЙЛОВИХ  
СХОВИЩ 39](#_Toc30523671)

[2.1 Технології організації сховищ даних 39](#_Toc30523672)

[2.2 Типи моделей даних сховищ 52](#_Toc30523678)

[2.3 Використання метаданих 57](#_Toc30523682)

[2.4. Висновки до розділу 59](#_Toc30523683)

[РОЗДІЛ 3 ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ 61](#_Toc30523684)

[3.1 Проектування архітектури 61](#_Toc30523685)

[3.2. Опис *API* 62](#_Toc30523686)

[3.3. Середовище розробки програмного продукту 62](#_Toc30523687)

[3.4. Робота з базою документів 64](#_Toc30523688)

[3.5. Головне вікно програми 67](#_Toc30523689)

[3.6. Перегляд документів 68](#_Toc30523690)

[3.7. Додавання нового документа до бази документів 70](#_Toc30523691)

[3.8. Видалення документа з бази документів 72](#_Toc30523692)

[3.9. Організація швидкого доступу до документів 74](#_Toc30523693)

[3.10. Висновки до розділу 76](#_Toc30523694)

[ВИСНОВКИ 77](#_Toc30523695)

[СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ 79](#_Toc30523696)

[ДОДАТОК А 82](#_Toc30523697)

# **ВСТУП**

Актуальність теми. Практика збереження документів у сховищах широко поширилася за останні дев’ять років. Підприємства переходять від традиційного ПК з декількома внутрішніми жорсткими дисками до ноутбуків з невеликими *SSD* накопичувачами, вважаючи віддалене сховище актуальним. Збільшення використання смартфонів змінює вимоги способу організації доступу до корпоративних документів. Працівники прагнуть отримати доступ до документів в будь-який час і з різних пристроїв. Вони не хочуть бути залежними від *Internet* з’єднання, об’єму доступного простору чи навіть від працездатності зовнішньої системи.

Більшість співробітників підприємств зберігають корпоративні дані на своїх персональних комп’ютерах, або використовують сторонні сервіси. У першому випадку подібна практика може привести до втрати документів при апаратному/програмному збої, в іншому дані можуть передаватися по незахищених каналах зв’язку, що може привести до втрати важливих документів. Крім того дані зберігаються в неструктурованому вигляді і розподілені по різних пристроях або комп’ютерах, що ускладнює доступ до них.

Вирішення цих проблем дозволить підприємствам покращити контроль над даними і надати користувачам зручний інструмент для збереження і спільного доступу до документів. Всі наявні програмні продукти є дорогими або не володіють достатньою функціональністю для усунення усіх цих проблем.

Переваги електронного документообігу:

* економія часу: службовці витрачають менше часу на пошук паперових документів. Завдяки центральній базі документів, регулярно створюються резервні копії файлів, завдяки чому виключається можливість того, що документ буде безповоротно втрачено, якщо його забудуть в літаку, випадково або навмисно знищать або ж просто згине в офісному безладі. Абсолютно виключається втрата часу на пошуки файлів і документів, яких, з якоїсь причини, не виявилося на своєму місці;
* більш адекватне використання фізичного простору і техніки: цінні квадратні метри, зайняті зайвими серверами і іншими пристроями для збереження документів можуть бути звільнені. Залежно від статусу і актуальності даних, документи та файли можуть безпечно віддалятися після закінчення терміну їх збереження. Управління даними не тільки допомагає відповідати корпоративним нормам, а й сприяє більш адекватному використанню місця для збереження;
* підвищення прозорості внутрішньої роботи підприємства: СЕД (системи електронного документообігу) дозволяють керівникам спостерігати за статусом документа, протягом усіх етапів його погодження та затвердження. На додаток до цього, СЕД дозволяє моментально і легко викликати не тільки запитуваний файл, але також і повний звіт про те, хто його створив, хто мав до нього доступ і хто його редагував;
* ведення особистої історії кожного файлу і супутньої документації: СЕД дозволяють централізовано керувати взаємовідносинами з клієнтами та постачальниками. Наприклад, досить лише одного кліка, щоб викликати всі необхідні документи, які містять вимоги, пов’язані з різними типами взаємин між організацією і зовнішніми суб’єктами;
* більше гнучкості щодо фізичного місцезнаходження співробітників: завдяки можливостям електронного доступу і комунікацій, службовці отримують можливість працювати віддалено. І навіть перебуваючи в одному і тому ж географічному місці, службовцям більше не буде потрібно чекати, поки паперові копії файлів будуть пересилатися з сусіднього офісу;
* підвищення безпеки даних та документів: як уже згадувалося, центральна база документів дозволяє робити резервні копії документів, завдяки чому знижується ризик випадкової або навмисної втрати файлів. При цьому, менше часу витрачається на пошуки необхідного документа, якщо його місцезнаходження з якоїсь причини змінилося;
* зниження витрат на роздруківку, поштові марки, конверти та пересилання: паперові документи, які пересилаються між відділами або постачальниками, можуть пересилатися в електронному вигляді;
* підвищення рівня задоволеності службовців і керівників: оптимізація щоденних завдань дозволяє співробітникам отримувати більше задоволення від робочого процесу. Звільнення співробітників від таких, часто нудних завдань, як обробка накладних, дозволяє їм присвятити себе іншій діяльності. У той же час, керівники відділів отримують більше можливостей контролювати роботу своїх підлеглих. В остаточному підсумку, деякі організації можуть виявити, що зекономлені кошти дозволяють їм вийти на новий бізнес-рівень.

Метою дипломної роботи є розробка системи збереження документів користувачів для швидкого та зручного отримання доступу до електронних документів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Проаналізувати існуючі системи збереження документів, здійснити їх порівняльний аналіз, визначити переваги та недоліки;
2. Дослідити структуру систем збереження та швидкого доступу до документів;
3. Проаналізувати сучасні алгоритми збереження та методи швидкого доступу до даних;
4. Дослідити існуючі механізми інтеграції запитів пошукових систем;
5. Створити систему для збегігання і швидкого доступу до електронних документів*.*

Об’єкт дослідження - автоматизовані засоби збереження електроних документів з використанням баз документів та комплекс засобів швидкого доступу до електронних документів.

Предмет дослідження - система збереження та швидкого доступу до електронних документів.

У дипломній роботі було використано такі методи дослідження:

* методи збереження документів - використовувалися при дослідженні процесу збереження електронних документів;
* методи пошуку у базах документів - використовується для обробки пошукових запитів та формування ключових слів;
* технології створення та функціонування систем збереження документів – при дослідженні систем та методів збереження електронних документів, вилучення необхідних документів з пошукових систем для інтеграції їх до системи збереження та швидкого доступу до електронних документів;
* методи об’єктно-орієнтованого програмування - використовувалися при розробці системи збереження та швидкого доступу до електронних документів;
* методи *UI/UX* - при створенні інтерфейсу користувача;
* теорія баз документів - при роботі з базами документів.

Практичне значення отриманих результатів. Матеріали дипломної роботи рекомендується використовувати у навчальному процесі для галузей, де необхідно зберігати електронні документи та швидко отримувати доступ до них. Використання створеної системи збереження та швидкого доступу до електронних документів дозволяє забезбечити конфіденційність документів, організувати документообіг та прискорити пошук необхідних документів.

Особистий внесок випускника. Особисто було здійснено аналіз існуючих систем збереження даних, які реалізують подібні функції та автоматизують бізнес-процеси, здійснено їх порівняльну характеристику, виявлено переваги та недоліки. Досліджено різноманітні алгоритми та методи пошуку. Створено та проведено тестування системи збереження та швидкого доступу до електронних документів. Шляхом використання різних видів електронних документів було отримано резульати, які задовольніють потреби користувачів.

**РОЗДІЛ 1****аналіз проблематики області дослідження**

## 1.1. Особливості роботи з даними

Для підприємств будь-якого типу і розміру, розвиток робочих процесів висуває вимоги, які кидають виклик звичайним методам обміну файлами. Якщо співробітники не мають доступу до системи збереження документів з можливістю інтенсивного обміну інформацією, використовуючи будь-які пристрої, інформація буде передаватися по неконтрольованим і потенційно небезпечним каналам зв’язку. Співробітники потребують ефективного і зручного способу організації спільної роботи з документами, і організації, які не реалізують цей функціонал внутрішньо будуть більш частіше виявляти, що конфіденційні дані передаються по каналах безконтрольно - в той час, коли методи викрадення документів досягли великого рівня складності.

Реалізуючи мобільні системи збереження та спільного доступу до документів, які використовують моделі розгортання на місці, компанії можуть зберегти конфіденційну інформацію під наглядом без зменшення продуктивності праці співробітників, ефективності роботи і конкурентної переваги.

З розповсюдженням публічних хмарних рішень, поширення даних стало легшим, ніж коли-небудь. І працівники почали користуватись такими сервісами для збереження корпоративних документів.

Збільшення використання мобільних пристроїв міняє вимоги способу організації доступу до електронних документів. Працівники прагнуть отримати доступ до документів в будь-який час і з різних пристроїв. Вони не хочуть бути залежними від *Internet* з’єднання, об’єму доступного простору чи навіть від працездатності хмарної системи.

Синхронізація та поширення документів на основі хмарних сервісів стають більш популярнішими. Ці послуги є простими для використання, але рівень контролю та безпеки документів компанії є неоднозначним.

Експоненціальне зростання документів з *Internet*, мобільних пристроїв, цифрових бізнес-процесів, а також «Інтернету речей», надає підприємствам безцінний потенціал активів, які можуть бути використані для досягнення конкурентної вигоди.

Без правильних засобів і стратегій, можливості цих активів, як правило, залишаються незавершеними. Більш того, неконтрольоване збільшення кількості документів може збільшувати вартість управління процесами збереження і резервного копіювання документів.

Крім того, зростаюча популярність хмарних схових означає, що документи компанії можуть бути розподілені географічно і бути розміщеними в різних місцях, включаючи фізичні сервери, віртуальні машини, персональні комп’ютери, мобільні пристрої, тощо.

Всі ці документи повинні надійно зберігатися і мати можливість бути відновленими з резервних копій в будь-який момент часу, щоб запобігти втрат від перебоїв в подачі електроенергії, стихійних небезпек, неправомірних дій і людських помилок. Неможливість надійно масштабувати сховища документів може створити перешкоди для орієнтованого на зростання компаній [1].

Замість того, щоб ефективно використовувати документи в якості конкурентного активу, компанії, які не в змозі управляти і захищати їх документи ризикують:

1. Дозволити корпоративній та клієнтській даних бути загубленою або вкраденою;
2. Витрачати велику кількість ресурсів на збереження документів;
3. Втратити конкурентоспроможність компанії в порівнянні з тими, хто використовує більш гнучкі і масштабовані системи.

Працівники мають потребу у засобі збереження і обміну документами, який буде:

1. Зберігати великі обсяги документів;
2. Мінімізувати втрати документів через апаратні сбої;
3. Давати можливість доступу з пристроїв різного типу через зручний універсальний *interface*;
4. Забезпечувати доступність документів у будь який час без затримки;
5. Надавати механізми контролю над потоками документів в межах компанії.

## 1.2. Способи збереження документів

В інформаційному суспільстві основним ресурсом є інформація, саме на основі володіння інформацією про якісь процеси і явища можна ефективно й оптимально будувати будь-яку справу. Більша частина населення в інформаційному суспільстві зайнята в сфері обробки даних або використовує інформаційні й комунікаційні технології у своїй кожнодневній виробничій діяльності.

Під інформацією розуміють корисний зміст явища факту, вилученого з документів. Тому на практиці дуже часто неправильно встановлюють знак рівності між даними та інформацією. Дані реєструють явище і події, що відбуваються, а інформація - результат переробки та аналізу цих документів. Інформація переважно носить порівняльний характер. Вона складається тільки з нових відомостей, які оцінюються користувачем як корисні знання. З загального переліку вилучаються тільки відомості, які потрібні для визначеного користувача.

Сьогодні інформаційні технології почали активно впливати на повсякденну діяльність будь-якої компанії і стали невід’ємною складовою частиною інформаційної інфраструктури цієї компанії. Інформаційні технології дозволяють досліджувати та зв’язувати складові ділянки діяльності компанії між собою. Вони розвиваються надзвичайно швидкими темпами і захоплюють все ширші сфери діяльності, таким чином, що будь-яка конкурентоспроможна активність в майбутньому не може бути сформована без детального аналізу можливостей застосування інформаційних технологій [2].

Матеріальним фундаментом сучасних інформаційних технологій є три основних технічні досягнення:

* поява нового середовища нагромадження даних на фізичних носіях (магнітні стрічки, мікрофільми, магнітні та оптичні диски, напівпровідникові пристрої, тощо);
* розвиток засобів зв’язку, що забезпечують доставку даних практично в будь-яку точку земної кулі без обмежень в часі та просторі, та масове охоплення населення засобами зв’язку (радіо, телебачення, мережі передачі документів, супутниковий зв’язок, телефонна мережа);
* можливість автоматизованої обробки документів за допомогою комп’ютера згідно заданого алгоритму.

В основу нової сучасної інформаційної технології, яка базується на широкому застосуванні персональної комп’ютерної техніки, покладені три основні принципи: інтегрованість, гнучкість, інтерактивність. Інтегрованість забезпечує комплексний підхід до розв’язання поставлених завдань за рахунок об’єднання можливостей, як технічних, так і програмних засобів. Гнучкість дозволяє динамічно і невеликими зусиллями виконувати поставлені завдання. Інтерактивність в свою чергу дозволяє динамічно переналагоджувати завдання з врахуванням конкретних вимог.

Особливе місце в організації нових інформаційних технологій займає комп’ютер, який створює широкі можливості для нагромадження необхідної даних (запис в пам’ять рефератів книг, статей, доповідей, результатів досліджень), забезпечення аналітичної обробки великих масивів документів, пересилання даних та її збереження в електронному вигляді, тощо.

Прикладами нових інформаційних технологій можуть бути система електронної пошти, системи факсимільної передачі зображення і мережі передачі документів. Електронна пошта вже стала загальноприйнятим видом інформаційного обслуговування, здатним суттєво потіснити традиційну пошту.

Іншим прикладом стали бази документів, які також зобов’язані своїм створенням комп’ютеру, здатному зберігати у цифровій формі значні обсяги даних. Комп’ютер з допомогою відповідного програмного забезпечення дозволяє оперувати необхідною інформацією, яка знаходиться в його пам’яті, подавати її в потрібній формі та послідовності.

Інформаційні технології проникають, як в професійну діяльність, так і в приватне життя. Вони впливають на міжнародний торговий баланс, рівень зайнятості, політику. В міру становлення інформаційного суспільства виникає потужна індустрія інформаційних технологій, призначена для задоволення потреб цього суспільства. Індустрія інформаційних технологій породжує нові види інформаційного продукту та засоби його доставки до споживачів. Під інформаційним продуктом слід розуміти різноманітні аспекти знань, відомості, твори мистецтва, інші форми даних та розваг, отримані, як традиційними шляхами, так і за допомогою електронної техніки. Особливо яскраво інформаційні технології проявляються в об’єднанні багато чисельних секторів економіки, таких, як видавнича справа, виробництво офісного обладнання, обчислювальної техніки, систем телезв’язку та побутової техніки, які до недавнього часу хоча й були взаємопов’язані, проте незначно. Інформаційні технології стимулюють розвиток та посилення цього зв’язку.

В цілому сучасні інформаційні технології спрямовані на підвищення рівня автоматизації усіх інформаційних процесів, що є основою для прискорення темпів науково-технічного прогресу. Інформаційні технології дозволяють ефективно поєднувати технічні можливості обчислювальної техніки, електрозв’язку, інформатики. Вони спрямовуються на збір, нагромадження, аналіз та доставку даних споживачам (вченим, інженерам, керівникам, лікарям, економістам) незалежно від відстані до джерел та обсягів, на автоматизацію одноманітних операцій управлінського процесу і підготовку аналітичної даних для прийняття рішень [3].

Для забезпечення роботи інформаційної системи будь-якого призначення необхідно:

* виявити інформаційні потреби;
* здійснити відбір джерел даних;
* здійснити збір даних;
* здійснити введення даних із зовнішніх або внутрішніх джерел;
* виконати дії по обробці даних, оцінці її повноти і значущості і за поданням її в зручному вигляді;
* вивести інформацію для надання споживачам або передачі в іншу систему;
* організувати використання даних для оцінки тенденцій, розробки прогнозів, оцінки альтернатив рішень і дій, вироблення стратегії;
* організувати зворотний зв’язок - за інформацією, переробленою людьми даною організацією, здійснювати корекцію вхідної даних.

Підприємства малого бізнесу мають широкий вибір способів збереження і обміну даними. Вони варіюються від портативних накопичувачах флеш-пам’яті до підключаються до мережі систем збереження документів, які можуть бути розташовані фізично в будь-якому місці мережі.

### 1.2.1. Флеш накопичувачі

Цей пристрій є ефективним в тому випадку, коли співробітник потребує зручний засіб збереження невеликого обсягу персональних документів, тому що він споживає мало енергії, має невеликі розміри і не має рухомих частин (рис. 1.1).

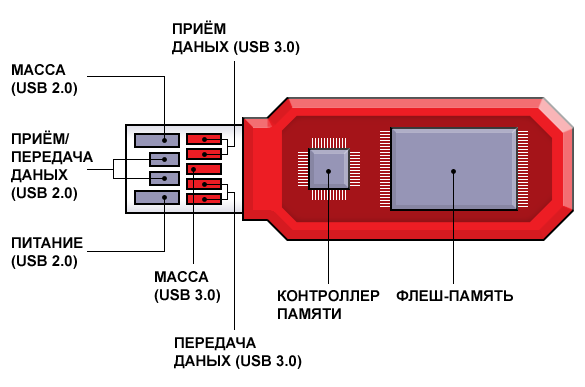


Рис. 1.1. Структура флеш накопичувача

### 1.2.2. Зовнішні жорсткі диски

Підключення зовнішнього жорсткого диску до комп’ютера є простим і відносно недорогим способом додати більше дискового простору (рис. 1.2). Однак зовнішні жорсткі диски прямо підключені до ПК мають ряд недоліків. Серед них:

* документи доступні лише в місці підключення диска;
* низька стійкість жорстких дисків;
* дані не захищені в разі пожежі чи іншої катастрофи на робочому місці.



Рис. 1.2. Структура жорсткого диску

### 1.2.3. Публічне хмарне сховище

Сервіси, які забезпечують віддалене збереження і резервне копіювання через *Internet* пропонують компаніям ряд безперечних переваг. Шляхом резервного копіювання документів на віддаленому сервері, вони дозволяють захиститись від втрат документів, що зберігаються на робочому місці.

Такі сервіси надають можливість обмінюватися великими даними з клієнтами і партнерами, надаючи їм захищений паролем доступ до хмарних сховищ, тим самим прибираючи необхідність відправки цих великих документів по електронній пошті ( див. рис. 1.3). І в більшості випадків, вони надають можливість увійти в обліковий запис користувача з будь-якого пристрою через *web browser* - це відмінний метод для отримання документів, коли користувач знаходиться далеко від робочого місця [4].



Рис. 1.3. Структура хмарного сховища

### 1.2.4. Мережеве сховище

Являє собою комп’ютер з деяким дисковим обладнанням, підключений до мережі (зазвичай локальної) і підтримує роботу з прийнятими в ній протоколами. Диски можуть бути об’єднані в *RAID*-масиви (рис. 1.4). Кілька таких комп’ютерів можуть бути об’єднані в одну загальну систему. Забезпечує надійність збереження даних, легкість доступу для багатьох користувачів, легкість адміністрування, масштабованість.

Дані, що міститься у вигляді повідомлень електронної пошти, документів, презентацій, баз документів, графіки, аудіо-файлів і електронних таблиць є важливою для більшості організацій. Зростаючі потреби зберігати велику к-сть мультимедійних файлів, таких як відео, і зробити їх доступними для користувачів мережі створює попит на більш складні рішення для збереження даних.

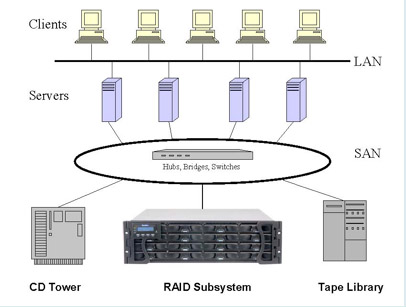


Рис. 1.4. Структура мережевого сховища

## 1.3. Огляд існуючих програмних рішень

### 1.3.1. *OwnCloud*

*ownCloud* - система для організації збереження, синхронізації й обміну інформацією, розміщеною на зовнішніх серверах. Спочатку продукт розвивався спільнотою *KDE*, але згодом засновники продукту створили комерційну компанію *ownCloud* *Inc*, яка взяла в свої руки розробку *ownCloud* й розпочала надання платних сервісів та *Enterprise*-версій платформи (рис. 1.5). Для доступу до документів, збережених в *ownCloud*, можна використовувати веб-інтерфейс або протокол *WebDAV*. Додатково до збереження документів можна відзначити функцію підтримки засобів для забезпечення спільного доступа та можливість синхронізації між різними пристроями таких документів, як адресна книга, календар-планувальник і закладки, з можливістю їх перегляду і редагування з будь-якого пристрою в будь-якій точці мережі. Сервер *ownCloud* можна розгорнути на будь-якому хостингу, який підтримує запуск *PHP*-скриптів і надає доступ до *SQLite*, *MySQL* або *PostgreSQL*.

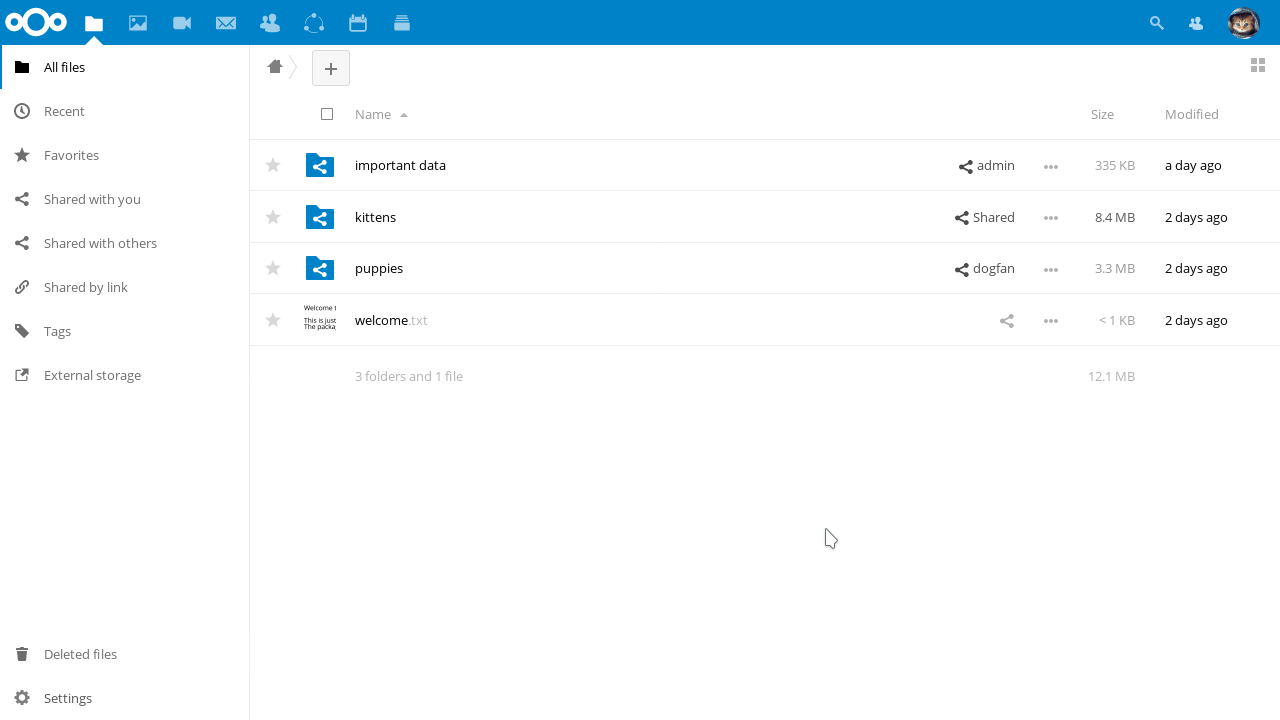


Рис. 1.5. Зовнішній вигляд додатку *OwnCloud*

### 1.3.2. *Git-annex*

*Git-annex* - продукт cинхронизації файлів направлений на розв’язання проблем загального доступу до документів й синхронізації, але не залежить від будь-якої комерційної служби або централізованого сервера (рис. 1.6). Він написаний на *Haskell* і доступний для *Linux*, *Android*, *OS X* і *Windows*. *Git-annex* зберігає тільки посилання на файли в *git* репозиторії і управляє ними в окремому місці. Він також створює копії файлів для відновлення втраченої даних. Слід зазначити, що *git-annex* доступний на різних *Linux* системах, включаючи: *Fedora*, *Ubuntu*, *Debian* і т.д.

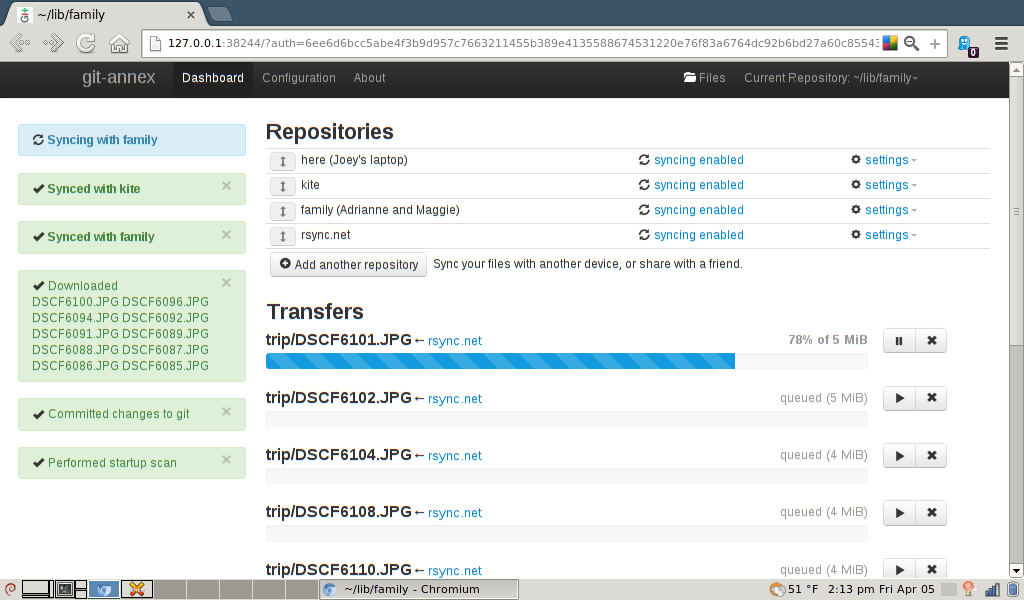


Рис. 1.6. Зовнішній вигляд додатку *git-annex*

### 1.3.3. *Seafile*

*Seafile* - відкритий проект для створення сервісу хмарного збереження документів. Крім базових функцій збереження на віддаленому сервері й забезпечення синхронізації документів між комп’ютерами, *Seafile* надає гнучкі можливості з організації спільної роботи з документами (рис. 1.7). Для зручності спільної роботи підтримується створення робочих областей, в яких члени групи можуть розміщувати будь-яку інформацію, цікаву для учасників групи. Набори файлів можуть об’єднуватися в «бібліотеки», до яких можна відкрити доступ для окремих користувачів або груп, а також загальний доступ.

Кожна бібліотека в сховищі представлена у формі, що нагадує *Git*-репозиторій. Ця особливість надає можливість використання версійного контролю, у тому числі підтримку доступа до минулих редакцій збереженого контенту, можливість відстежити всі внесені зміни (хто, коли й що міняв), повернути колишній стан файла або відновити випадково видалений файл. В основі *Seafile* лежать технології, використовувані в системі управління «сирцевими» текстами *Git*. При цьому *Seafile* не залежить від *Git* й самостійно реалізує потрібні методи, які спрощені й перероблені для виконання завдань автоматичної синхронізації документів, забезпечення оновлення передачі даних у випадку розриву з’єднання й підтримки різних сервісів збереження на стороні серверу. Документи зберігаються з розбиттям на блоки, що підвищує ефективність збереження і дає можливість прискорення передачі файлів за рахунок паралельного завантаження блоків з різних серверів збереження.

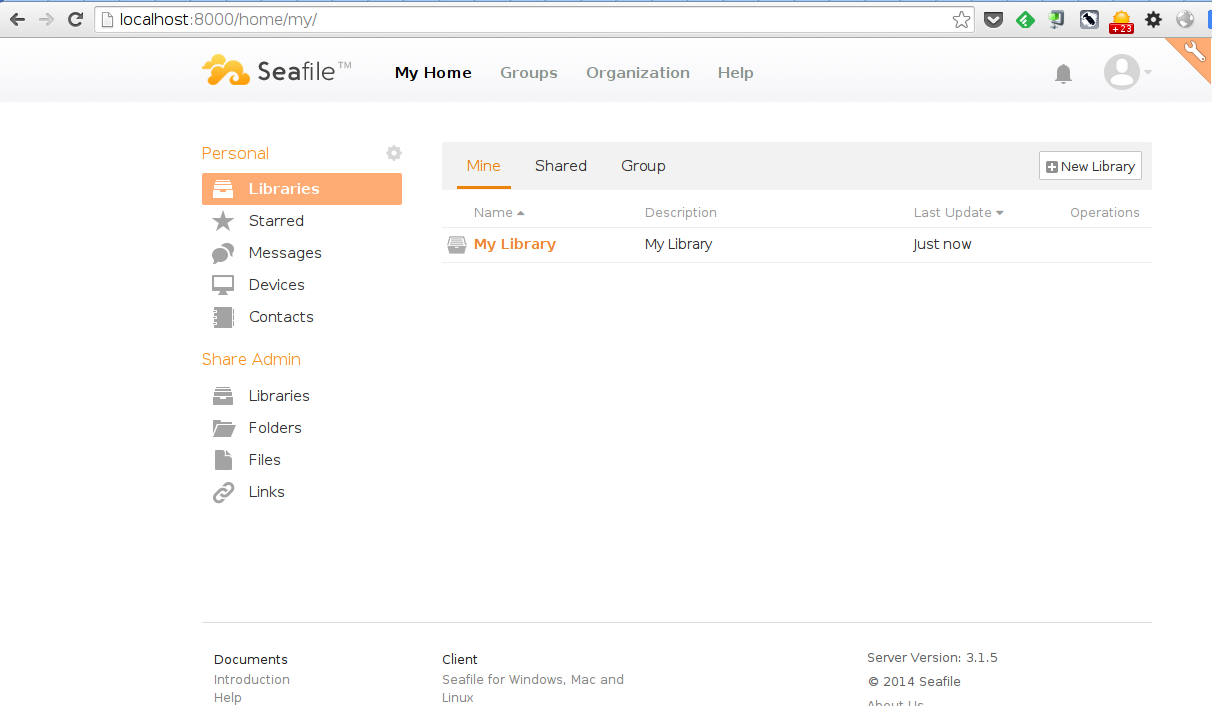


Рис. 1.7. Зовнішній вигляд додатку *Seafile*

## 1.4. Бази документів для збереження даних локально (на сервері)

### 1.4.1. Класифікація баз документів

База документів (БД) - упорядкований набір логічно взаємопов’язаних документів, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система управління БД.

Система управління базами документів (СУБД) - це комплекс програмних і мовних засобів, необхідних для створення баз документів, підтримання їх в актуальному стані та організації пошуку в них необхідної даних [5].

Централізований характер управління даними в базі документів передбачає необхідність існування деякої особи (групи осіб), на яку покладаються функції адміністрування даними, що зберігаються в базі.

Головним завданням БД є гарантоване збереження значних обсягів даних та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої даних та системи управління нею. З метою забезпечення ефективності доступу записи документів організовують як множину фактів (елемент документів).

Існує величезна кількість різновидів баз документів, що відрізняються за критеріями (наприклад, в Енциклопедії технологій баз документів визначаються понад 50 видів БД). Відзначимо тільки основні класифікації.

Класифікація БД за моделлю документів:

* ієрархічні;
* мережеві;
* реляційні;
* об’єктні;
* об’єктно-орієнтовані;
* об’єктно-реляційні.

Класифікація БД за технологією фізичного збереження:

* БД у вторинній пам’яті (традиційні);
* БД в оперативній пам’яті (*in-memory databases*);
* БД у третинній пам’яті (*tertiary databases*).

Класифікація БД за вмістом:

* географічні;
* історичні;
* наукові;
* мультимедійні.

Класифікація БД за ступенем розподіленості:

* централізовані (зосереджені);
* розподілені.

Окреме місце в теорії та практиці займають просторові (англ. *spatial*), тимчасові, або темпоральні (*temporal*) і просторово-часові (*spatial-temporal*) БД.

Ієрархічні бази документів можуть бути представлені як дерево, що складається з об’єктів різних рівнів. Верхній рівень займає один об’єкт, другий - об’єкти другого рівня і т.д.

Між об’єктами існують зв’язки, кожен об’єкт може включати в себе декілька об’єктів більш низького рівня. Такі об’єкти перебувають у відношенні предка (об’єкт більш близький до кореня) до нащадка (об’єкт більш низького рівня), при цьому можлива ситуація, коли об’єкт-предок не має нащадків або має їх декілька, тоді як у об’єкта-нащадка обов’язково тільки один предок. Об’єкти, що мають загального предка, називаються близнюками [6].

Мережеві бази документів подібні до ієрархічних, за винятком того, що в них є покажчики в обох напрямках, які з’єднують споріднену інформацію.

До основних понять мережевої моделі бази документів відносяться: рівень, елемент (вузол), зв’язок.

Вузол - це сукупність атрибутів документів, що описують деякий об’єкт. На схемі ієрархічного дерева вузли представляються вершинами графа.

Незважаючи на те, що ця модель вирішує деякі проблеми, пов’язані з ієрархічною моделлю, виконання простих запитів залишається досить складним процесом.

Також, оскільки логіка процедури вибірки документів залежить від фізичної організації цих документів, то ця модель не є повністю незалежною від програми. Іншими словами, якщо необхідно змінити структуру документів, то потрібно змінити і додаток.

Реляційна модель орієнтована на організацію документів у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

* кожен елемент таблиці - один елемент документів;
* всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символьний тощо);
* кожен стовпчик має унікальне ім’я;
* однакові рядки в таблиці відсутні;
* порядок проходження рядків і стовпчиків може бути довільним.

Об’єктна СУБД ідеально підходить для інтерпретації складних документів, на відміну від реляційних СУБД, де додавання нового типу документів досягається ціною втрати продуктивності або за рахунок різкого збільшення термінів і вартості розробки додатків. Новий клас і його екземпляри просто надходять у зовнішні структури бази документів. Система управління ними залишається без змін [7].

Об’єктно-орієнтована база документів (ООБД) - база документів, в якій дані оформлені у вигляді моделей об’єктів, що включають прикладні програми, які управляються зовнішніми подіями. Результатом поєднання можливостей (особливостей) баз документів і можливостей об’єктно-орієнтованих мов програмування є об’єктно-орієнтовані системи управління базами документів (ООСУБД). ООСУБД дозволяють працювати з об’єктами баз документів також, як з об’єктами у програмуванні в об’єктно-орієнтованих мовах програмування. ООСУБД розширює мови програмування, прозоро вводячи довготривалі дані, управління паралелізмом, відновлення документів, асоційовані запити й інші можливості.

Об’єктно-орієнтовані бази документів звичайно рекомендовані для тих випадків, коли потрібна високопродуктивна обробка документів, які мають складну структуру.

Система, яка забезпечує об’єктну інфраструктуру і набір реляційних розширювачів, називається «об’єктно-реляційною».

Об’єктно-реляційні системи поєднують переваги сучасних об’єктно-орієнтованих мов програмування з такими властивостями реляційних систем як множинні представлення документів і високорівневі непроцедурні мови запитів.

За технологією обробки документів бази документів поділяються на централізовані й розподілені.

Централізована база документів зберігається у пам’яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом мережі ЕОМ, можливий розподілений доступ до такої бази.

Розподілена база документів складається з декількох, можливо пересічних або навіть дублюючих одна одну частин, які зберігаються в різних ЕОМ обчислювальної мережі. Робота з такою базою здійснюється за допомогою системи управління розподіленою базою документів (СУРБД).

За способом доступу до документів бази документів поділяються на бази документів з локальним доступом і бази документів з віддаленим (мережевим) доступом.

Системи централізованих баз документів з мережевим доступом припускають різні архітектури подібних систем:

* файл-сервер;
* клієнт-сервер.

Файл-сервер. Архітектура систем БД з мережевим доступом передбачає виділення однієї з машин мережі в якості центральної (сервер). На такій машині зберігається спільно використовувана централізована БД (рис. 1.8). Усі інші машини мережі виконують функції робочих станцій, за допомогою яких підтримується доступ користувальницької системи до централізованої бази документів. Файли бази документів відповідно до призначених для користувача запитів передаються на робочі станції, де в основному і проводиться обробка. Користувачі можуть створювати також на робочих станціях локальні БД, які використовуються ними монопольно [8].

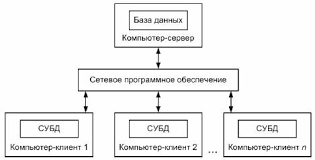


Рис. 1.8. Архітектура файл-сервер

Клієнт-сервер. У цій концепції мається на увазі, що крім збереження централізованої бази документів центральна машина (сервер бази документів) повинна забезпечувати виконання основного обсягу обробки документів( рис. 1.9). Запит на дані, який видається клієнтом (робочою станцією), породжує пошук і вилучення документів на сервері. Витягнуті дані (але не файли) транспортуються по мережі від сервера до клієнта. Специфікою архітектури клієнт-сервер є використання мови запитів *SQL*.

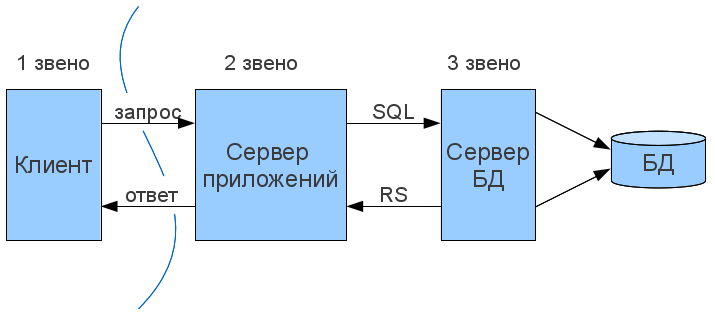


Рис. 1.9. Архітектура клієнт-сервер

### 1.4.2. Області застосування баз документів

Історично системи управління базами документів орієнтувалися на вирішення завдань, пов’язаних у першу чергу з транзакційною обробкою структурованої даних. Безумовно, найкращим, перевіреним часом рішенням тут була і залишається реляційна модель СУБД. Однак в останні роки область застосування баз документів незмінно розширювалася. З одного боку, потрібно керувати більш широким набором форматів документів, переходячи до вирішення спільних проблем управління корпоративною інформацією. З іншого - саме СУБД беруть на себе основні функції інтеграції документів і додатків корпоративних систем. (За даними *Gartner Group*, інформаційні відділи підприємств витрачають до 40% свого бюджету на вирішення завдань інтеграції діючих компонентів баз документів.) Саме цим пояснюється активний інтерес до обговорення архітектурних принципів і можливостей реалізації баз документів різних моделей - постреляційних, об’єктно-реляційних, *XML*.

Якщо постаратися класифікувати існуючі області застосування баз документів, а так само оцінити перспективи їхнього розвитку в даний час, то можна отримати приблизний список найбільш поширених класів:

* документографічні й документальні застосовуються у усіх базах органів влади та управління;
* бази документів з промислової, будівельної та сільськогосподарської продукції;
* бази документів з економічної та кон’юнктурної даних (статистична, кредитно-фінансова, зовнішньоторговельна);
* фактографічні бази соціальних документів, які включають відомості про населення і про соціальні середовища;
* бази документів транспортних систем;
* довідкові дані для населення та установ (енциклопедії та довідники, розклади літаків і поїздів, адреси та телефони громадян і організацій);
* ресурсні бази документів, що включають фактографічну інформацію про природні ресурси (земля, вода, надра, біоресурси, гідрометеорологія, вторинні ресурси і відходи, екологічна обстановка);
* фактографічні бази і банки наукових документів, щоб забезпечити фундаментальні наукові дослідження;
* фактографічні бази документів у галузі культури і мистецтва;
* лінгвістичні бази документів, тобто машинні словники різного типу і призначення.

Останнім часом утворилися нові важливі області застосування баз документів, і кожна з них представляє принципово нове середовище, до якого необхідно адаптувати технології СУБД. Ці області отримали на ринку назви інтелектуально аналізу документів (*data mining*), сховищ документів (*data warehousing*), репозитаріїв документів (*data repository*).

Інтелектуальний аналіз документів. Ідея інтелектуально аналізу документів (*data mining*), тобто добування даних з величезних масивів документів, накопичених зовсім для інших цілей, викликає сьогодні підвищений ентузіазм. Наприклад, авіакомпанії домагаються оптимального заповнення рейсів за рахунок аналізу накопичених раніше документів про резервування квитків. Можна навести одну історію про те, як була виявлена несподівана кореляція між покупками пива і покупками серветок у післяобідній період. Власник магазину наблизив один до одного відділи, які торгують пивом і серветками, а між ними помістив ще прилавки з картопляними чіпсами. У результаті збільшилися продажі усіх трьох видів товару[9].

Із запитами, характерними для систем інтелектуально аналізу документів, пов’язана низка незвичайних проблем:

1. Вони включають, як правило, агрегацію величезних обсягів документів.
2. Вони мають нерегламентований характер; їх формулюють особи, відповідальні за прийняття рішень, коли їм необхідно виявити будь-які неочевидні взаємозв’язки.
3. У додаткух, пов’язаних, наприклад, з торгівлею цінними паперами, дуже важливий малий час відповіді. Суть проблеми полягає тут у тому, щоб скоротити загальний час, необхідний для написання, налагодження та виконання запиту.

Досить часто користувач не в змозі точно сформулювати запит - йому просто потрібно виявити «що-небудь цікаве».

Таким чином, з видобутком документів пов’язані такі дослідницькі напрями:

* методи оптимізації складних запитів, які включають, наприклад, агрегацію та групування;
* методи підтримки «багатовимірних» запитів, що відносяться до документів, організованих у вигляді «куба», в осередках якого знаходяться потрібні дані;
* методи оптимізації використання третинної пам’яті;
* мови запитів дуже високого рівня, а також інтерфейси для підтримки користувачів, які не є експертами, і яким потрібні відповіді на нерегламентовані запити.

Сховища документів. У сховищі документів накопичуються дані з однієї або більше баз документів. Існує безліч потенційних застосувань, а також підходів до організації сховищ документів. Наприклад, великий магазин може підтримувати сховище документів на основі транзакційних документів про касові операції для цілей видобутку документів. У сховищі документів може зберігатися інформація з багатьох баз документів для використання в надзвичайних ситуаціях. Наприклад, в єдиному сховищі документів підтримуються відомості про цивільну інфраструктуру (дороги, мости, трубопроводи тощо), оскільки, наприклад, після землетрусу навряд чи вдасться отримати цю інформацію з міст, що знаходяться поблизу епіцентру. Ще один приклад - використання сховища документів як «матеріалізованого уявлення» інтегрованої даних. Альтернативою медіаторних систем, які дають цілісне уявлення документів, витягнутих з безлічі джерел, можуть служити сховища документів, що забезпечують фізичне збереження інтегрованих документів. На відміну від сховищ, медіатори надають інформацію, розсилаючи запити декільком джерелам, подібно до того, як це відбувається при реалізації уявлень [10].

Деякі дослідницькі проблеми, що стосуються сховищ документів, збігаються з тими, які характерні для інтеграції документів у цілому, але є і деякі специфічні проблеми.

1. Інструменти для створення насосів документів (*data pump*), тобто модулів, що функціонують за середовищем джерел документів і поставляють у сховище ті зміни, які істотні з точки зору сховища; при цьому дані мають транслюватися у відповідності з глобальною моделлю і схемою сховища.
2. Методи «чистки документів» (*data scrubbing*), які забезпечують узгодження документів, видалення елементів, що відповідають різним уявленням одного й того ж об’єкта (наприклад «*Sally Tones*» і «*SA Tones*»), а також видалення неправдоподібних значень.
3. Засоби для створення і підтримки метасловника, який інформує користувачів про способи отримання документів.

Репозитарії. Програми, що відносяться до категорії репозитаріїв, характеризуються тим, що вони призначені для збереження і управління як даними, так і метаданими, тобто інформацією про структуру документів. Приклади репозитаріїв - бази документів для підтримки комп’ютерного проектування, включаючи *CASE* (системи проектування програмного забезпечення), а також системи управління документами. Відмінна риса цих систем - часті зміни метадокументів, характерні для будь-якого середовища проектування.

У репозитарії необхідно підтримувати безліч уявлень однієї й тієї ж або схожої даних. Репозитарії повинні підтримувати поняття версій (моментальних знімків елементів документів, що змінюються в часі) і конфігурацій (версіонних колекцій версій). Наприклад, різні релізи програмної системи будуть зазвичай формуватися як конфігурації з певних версій файлів вихідного коду.

Сховище повинне підтримувати еволюцію структури даних та її метадокументів таким чином, щоб при додаванні нових властивостей документів або нових зв’язків не була потрібна повна перекомпіляція.

## 1.5. Коротка характеристика деяких СУБД

### 1.5.1. *MySQL*

*MySQL* - вільна система управління базами документів. *MySQL* є власністю компанії *Oracle Corporation*, що отримала її разом з поглиненою *Sun Microsystems*, яка здійснює розробку і підтримку додатку. Розповсюджується під *GNU* *General* *Public* *License* і під власною комерційною ліцензією, на вибір. Крім цього розробники створюють функціональність на замовлення ліцензійних користувачів, саме завдяки такому замовленню майже в найраніших версіях з’явився механізм реплікації.

Цю систему управління базами документів з відкритим кодом було створено як альтернатива комерційним системам. *MySQL* із самого початку була дуже схожою на *mSQL*, проте з часом вона все розширювалася і зараз *MySQL* - одна з найпоширеніших систем управління базами документів.

*MySQL* є рішенням для малих і середніх додатків (рис. 1.10). Зазвичай *MySQL* використовується як сервер, до якого звертаються локальні або віддалені клієнти, проте до дистрибутиву входить бібліотека внутрішнього сервера, що дозволяє включати *MySQL* до автономних програм. Вихідні коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в *UNІХ*-системах, де є підтримка багатонитевості, що підвищує продуктивність системи в цілому.

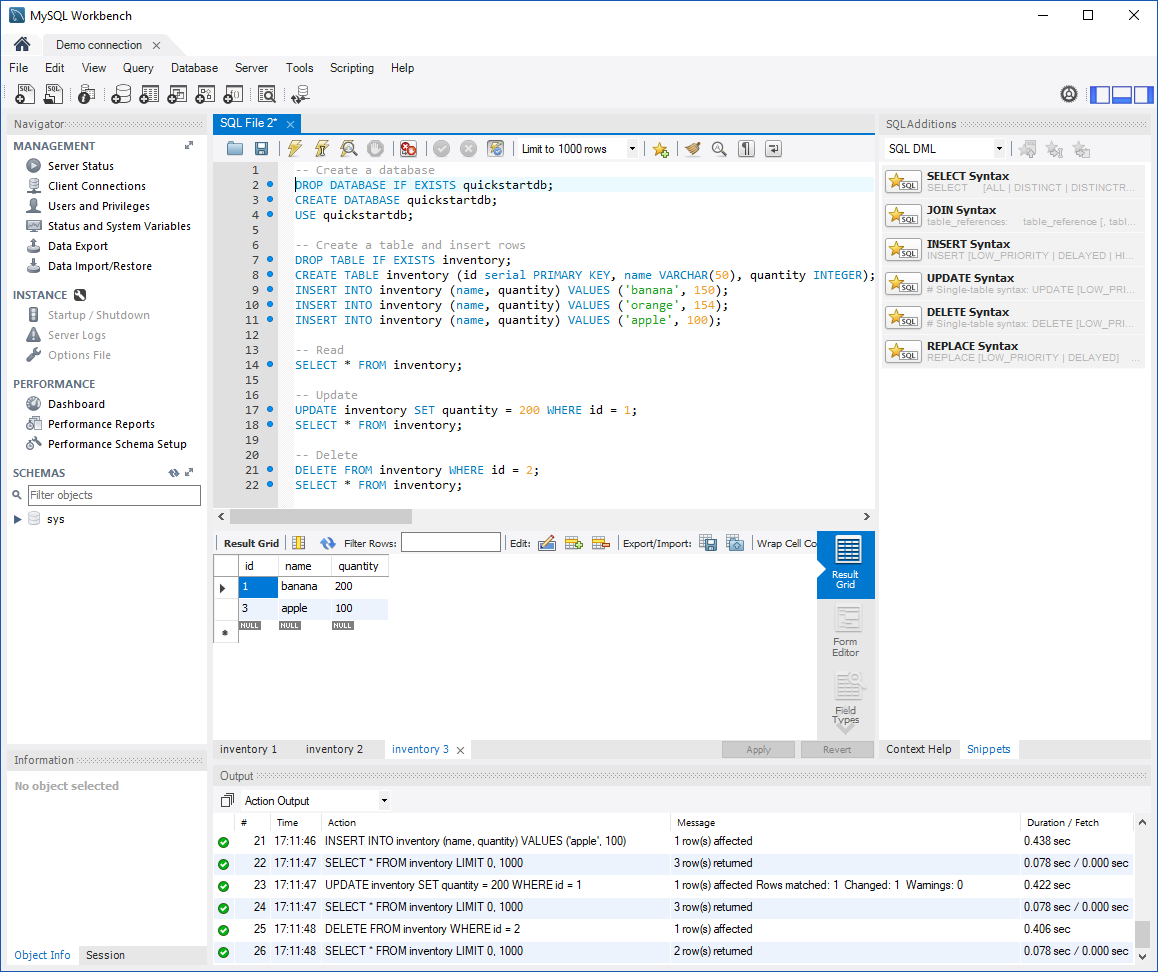


Рис. 1.10. Додаток для управління СУБД *MySQL*

Гнучкість СУБД *MySQL* забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу *MyISAM*, що підтримують повнотекстовий пошук, так і таблиці *InnoDB*, що підтримують транзакції на рівні окремих записів. Більш того, СУБД *MySQL* поставляється із спеціальним типом таблиць *EXAMPLE*, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі й *GPL*-ліцензуванню, в СУБД *MySQL* постійно з’являються нові типи таблиць. *MySQL* характеризується великою швидкістю, стійкістю і простотою використання.

Для некомерційного використання *MySQL* є безкоштовною. Можливості сервера *MySQL*:

* простота у встановленні та використанні;
* підтримується необмежена кількість користувачів, що одночасно працюють із БД;
* кількість рядків у таблицях може досягати 50 млн.;
* висока швидкість виконання команд;
* наявність простої та ефективної системи безпеки.

### 1.5.2. *Oracle*

СУБД *Oracle* - це найпотужніший програмний комплекс, що дозволяє створювати додатки будь-якої складності (рис. 1.11). Ядром цього комплексу є база документів, що зберігає інформацію, кількість якої за рахунок надокументів засобів масштабування практично безмежна. З високою ефективністю працювати з цією інформацією одночасно може практично будь-яка кількість користувачів (за наявності достатніх апаратних ресурсів), не проявляючи тенденції до зниження продуктивності системи при різкому збільшенні їхньої кількості. Компанія до цьогочасу підтримує даний проект, тому в надійності системи можна не сумніватися.

Захищеність документів в СУБД організована таким чином, що тільки маючи спеціальний пароль доступу можна отримати доступ до СУБД. Це дає багато переваг у виборі даної СУБД. Але є і мінуси, наприклад, складність управління.

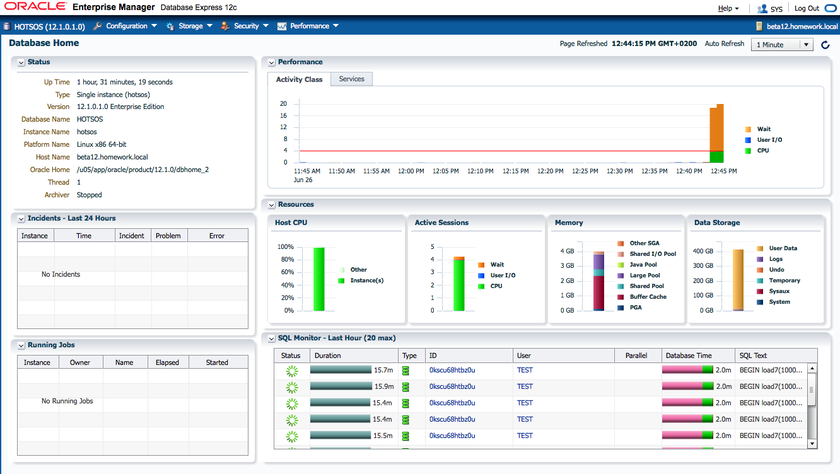


Рис. 1.11. Додаток для управління СУБД *Oracle*

Механізми масштабування в СУБД *Oracle* останньої версії дозволяють безмежно збільшувати потужність і швидкість роботи сервера *Oracle* і своїх додатків, просто додаючи нові й нові вузли кластеру. Це не вимагає зупинки працюючих додатків, не вимагає переписування старих додатків, розроблених для звичайної одномашинної архітектури. Крім того, вихід з ладу окремих вузлів кластера також не призводить до зупинки програми.

Вбудовування до СУБД *Oracle JavaVM*, повномасштабної підтримки серверних технологій (*Java Server Pages*, *Java*-сервлети, модулі *Enterprise JavaBeans*, інтерфейси прикладного програмування *CORBA*), призвели до того, що *Oracle* на сьогоднішній день де-факто є стандартом СУБД для *Internet*.

Ще однією складовою успіху СУБД *Oracle* є те, що вона поставляється практично для усіх існуючих на сьогодні операційних систем. Працюючи під *Sun Solaris*, *Linux*, *Windows* або на інший операційній системі з продуктами *Oracle* не буде виникати ніяких проблем у роботі. СУБД *Oracle* однаково добре працює на будь-якій платформі. Таким чином, компаніям, які розпочинають роботу з продуктами *Oracle* не доводиться міняти мережеве оточення. Існує лише невелика кількість відмінностей при роботі з СУБД, обумовлених особливостями тієї або іншої операційної системи. У цілому ж це завжди та ж сама безпечна, надійна і зручна СУБД *Oracle*.

### 1.5.3. *Microsoft SQL Server*

*Microsoft SQL Server* - система управління реляційними базами документів, розроблена корпорацією *Microsoft* (див. рис. 1.12). Основна використовувана мова запитів - *Transact*-*SQL*, створена спільно *Microsoft* та *Sybase*. *Transact-SQL* є реалізацією стандарту *ANSI* / *ISO* щодо структурованої мови запитів (*SQL*) із розширеннями. Використовується для роботи з базами документів розміром від персональних до великих баз документів масштабу підприємства, конкурує з іншими СУБД у цьому сегменті ринку.

При взаємодії з мережею *Microsoft SQL Server* і *Sybase* *ASE* використовують протокол рівня додатків під назвою *Tabular Data Stream* (*TDS*, протокол передачі табличних документів). Протокол *TDS* також був реалізований у проекті *FreeTDS* з метою забезпечити різні додатки можливістю взаємодії з базами документів *Microsoft SQL Server* і *Sybase*.

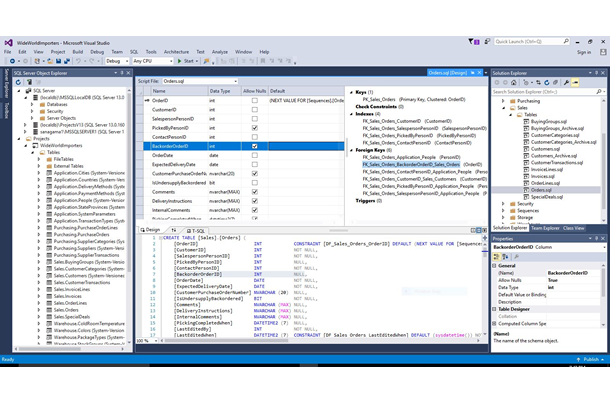


Рис. 1.12. Додаток для управління СУБД *Microsoft SQL Server*

Для забезпечення доступу до документів *Microsoft SQL Server* підтримує *Open* *Database Connectivity* (*ODBC*) - інтерфейс взаємодії додатків з СУБД. *SQL* *Server* надає можливість підключення користувачів через веб-сервіси, що використовують протокол *SOAP*. Це дозволяє клієнтським програмам, не призначеним для *Windows*, кросплатформно з’єднуватися з *SQL Server* [11].

## 1.6. Порівняння найактуальніших серверних СУБД

На підставі аналізу розглянутих СУБД можна скласти таблицю ефективності СУБД із виділенням переваг і недоліків.

Таблиця 1.1

Ефективність СУБД із виділенням переваг і недоліків

|  |  |  |
| --- | --- | --- |
| СУБД | Переваги | Недоліки |
| *Informix Dynamic Server* | Широкий набір архітектурних особливостей, що забезпечують високу продуктивність, внутрішні механізми підтримки масштабованості, багаті можливості вбудованого язика маніпулювання даними, підтримка декількох апаратних платформ й операційних систем. | Чим більше можливостей, тим більше тонкощів у настроюванні сервера для того або іншого завдання. Адміністрування системи вимагає високого рівня професіоналізму. Набагато складніше функціонал і сама робота в *Informix DS*, ніж в *MS SQL Server.* |
| *MS SQL Server* | Ідеально підходить для операційної системи *Windows*. Крім багатого набору програмних засобів для розробки й адміністрування, привабливість серверу надає наявність високоінтелектуального процесора запитів і добре | Такі важливі параметри СУБД, як продуктивність і масштабованість, залежать від операційного середовища - *Windows*. Сервер орієнтований головним чином на інтеграцію з іншими продуктами |

Закiнчення таблицi 1.1

|  |  |  |
| --- | --- | --- |
| СУБД | Переваги | Недоліки |
|  | розвиненого діалекту мови *SQL* (*Transact-SQL*). | *Microsoft*; можливостей інтеграції із програмним забезпеченням інших виробників набагато менше. В основі СУБД лежить принципова відмова від підтримки операційних систем, що не належать сімейству *Windows*. |
| *DB2 Universal Database* | Висока продуктивність, багаті можливості масштабування, графічний інтерфейс розроблювача й адміністратора баз документів, багатоплатформеність; підтримка об’єктно-реляційної парадигми й стандарту *SQL3*. | На відміну від *MS SQL Server*, дуже складний і незручний інтерфейс і функціонал. |
| *Oracle 9i* | Орієнтація на *Internet*, підтримка великої кількості апаратних і програмних платформ; багаті можливості для розроблювачів (об’єктно-реляційна БД, *PL/SQL* і т.д.). | Відносно висока ціна, тяжке адміністрування; широкі можливості сервера вимагають високу кваліфікацію розроблювачів й адміністраторів. |

**Висновки**. На підставі аналізу (див. табл. 1.1) можна зробити такі висновки, що майже всі серверні СУБД мають можливість:

* реалізації на декількох платформах;
* використання зручних адміністративних утиліт;
* здійснення резервного копіювання документів;
* підтримки декількох сценаріїв реплікацій;
* підтримки паралельної обробки документів у багатопроцесорних системах;
* підтримки *OLAP* і створення сховищ документів;
* виконання розподілених запитів і транзакцій;
* використання різних засобів проектування документів для створення своїх об’єктів;
* підтримки засобів розробки й генераторів звітів як власного виробництва, так й інших виробників;
* підтримки як мінімум публікації документів в *Internet*.

## 1.7. Висновки до розділу

Розглянута проблема організації роботи з документами. Сучасні компанії працюють з великими обсягами різнорідних документів. За відсутності засобу організації процесів збереження та обміну документами працівники змушені користуватись неоптимальними, відносно їх потреб інструментами. Сформульовані характеристики, якими повинна володіти система збереження та швидкого доступу до електронних документів для сучасних малих компаній.

Розглянуті способи збереження даних. Можна зробити висновок, що персональні методи збереження документів, такі як жорсткі диски та флеш накопичувачі не відповідають вимогам працівників. Ця проблема потребує більш комплексного рішення, яке об’єднає документи компінії в єдину систему, яка повинна мати функціонал структурування та захисту великих обсягів документів.

Досліджені відкриті програмні рішення не відповідають сформульованим вимогам. Організації постачальники пропонують платні програмні системи, спеціалізовані на рішенні даної проблеми, які не підходять малим компаніям, які не мають достатньої кількості грошових ресурсів.

Внаслідок проведеного аналізу можна стверджувати, що використання *MS SQL Server* ідеально підходить для операційних систем *Windows*, має у наявності високоінтелектуальний процесор запитів й добре розвинений діалект мови *SQL* (*Transact-SQL*). *Oracle 9i* орієнтується на *Internet*, підтримує велику к-сть апаратних й програмних платформ, має багаті можливості для розробників (об’єктно-реляційна БД, *PL/SQL,* тощо). СУБД *Informix* здатна обслуговувати одночасно працюючі додатки оперативної обробки транзакцій й системи підтримки прийняття рішень для локальних й розподілених баз документів, з великою кількістю користувачів. *DB2 Universal Database* поєднує у собі високу продуктивність систем обробки транзакцій в режимі *on-line*, об’єктно-реляційні розширення, удосконалені методи оптимізації з можливостями паралельної обробки і підтримкою дуже великих за обсягом баз документів.

Існуючі можливості серверних СУБД відбивають сучасні тенденції розвитку інформаційних систем, такі, як використання багатопроцесорних систем і розподіленої обробки документів, створення розподілених систем, зокрема з використанням технологій *Internet*, застосування засобів швидкої розробки додатків, створення систем підтримки прийняття рішень із використанням аналітичної обробки документів, а також усе більше підвищуються вимоги до надійності інформаційних систем.

Отже, було вирішено спроектувати та розробити власну програмну систему. Необхідно виконати такі завдання:

* розробити архітектуру системи;
* розробити функціонал системи;
* розробити інтерфейс користувача;
* виконати тестування системи.

# **РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ СТВОРЕННЯ ФАЙЛОВИХ СХОВИЩ**

## 2.1. Технології організації сховищ даних

Основою усіх сховищ є деякий набір протоколів фізичного збереження документів. Сьогодні використовуються три основні класи фізичних моделей збереження документів.

Методи інтелектуального аналізу даних часто розглядають як природний розвиток концепції сховищ документів. Головна відмінність сховища від бази документів полягає в тому, що їх створення і експлуатація переслідують різну мету. База документів відіграє роль помічника в оперативному управлінні організацією. Це щоденні задачі отримання актуальної даних: бухгалтерські звітності, облік договорів, тощо. Сховище документів накопичує всі необхідні дані для здійснення задач стратегічного управління в середньостроковому і довгостроковому періоді. Наприклад, продаж товару і генерація рахунку проводяться з використанням бази документів, а аналіз динаміки продажів за декілька років, що дозволяє спланувати роботу з постачальниками - за допомогою сховища документів [12].

Сховище документів (*Data Warehouse*) - це систематизована інформація з різнорідних джерел, яка є необхідною для обробки з метою ухвалення стратегічно важливих рішень.

Сховище будується на основі клієнт-серверної архітектури, СУБД і утиліт підтримки прийняття рішень. Дані, що надходять у сховище, стають доступні тільки для читання.

Властивості сховища документів:

* предметна орієнтація (інформацію організовано відповідно до основних аспектів діяльності);
* інтегрованість документів (дані в сховище надходять з різних джерел і відповідно агрегуються);
* стабільність, інваріантність у часі (записи в *DW* ніколи не змінюються, являючи собою відбитки документів, зроблені у певний час);
* мінімізація збитковості даних (перед завантаженням у сховища дані фільтруються, зберігаються у певній послідовності, а також формується деяка підсумкова інформація).

В сховищах документів надмірність документів є мінімальною (приблизно 1%), оскільки:

* при завантаженні у сховище дані сортуються і фільтруються;
* інформація у сховищах зберігається в хронологічному порядку, що майже повністю виключає перекриття документів;
* при завантаженні у сховище дані зводяться до єдиного формату, включаючи обчислення підсумкових (агрегованих) показників.

Сервери багатовимірних баз документів можуть зберігати дані по-різному, крім агрегованих показників формується ще й додаткова інформація: поля часу, дати; адресні посилання, таблиці метадокументів тощо. Це приводить до значного збільшення даних. Вхідний масив розміром 200 *Mb* може розростись до об’єму 5 *Gb*. Сховище документів повинне бути оптимально організованою базою документів, яка забезпечує максимально швидкий і оперативний пошук даних.

Вітрина документів - це спрощений варіант сховища документів, що містить лише тематично орієнтовані, агреговані дані [13].

Глобальне сховище документів складається з трьох рівнів:

1. Сховище агрегованих документів;
2. Вітрини документів, які базуються на даних зі сховища документів;
3. Клієнтські робочі місця, на яких встановлено засоби оперативного аналізу документів.

У розпорядженні виробників прикладних програмних засобів є три різні технології роботи з базами документів:

* *DAO* (*Data Access Objects*) - доступ до локальних баз документів;
* *RDO* (*Remote Data Objects*) - доступ до віддалених баз документів;
* *ADO* (*ActiveX Data Objects*) - доступ до *Windows*-додатків через Інтернет. В основному використовується з міркувань безпеки.

Одним із перспективних напрямів удосконалення доступу до документів є гнучке конфігурування системи, коли розподіл між клієнтською і серверною частинами можливий за допомогою використання механізму віддалених процедур.

Поряд із потоками документів існують й потоки метадокументів, які розміщуються в репозитарії. Він дає змогу визначити семантичну структуру додатку у вигляді опису термінів предметної галузі, їхні взаємозв’язки й атрибути.

Метадані - це інформація про дані, які визначають джерело, приймач та алгоритм трансформації документів під час перенесення їх від джерела до приймача.

Метадані містять:

* описи структур документів та їхніх взаємозв’язків;
* інформацію про джерела документів і про ступінь їх вірогідності;
* інформацію про власників документів, права доступу;
* схему перетворення стовпців вхідних таблиць у стовпці кінцевих таблиць;
* правила підсумовування, консолідації та агрегування документів;
* інформацію про періодичність оновлення документів;
* каталог використаних таблиць, стовпців та ключів;
* фізичні атрибути стовпців;
* кількість табличних рядків та обсяг документів;
* часові ярлики (дата та час створення/модифікації записів);
* статистичні оцінки часу виконання запитів.

Контроль модифікації (*versioning*) полягає у властивості метадокументів відслідковувати зміни в структурі документів та їх значення в часі.

Функціональна архітектура сховища документів містить наступні компоненти:

* сховище документів;
* клієнтська частина системи (дизайнери сховища, засоби розробки додатків, засоби адміністрування, інструменти аналізу документів, завантаження словника метадокументів з *XML*-файлу у сховище і експорт його зі сховища в *XML*-файл;
* сервер обміну даними (*Data Exchange Server*) - набір програм імпорту/експорту документів зі сховища й каталогів для організації обміну даними із зовнішніми *OLTP*-системами;
* бібліотеки прикладних класів: *ACL* (*Application Class Library*), *VCL* (*Visual Component Library*), *Win Lite*.

Наповнення інформаційних сховищ відбувається у декілька етапів:

* екстракція (витяг) - імпорт документів у сховище з інформаційних підсистем, виробничих відділів та інших джерел;
* трансформація - консолідування, агрегування документів, розбиття їх на фракції, коригування та трансформування у відповідні формати;
* завантаження - у сховище, синхронізація з датою або зовнішніми подіями.

Обслуговування інформаційних сховищ полягає в: копіюванні баз документів, налаштуванні, тиражуванні, надсиланні застарілих баз документів до архіву, управлінні правами користувачів, створенні та редагуванні графічних діаграм баз документів, тощо.

Типи архівації у сховищах поділяють на:

* звичайна;
* копіювальна;
* додаткова;
* диференціальна;
* щоденна.

Архівні магнітні носії зберігають у вогнетривких сейфах або за межами обчислювального центру. Крім того, розробляється план архівації компонентів сервера баз документів. Сучасні сервери автоматично підтримують копію свого каталогу на кожному сервері вузла. Цей процес називається копіюванням каталогів (*directory replication*).

Звичайна архівація каталогів на усіх серверах здійснюється раз на тиждень у вихідні дні, а диференціальна – щодня, у робочі дні. У річному архіві, як правило, зберігаються дані останнього тижня місяця. Усі зміни в каталозі сервера, а також в особистих і загальних сховищах записуються у файли, які називаються журналами транзакцій (*transaction log files*).

Під час виконання додаткової архівації каталога або інформаційного сховища архівуванню підлягають лише журнали транзакцій [14].

Для ефективної роботи із сховищем документів, необхідно зібрати максимум даних про процес. Наприклад, для прогнозування обсягів продажів можуть бути використані бази документів облікових систем підприємства, маркетингові дані, відгуки клієнтів, дослідження конкурентів і т.п.

Необхідною для прогнозу є наступна інформація:

* хронологія продажів;
* стан складу на кожний день - якщо спад продажів буде пов’язаний із відсутністю товару на складі, а не через відсутність попиту;
* відомості про ціни конкурентів;
* зміни у законодавстві;
* загальний стан ринку;
* курс долара, інфляція;
* відомості про рекламу;
* відомості про відношення до продукції клієнтів;
* різного роду специфічну інформацію.

Наприклад, для продавців морозива - температуру, а для фармакологічних складів - санітарно-епідеміологічний стан, тощо.

Проблема полягає в тому, що зазвичай у системах оперативного обліку більша частина ціх даних відсутня, а наявна - неповна або спотворена. Кращим варіантом в цьому випадку буде створення сховища документів, куди б із певною заданою періодичністю надходила вся необхідна інформація, заздалегідь систематизована і очищена (див. рис. 2.1).



Рис. 2.1. Приклад сховища документів

Ефективна архітектура сховища документів організовується таким чином, щоб бути складовою частиною інформаційної системи управління підприємством [15].

Найбільш поширений випадок, коли сховище організовано за типом «зірка», де в центрі розміщуються факти і агрегатні дані, а «проміннями» є виміри. Кожна «зірка» описує певну дію, наприклад, продаж товару, його відвантаження, надходження коштів й інше (рис. 2.2):

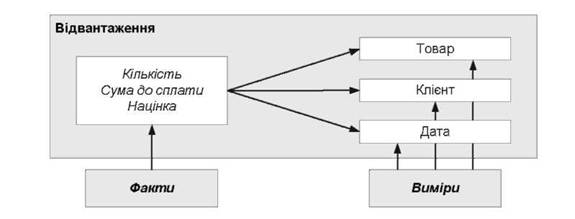


Рис. 2.2. Приклад організації сховища документів за типом «зірка»

Як правило, дані копіюються в сховище з оперативних баз документів і інших джерел відповідно до певних правил.

### 2.1.1. *Direct Attached Storage*

Технологія *DAS* - пряме (безпосереднє) підключення накопичувачів до серверу або до ПК (рис. 2.3). При цьому накопичувачі (жорсткі диски, стрічкові накопичувачі) можуть бути як внутрішніми, так і зовнішніми. Випадок найпростішої *DAS*-системи - це один диск всередині сервера або ПК. Крім того, до *DAS*-системі можна віднести й організацію внутрішнього *RAID*-масиву дисків із використанням *RAID*-контролера.

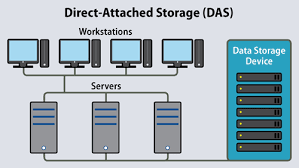


Рис. 2.3. Структура *DAS*

Варто відзначити, що, незважаючи на формальну можливість використання терміну «*DAS*-системи» по відношенню до одиночного диску або до внутрішнього масиву дисків, під *DAS*-системою прийнято розуміти зовнішній блок з дисками, який можна розглядати як автономний СЗД. Крім незалежного живлення, автономні *DAS*-системи мають спеціалізований контролер (процесор) для управління масивами накопичувачів. Наприклад, в якості такого контролера може виступати *RAID*-контролер із можливістю організації *RAID*-масивів різних рівнів [16].

Слід зазначити, що автономні *DAS*-системи можуть мати кілька зовнішніх каналів введення-виведення, що забезпечує можливість підключення до *DAS*-системи декількох комп’ютерів одночасно.

В якості інтерфейсів для підключення накопичувачів (внутрішніх або зовнішніх), в технології *DAS* можуть виступати інтерфейси *SCSI* (*Small Computer Systems Interface*), *SATA*, *PATA* й *Fibre* *Channel*. Якщо *SCSI* інтерфейси, *SATA* і *PATA* застосовуються переважно для підключення внутрішніх накопичувачів, то інтерфейс *Fibre Channel* служить виключно для підключення зовнішніх накопичувачів й автономних СЗД. Перевага інтерфейсу *Fibre Channel* тут полягає у тому, що він не має жорсткого обмеження по довжині й може використовуватися у тому випадку, коли сервер або ПК, що підключається до *DAS*-системи, знаходиться від неї на значній відстані. Інтерфейси *SCSI* та *SATA* також можуть застосовуватися для підключення зовнішніх СЗД (в цьому випадку інтерфейс *SATA* *ESATA*), однак вони мають суворе обмеження по максимальній довжині кабелю, що з’єднує *DAS*-систему і підключаємий сервер.

До основних переваг *DAS*-систем можна віднести їх низьку вартість (у порівнянні із іншими рішеннями СЗД), простоту розгортання й адміністрування, а також високу швидкість обміну документами між системою збереження і з сервером. Власне, саме із цієї причини вони стали дуже популярні у сегменті малих офісів невеликих корпоративних мереж. У той же час *DAS*-системи мають і свої недоліки - в першу чергу - це висока вартість збереження і управління даними внаслідок їх розкиданості по організації, а також вимушений простій мережі в момент додавання нових дисків й необхідність нарощування пам’яті або процесорної потужності сервера при перевищенні певного розміру дискового масиву. Перевантаженість мережевого трафіку із додаванням нових серверів ускладнює проблему захисту документів, перешкоджає ефективному використанню ресурсів і т.д. Витрати і нові проблеми ростуть як сніжний ком [17].

В даний час *DAS*-системи займають лідируюче положення, однак частка цих систем постійно скорочується, і на зміну їй приходять або універсальні рішення із можливістю плавної міграції до *NAS*-систем, або системи, що передбачають можливість їх використання у якості *DAS*-, і *NAS* так - і навіть *SAN*-систем.

### 2.1.2. *Network Attached Storage*

*NAS*-системи - це мережеві системи збереження документів, що підключаються безпосередньо до мережі точно так же, як і мережевий принт-сервер, маршрутизатор або будь-який інший мережевий пристрій. Фактично *NAS*-системи являють собою еволюцію файл-серверів. Для того, щоб зрозуміти різницю між традиційним файл-сервером і *NAS*-пристроєм, згадаємо, що традиційний файл-сервер являє собою виділений комп’ютер (сервер), на якому зберігаються документи, доступні користувачам мережі. Для збереження документів можуть використовуватися жорсткі диски, що встановлюються у сервер (як правило, вони розташовуються у спеціальних кошиках), або підключення до серверу *DAS*-пристрої. Адміністрування файл-сервера реалізується за допомогою серверної операційної системи. Такий підхід до організації систем збереження документів в даний час є найбільш популярним у сегменті невеликих локальних мереж, але має один суттєвий недолік. Справа в тому, що універсальний сервер (та ще й в поєднанні з серверної операційної системою) - аж ніяк не доступне рішення. У той же час більшість функціональних можливостей, властивих універсальному серверу, в файл-сервері просто не використовуються. Ідея полягає в тому, щоб створити оптимізований файл-сервер з оптимізованою операційною системою і збалансованою конфігурацією. Саме цю концепцію і втілюють в собі *NAS*-пристрої, які в цьому сенсі можна розглядати як «тонкі» файл-сервери, або, як їх ще називають, файлери (див. рис. 2.4).

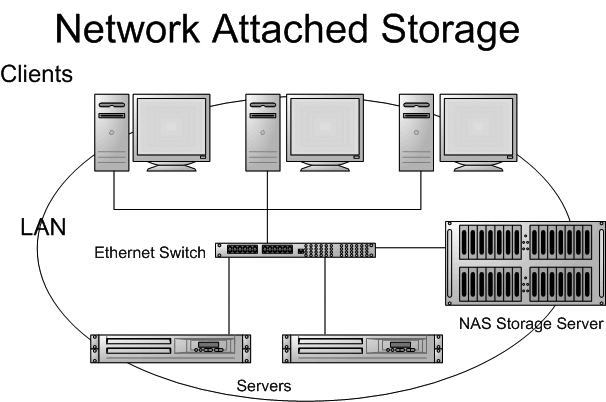


Рис. 2.4. Структура *NAS*

Крім оптимізованої ОС, звільненої від усіх функцій, не пов’язаних із обслуговуванням файлової системи та реалізацією введення-виведення документів, *NAS*-системи мають оптимізовану по швидкості доступу файлову систему. *NAS*-системи проектуються таким чином, що уся їх обчислювальна потужність фокусується виключно на операціях обслуговування та збереженні файлів. Сама операційна система розташовується у флеш-пам’яті та встановлюється фірмою-виробником. Підключення *NAS*-пристроїв до мережі та їх конфігурація є досить простою задачєю і під силу будь-якому досвідченому користувачеві, не кажучи вже про системного адміністратора [18].

У порівнянні із традиційними файловими серверами, *NAS*-пристрої є більш продуктивними та менш дорогими. В даний час практично усі *NAS*-пристрої орієнтовані на використання у мережах *Ethernet* (*Fast Ethernet, Gigabit Ethernet*) на основі протоколів *TCP / IP*. Доступ до пристроїв *NAS* здійснюється з допомогою спеціальних протоколів доступу до інформвції. Найбільш поширеними протоколами файлового доступу є протоколи *CIFS*, *NFS* і ДАФСА.

### 2.1.3. *Storage Area Network*

*SAN* - це спеціалізована мережева інфраструктура для збереження документів (мережа збереження документів). Ці мережі інтегруються у вигляді окремих спеціалізованих підмереж до складу локальної (*LAN*) або глобальною (*WAN*) мережі (див. рис. 2.5).

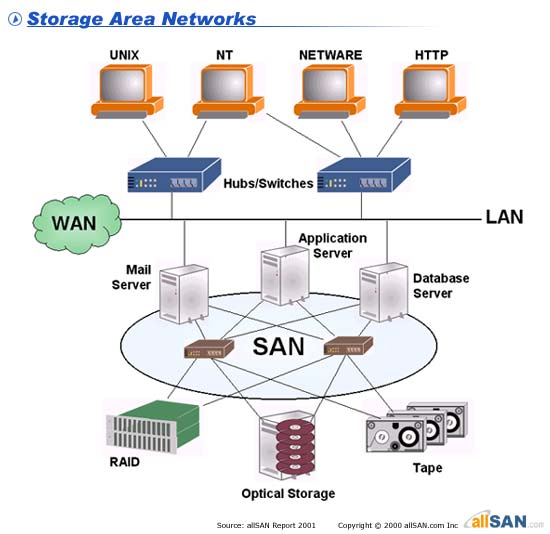


Рис. 2.5. Структура *SAN*

По суті, *SAN-*мережі пов’язують один або кілька серверів (*SAN*-серверів) з одним або декількома пристроями збереження документів. *SAN*-мережі дозволяють будь-якому *SAN*-серверу отримувати доступ до будь-якого пристрою збереження документів, не завантажуючи при цьому на інші сервери та на локальну мережу. Крім того, можливий обмін даними між пристроями збереження документів без участі серверів. *SAN*-мережі дозволяють дуже великому числу користувачів зберігати документи в одному місці (з швидким централізованим доступом) і спільно використовувати її. Так пристрої збереження документів можуть застосовувати *RAID*-масиви, різні бібліотеки (стрічкові, магнітооптичні і ін.), а також *JBOD*-системи (масиви дисків, не об’єднані в *RAID*).

Для побудови мереж *SAN* використовується або стандарт *Fibre Channel* (*FC*), або стандарт *ISCSI*.

### 2.1.4. Віртуалізація

Віртуалізація змінила вигляд сучасних сховищ документів. Подібно до того, як фізичні машини абстрагувалися у віртуальні машини, фізичні сховища абстрагуються у віртуальні диски. Використовуючи приципи віртуалізації монітор віртуальних машин забезпечує емуляцію апаратного середовища для кожної віртуальної машини, включаючи комп’ютер, пам’ять та збереження даних. Основні сучасні рішення, використовують емуляцію локальних фізичних дисків як спосіб надати сховище для кожної віртуальної машини. Іншими словами, використовують *DAS* модель як спосіб реалізувати сховище у віртуальних машинах. Подібно до того, як основною одиницею збереження в *DAS* є фізична машина, основною одиницею у пам’яті віртуального диску є віртуальна машина. Віртуальні диски розглядаються не як незалежні об’єкти, але як частина конкретної віртуальної машини, точно так, як локальні диски є частиною фізичного комп’ютера. Як і *DAS*, віртуальний диск існує та знищується з самою віртуальною машиною. Більшість традиційних засобів віртуалізації використовують модель віртуального дискового сховища [19].

Реалізація віртуальних дисків, у вигляді моделі збереження блочного сховища *DAS*-стилі, на вершині *NAS* або *SAN*, ілюструє одну з важливих характеристик сучасного збереження центрів обробки документів. Так як потік документів із віртуальної машини передається від програмного забезпечення в гіпервізор, а не до апаратних засобів на шині пристрою, протокол, який використовується у віртуальній машині для обміну даними з гіпервізором не повинен відповідати протоколу використовуваному для зв’язку із пристроєм збереження. Це призводить до поділу між моделлю збереження, яка переміщається вгору до віртуальної машини, і протоколом, який використовується гіпервізором щоб зберігати документи, що дає гнучкість змішувати та поєднувати моделі збереження документів і протоколи збереження, та навіть динамічно змінювати протокол збереження без шкоди для віртуальної машини. Вибір реалізації повністю прозорий для додатку, тому що в кінцевому підсумку усі ці протоколи будуть виглядати однаково для віртуальної машини та адміністратора; вони будуть виглядати як локальні налаштовані фізичні диски. Таким чином, розробник програми у більшості інфраструктур публічних хмарних сховищ не може знати, який протокол збереження використовується; Справді, протокол може навіть змінюватися динамічно у режимі реального часу.

Через поділ між моделлю збереження та протоколом збереження, протокол для збереження стає інфраструктурним завданням, в першу чергу важливим для вартості і продуктивності. Функціональні можливості надають програмно визначені рішення [20].

### 2.1.5. Програмно визначене сховище

Програмно визначені сховища (*Software-defined storage, SDS*) - це наступний крок в розвитку програмного забезпечення для організації збереження документів, для управління послугами збереження на основі бізнес-орієнтованих політик (правил) незалежно від апаратного забезпечення (рис. 2.6).

Ідея використання систем збереження документів в яких програмне забезпечення відділене від апаратних засобів є привабливою для малих організацій оскільки дає їм можливість конкурувати з великими компаніями.

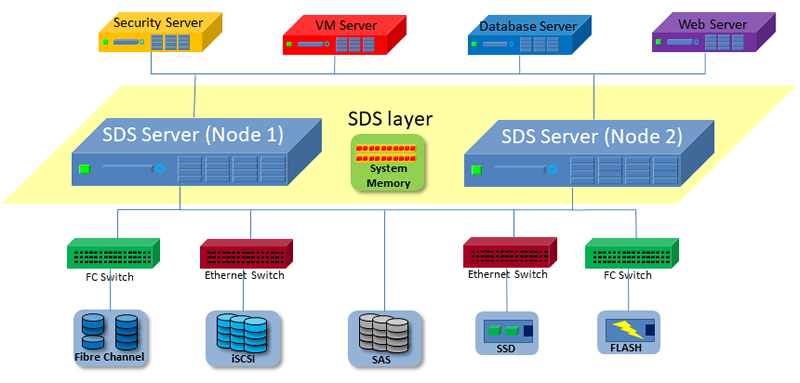


Рис. 2.6. Структура *SDS*

Проте програмно визначені сховища відрізняються від віртуальних сховищ, які дозволяють безлічі пристроїв збереження бути об’єднаними у єдиний віртуальний пристрій. Програмно визначені сховища абстрагують від апаратного забезпечення функціоналу замість ємності. Програмно визначене сховище може бути представлено як програмна система, яка надає послуги збереження документів, включаючи такий функціонал як реплікації, знімки сховища документів і так далі.

Програмно визначені сховища дозволяють інтегрувати наявні апаратні ресурси компанії у єдину інфраструктуру, яка надає можливість змінювати свої параметри в відповідність до потреб організації.

Відокремлюючи апаратне забезпечення сховища від програмного забезпечення яке управляє ним, програмно визначені сховища дозволяють підприємствам використовувати доступне різнорідне апаратне забезпечення.

## 2.2. Типи моделей даних сховищ

Існують 3 основні категорії сховищ документів: файлове, блочне та об’єктне сховище.

### 2.2.1. Файлове сховище

Файлове сховище забезпечує доступ до файлової системи. Це самий знайомий вигляд збереження - це те, з чим ми взаємодіємо на щоденній основі. Користувачі збереження файлів мають доступ до файлів і можуть читати та переписати або весь файл, або його частину. Файлові системи, які операційні системи надають на усіх наших персональних комп’ютерів. У загальному середовищі, для збереження файлів часто розглядається як мережевий диск. Хоча файлове сховище забезпечує зручну модель документів, існують проблеми із масштабуванням. Файлове сховище потребує суворої узгодженості, що створює труднощі коли система зростає і кількість запитів збільшується [21].

### 2.2.2. Блочне сховище

Блочне сховище зберігає структуровані документи, які представлені у вигляді блоків однакового розміру, не змінюючи інтерпретацію збережених бітів. Більшість пристроїв збереження має вбудований блоковий інтерфейс на рівні драйверів. Наприклад, драйвер жорсткого диску записує та зчитує блоки та їхню адресу на відформатованому диску.

Блокові системи збереження використовуються багатьма додаткуми для постійних операцій введення-виведення. Показовий приклад - більшість додатків реляційних СУБД (*Oracle, DB2* і т. П.). Програми, що використовують інші засоби збереження (наприклад, файлові системи), делегують вихідні операції введення-виведення блокових систем збереження базової файлової системи. Ця система відіграє роль проміжного елемента між вихідним інтерфейсом блокової системи збереження та файловими операціями введення-виведення, які генеруються додаткуми (наприклад, *HDFS*, *NFS* і ін.). Мережі збереження документів завжди надають клієнтським застосуванням інтерфейс блокової системи збереження. У багатьох випадках у залежності від підтримки обладнання можна налаштувати розмір блоків та різні параметри (наприклад, розміщення блоків на фактичному носії). Клієнтська програма виконує зіставлення власних форматів збереження із базовою блоковою системою збереження.

В багатьох випадках у залежності від підтримки обладнання можна налаштувати розмір блоків та різні параметри (наприклад, розміщення блоків на фактичному носії). Клієнтська програма виконує зіставлення власних форматів збереження із базової блоковою системою збереження.

Блокова система збереження пропонує більш високу продуктивність та швидкодію у порівнянні із файловими системами збереження. Кожен блок тоді можна розглядати як незалежний диск, керований за допомогою зовнішньої ОС сервера.

### 2.2.3. Об’єктне сховище

Особливістю об’єктного сховища є те, що кожен документ має унікальний глобальний ідентифікатор крім користувацького. Якщо фізичне розташування об’єкта змінюється, зміни обробляються внутрішнім механізмом об’єктного сховища, а ідентифікатор, призначений користувачем або додатком залишається незмінним, що дозволяє легко отримати доступ до нього в будь-який інший момент часу.

Об’єктне сховище може зберігати велику к-сть документів та доступ до них є більш простим, оскільки воно має декілька незалежних вузлів, які зберігають дані та центральний вузол не має знати місцезнаходження об’єкта для того, щоб отримати його.

Ідентифікатори можна використовувати для того, щоб легко порівняти два файла без необхідності завантаження. Об’єктне сховище також дає можливість використання великої кількості, простого, більш дешевого апаратного забезпечення різних типів, пов’язуючи все разом у єдину програмну систему.

Кожен об’єкт, як правило, має кілька копій, можливо в географічно розподілених кластерах. Кожен об’єкт також містить контрольну суму, яка дозволяє легко встановити пошкодження документів, у цьому випадку можлива генерація нової копії об’єкта для заміни пошкодженої.

Для досягнення економічності хмарної системи збереження слід почати із доступної СЗД. Складнощі та обмеження неефективних файлових систем, що лежать в основі традиційних масивів збереження *NAS* і *SAN*, можуть легко перемістити на другий план потенційного скорочення витрат, яке забезпечує хмарна СЗД. У таких системах ускладнюється адміністрування СЗД, масштабованість обмежується штучними межами ємності та відбувається вимушена прив’язка до одного постачальника, який пропонує дороге клієнтське обладнання. Об’єктна система збереження набагато краще підходить для хмарних інфраструктур. Замість використання складної та застарілої файлової системи, якою до того ж важко керувати, об’єктні системи збереження застосовують єдиний однорівневий адресний простір. Воно дозволяє автоматично направляти документи в правильну систему збереження, визначає життєвий цикл змісту та зберігає активні і архівні дані на одному рівні, забезпечуючи їх належний захист. Завдяки цьому об’єктна система збереження показує кращі результати. Вони обумовлені узгодженням важливості документів та вартості їх збереження, а також відсутністю великих витрат на управління для переміщення документів вручну на необхідний рівень. При цьому забезпечується необмежена масштабованість для підтримки можливості хмарної СЗД надавати ємність на вимогу.

Інші переваги об’єктної системи збереження:

* + об’єктна система збереження може працювати максимально ефективно на стандартному серверному обладнанні;
  + об’єктна система збереження забезпечує простий доступ до сховища з будь-якого розташування, в будь-який час і для будь-якого пристрою по протоколу *HTTP*;
  + хмарна СЗД зазвичай надається у вигляді додатку моделі «збереження документів як послуга» через *Internet*.

Тому використання *HTTP* в якості основного протоколу для доступу до об’єктним пулів збереження документів значно спрощує постачальникам хмарних систем збереження процес інтеграції систем збереження в їх пропоновані послуги.

Об’єктні сховища вже знайшли застосування у галузі охорони здоров’я, фінансів та індустрії розваг, на додаток до хмарних сервісів та зараз стають привабливими для ще більш широкої групи компаній, в той час як технічні, управлінські, а також архівні вимоги ввели попит на масивні сховища документів в межах *IT* відділів.

*Facebook*, *Instagram* та *Twitter* використовують дану технологію для збереження мільйонів користувальницьких фотографій кожен день. *Spotify* використовує її для збереження мільйонів музичних треків. *Dropbox* та інші сервіси зберігають мільйони завантажених документів в об’єктному сховищі, прихованому за звичним, що містить файли і папки, інтерфейсом.

Використовуючи об’єктне локальне файлове сховище, компанії зберігають повний контроль над безпекою та фізичним місцезнаходженням своїх документів, і в той же час проявляються ключові переваги хмари і об’єктного сховища такі, як масштабованість, ефективність, завадостійкість та простота.

Також локальне об’єктне сховище дозволяє значно знизити вартість збереження об’єкта, у порівнянні з публічним хмарним сховищем.

Іншою характерною особливістю збереження об’єкта є використання метадокументів, або документів про дані. Кожен об’єкт при зберіганні маркований не тільки своїм унікальним ідентифікатором, але також інформацією про дані в об’єкті. Наприклад, оцифрована пісня поміщена у об’єктне сховище може зберігатися із додатковими мета-даними про назву пісні, виконавця, альбомом, тривалістю, роком, тощо. Важливо відзначити, що метадані кожного об’єкта фіксуються та зберігаються із ним. Немає необхідності тримати їх в окремій централізованій базі документів, яка може некеровано збільшитись [22].

Роль метадокументів стає все більш суттєвою в той час, як їх використання дозволяє поглянути на дані у якості стратегічного активу різним людям починаючи від фахівців з аналізу документів та бізнес-аналітиків до посадових осіб та фінансових аудиторів. Метадані - це безцінний елемент у стратегії компаній, який дозволяє оцінити статус, місце розташування та володіння корпоративними даними. Метадані також грають важливу роль у трансформаційних робочих навантаженнях, таких, як інтелектуальний аналіз документів та складна аналітика, які перетворюють вихідні дані у матеріальні цінності, які можуть бути кількісно оцінені у таких показниках, як дохід, прибуток, частка ринку та задоволеність клієнтів. Відсутність структури управління метаданими часто призводить до таких проблем, як більш тривалий пошук у сховищах документів або завданнях електронного виявлення. Прикріплення метадокументів дозволяє здійснювати набагато складніший пошук документів у об’єктному сховищі, ніж у файловому, де ім’я файлу часто є єдиним ключем до вмісту документу. Використання метадокументів у якості частини платформи управління інформацією в масштабах компанії підвищує видимість та допомагає знизити ризик, виявляти нові можливості для бізнесу і краще зрозуміти економічну цінність документів.

Важливою характеристикою об’єктного сховища є те, що схема метадокументів може бути спроектована відповідно до предметної області, із урахуванням потреб користувачів файлових сховищ та додатків. Також якщо не всі документи мають однакову здатність для пошуку, але можливість додати теги дозволяє зберегти інформацію про зміст та призначення файлу. Метадані також можуть бути проаналізовані, щоб отримати інформацію про переваги користувачів та клієнтів.

Оптимізований процес додавання нових файлів у об’єктне сховище має наступний вигляд. Спочатку дані, які зберігаються через інтерфейс користувача поступають на проміжний сервер, який додає до об’єкта основні метадані системи та іншу інформацію, корисну для емуляції файлової системи. Отже, дані перехоплюються під час процесу збереження, їх вміст сканується та аналізується, і вся створена інформація додається до об’єкта в вигляді метадокументів.

## 2.3. Використання метаданих

Метадані - це дані про використання документів, які часто не записуються а лише запам’ятовуються працівниками компанії.

З тих пір, як дані мають велику цінність, метадані, що описують їх, стали одним із ключових активів сучасних підприємств, що грають велику роль в виявленні дійсного значення документів. Організований процес управління корпоративними метаданими може допомогти виділити цінну інформацію із неявних джерел. Бізнес метадані додають контекст до документів, зберігаючи інформацію про їх ролі в бізнес процесах із урахуванням предметної області та службові дані про стан файлів, такі як формат, місце розташування, інформацію про те, хто, коли і яким чином використовував документ.

Сховище метадокументів дозволяє зберігати всю інформацію про структуру контейнерів із цими організації у одному місці. Це надає велику кількість матеріалу для обробки та аналізу із подальшою оптимізацією бізнес процесів. Зазвичай підприємства витрачають велику кількість часу та коштів на дослідження та прийняття рішень щодо впровадження і управління новими структурами документів. Сховища метадокументів грають велику роль у організації великих обсягів неструктурованих документів, надаючи інфраструктуру для обробки метадокументів та складання бізнес глосарію.

Добре складений бізнес глосарій є незамінним ресурсом для надання даних про зв’язки та відносини між документами в базах документів, графах, моделях і т. д. Управління метаданими спрощує процес побудови бізнес словника у разі, якщо кількість визначень виростає до сотень або тисяч, спрощуючи взаємодію між різними організаційними структурами підприємства.

Фіксація і освоєння цих метадокументів в легко доступному каталозі може відкрити великі можливості для організації. Зокрема, каталог метадокументів може підвищити доступність документів компанії. Працівники можуть швидко та впевнено зібрати необхідні дані для аналізу, зрозуміти, як взаємодіють дані. Користувацькі інтерфейси дозволяють легко, для фахівців по роботі з даними, власникам і користувачам, отримати зручний доступ до каталогу, що може допомогти в організації спільної роботи та отримання загального розуміння джерел документів по всьому підприємству.

Метадані можуть зберігатися або всередині, у тому ж файлі , або зовні, у окремому файлі або полі із описом документів. Сховище документів зазвичай зберігає метадані окремо від документів, але воно може бути спроектовано для підтримки змішаних підходів до збереження метадокументів. Кожен варіант має свої переваги та недоліки.

Внутрішнє збереження означає, що метадані завжди переміщуються як частина описуваних ними документів. Таким чином, метадані завжди доступні із даними, та ними можна маніпулювати локально. Цей метод створює надлишковість та не дозволяє управляти всіма метадані системи в одному місці.

Це, можливо, збільшує цілісність, так як метадані легко змінюються щоразу, коли змінюються дані.

Відокремлене сховище метадокументів дозволяє локалізувати метадані для всього сховища, наприклад, в базі документів, для більш ефективного пошуку та управління. При такому підході, метадані можуть бути об’єднані із контентом, коли інформація передається, або можна вставляти веб посилання.

Метадані можуть бути збережені в бінарному вигляді або зручному для читання форматі. У другому випадку, наприклад, використовуючи формати *XML* або *JSON*, користувачі можуть зрозуміти та редагувати його без спеціальних інструментів. З іншого боку, ці формати рідко оптимізовані для ємності, часу передачі, та швидкості обробки документів. Бінарні метадані позбавлені цих недоліків, але вимагають спеціальних бібліотек для перетворення двійковій даних у формат прийнятний для людини [23].

## 2.4. Висновки до розділу

У даному розділі були досліджені технології які використовуються для побудови сучасних систем збереження та швидкого доступу до електронних документів. Розглянуто основні типи фізичних моделей, які використовуються для побудови апаратних сховищ документів. Розглянуто використання прийомів віртуалізації для поділу програмних засобів управління розподіленим сховищем та апаратних засобів збереження документів.

В якості основи для побудови програмного продукту вибрані програмно-обумовлені сховища, оскільки вони реалізують більшу частину необхідного функціоналу із управлінням розподілу сховища і добре підходять для сформульованої задачі.

Розглянуто три основних типи моделей документів розподілених сховищ. Для реалізації поставленого завдання обрана об’єктна модель документів оскільки вона є оптимальною для вирішення завдання збереження великих обсягів неструктурованих документів.

Розглянуто можливості використання метадокументів для оптимізації роботи з сховищем. Можливість об’єктне сховище додавати метадані до файлів можна використовувати для структурування документів. Метадані файлів можна зберігати разом у вигляді пошукових індексів що дає можливість обробляти складні пошукові запити.

**РОЗДІЛ 3  
ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ**

**3.1. Проектування архітектури**

Під архітектурою програмної системи розуміється уявлення, яке т інформацію про компоненти складають систему, про взаємозв’язки між цими компонентами і правилах, що регламентують ці взаємозв’язки. Програмний продукт являє собою систему з трьох основних елементів, яка зображена на рисунку 3.1:

1. Проміжний сервер;
2. Файлове сховище;
3. Пошукова система.

Взаємодія між компонентами системи здійснюється з використанням протоколів *TCP / IP* на транспортно-мережевому рівні і *HTTP* на прикладному рівні.

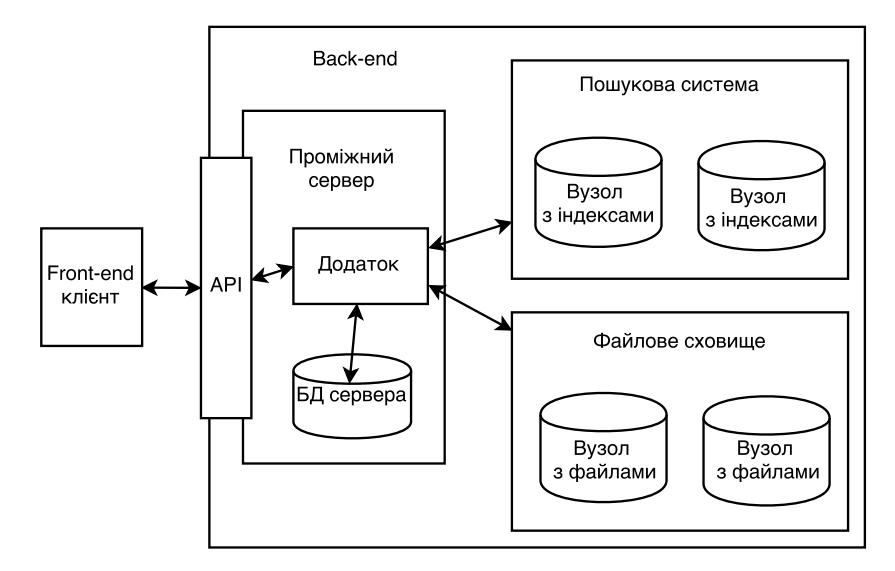


Рис. 3.1. Загальна архітектура системи

Проміжний сервер містить серверний додаток, який пов’язує клієнтську програму та інші компоненти архітектури. Для збереження його сутностей сервер містить базу документів. Додаток надає *API* для клієнтського додатку та містить модулі-клієнти для відправлення запитів до *API* сховища і пошукової системи. Також воно надає функціонал адміністрування системи, реєстрації і авторизації користувачів, створення та збереження схем метадокументів.

Файлове сховище являє собою кластер, на якому розгорнуто об’єктне сховище. Надає *API* для додавання, зміни і видалення об’єктів і адміністрування системи.

Пошукова система являє собою кластер, на якому розгорнута пошукова система, що зберігає індекси файлів сховища і їх метадокументів і надає *API* для пошуку і індексації об’єктів і адміністрування системи.

**3.2. Опис *API***

*API* (скор. Від англ. *Application Programming Interface* - «прикладний програмний інтерфейс») - набір функцій, який визначає спосіб взаємодії користувача з програмною системою і надає таку функціональність:

1. Робота з користувачами;
2. Робота з файлами: створення, видалення, перегляд папок, додавання, видалення і зміна файлів;
3. Робота з метаданими: додавання, редагування, видалення тегів;
4. Рбробка пошукових запитів;

## 3.3. Середовище розробки програмного продукту

Для розробки системи збереження і доступу до документів було використано *IDE Microsoft Visual Studio Ultimate 2012* (див. рис. 3.2).

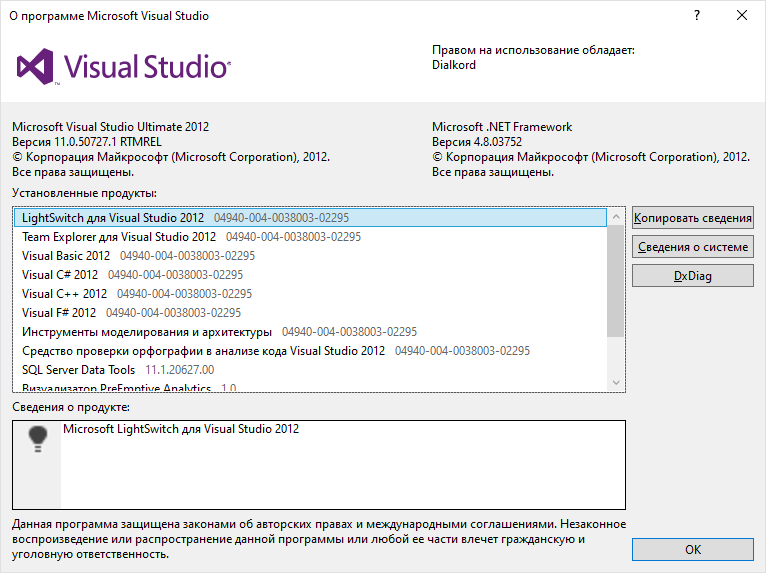


Рис. 3.2. Про програму *Visual Studio* *2012*

*Microsoft Visual Studio* - серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів (див. рис. 3.3). Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології *Windows Forms*, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для усіх платформ, що підтримуються *Microsoft Windows*, *Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight*.

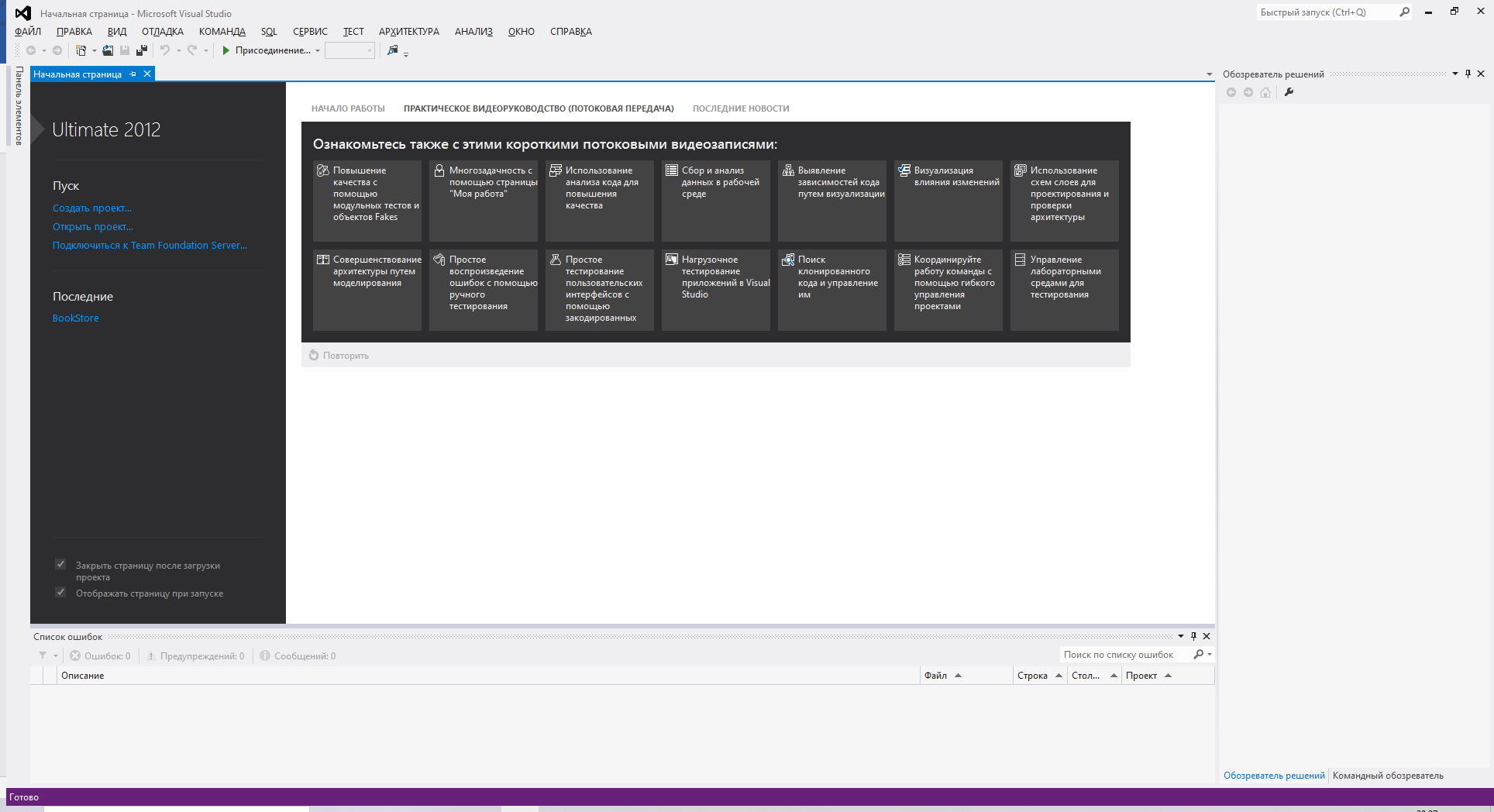


Рис. 3.3. Зовнішній вигляд середовища розробки *Visual Studio* *2012*

## 3.4. Робота з базою документів

При першому відкритті програми, користувач бачить вікно створення бази документів (рис. 3.4).

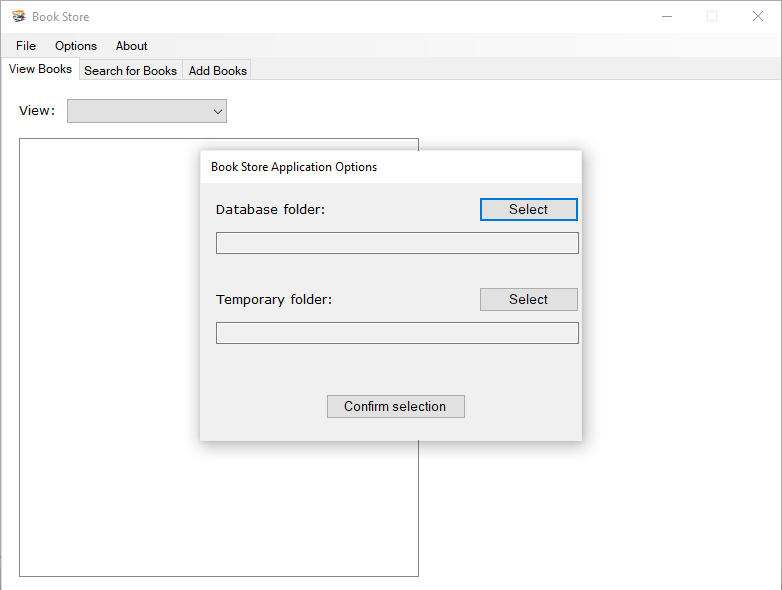


Рис. 3.4. Вікно створення нової бази документів

Користувач вибирає папку на своєму персональному комп’ютері, де він хоче зберігати файл бази документів (див. рис. 3.5).

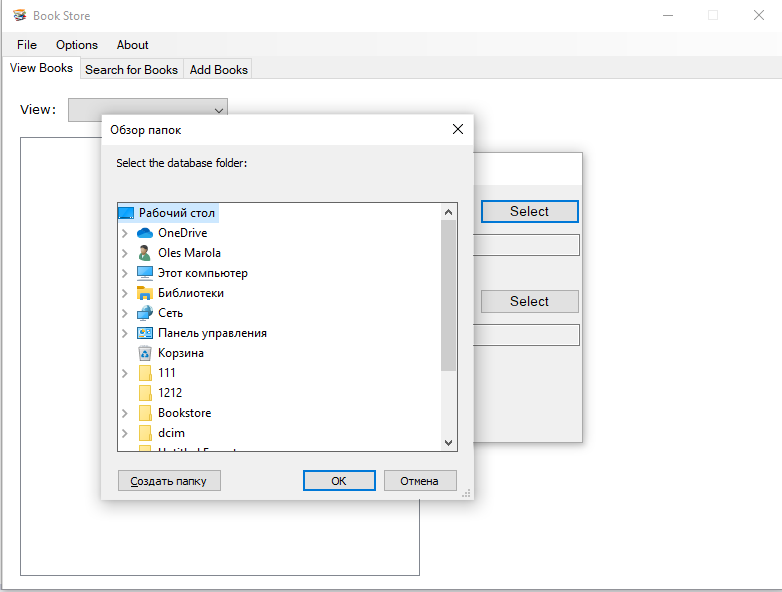


Рис. 3.5. Вибір папки для створення в ній нової бази документів

За створення нової бази документів відповідає функція *btSelectDatabaseFolder*, код якої наведений нижче:

*private void btSelectDatabaseFolder\_Click(object sender, EventArgs e)*

*{*

*string message = String.Format("{0}{1}",*

*"Please note that the database folder must be on a drive ",*

*"with enough free space to store the application data.");*

*MessageBox.Show(message,*

*"Database Folder Information",*

*MessageBoxButtons.OK,*

*MessageBoxIcon.Information);*

*FolderBrowserDialog folderBrowser = new FolderBrowserDialog();*

*folderBrowser.ShowNewFolderButton = true;*

*folderBrowser.Description = "Select the database folder:";*

*DialogResult folderDialogResult = folderBrowser.ShowDialog();*

*if (folderDialogResult == DialogResult.Cancel)*

*return;*

*string completePath = folderBrowser.SelectedPath;*

*try*

*{*

*DirectoryInfo dirInfo = new DirectoryInfo(completePath);*

*dirInfo.GetDirectories();*

*}*

*catch (Exception)*

*{*

*MessageBox.Show("The folder selected cannot be accessed.",*

*"Invalid Database Folder",*

*MessageBoxButtons.OK,*

*MessageBoxIcon.Error);*

*return;*

*}*

*databaseFolderPath = completePath;*

*tbDatabaseFolder.Text = databaseFolderPath;*

*}*

Для правильної роботи програми також необхідно створити тимчасову базу документів, операції в якій будуть проходити під час роботи програми, а потім при закритті програми, усі документи, з якими проводились зміни, будуть додані до основної бази документів [24].

## 3.5. Головне вікно програми

Програма має назву «*Book Store*» (з англ. «Сховище документів (звітів)»). Головне вікно програми має вигляд (рис. 3.6):

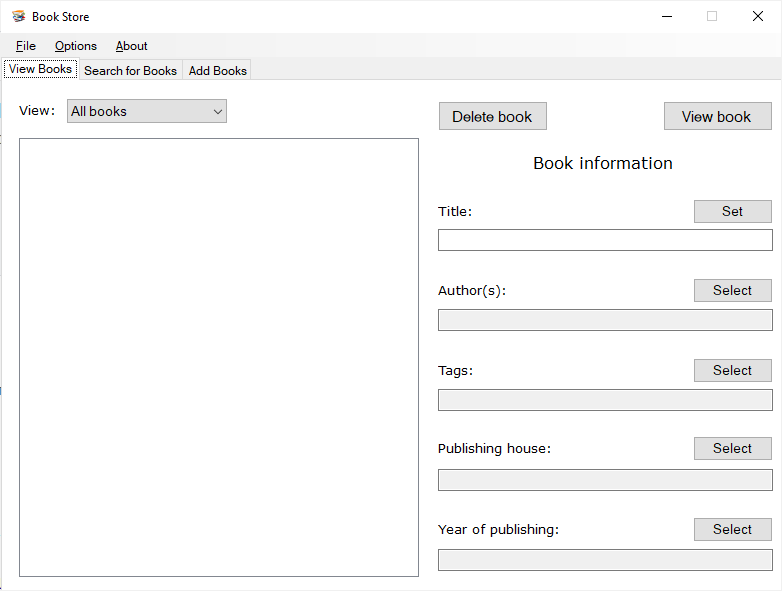


Рис. 3.6. Головне вікно програми

Головне вікно програми складається з трьох основних «табів»:

* переглянути усі документи (див. рис. 3.7);
* шукати документи (див. рис. 3.8);
* додати документи (див. рис. 3.9).

Також є деякі опції, такі як видалити існуючу базу документів при наступному запуску програми, дізнатися інформацію про версію програми та її автора, можливість дефрагментувати базу документів, для єкономії дискового простору [25].

## 3.6. Перегляд документів

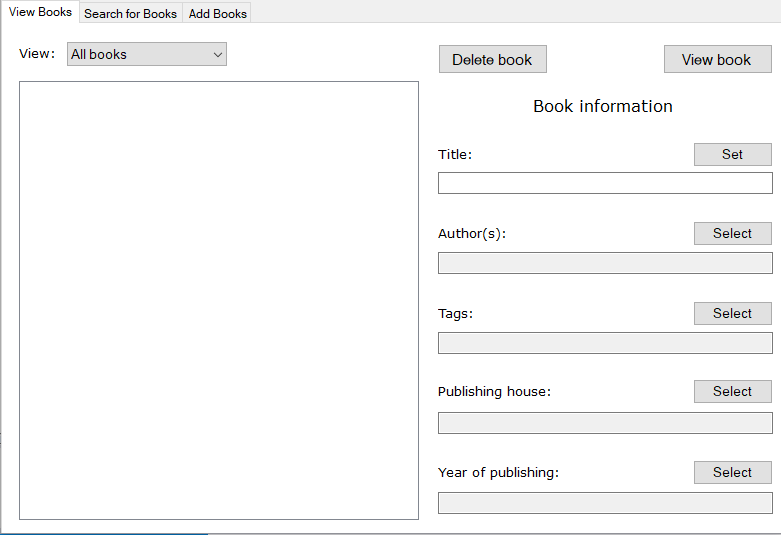


Рис. 3.7. Вікно перегляду документів

Для перегляду додокументів до бази документів документів використвується ф-ція *private void btViewBook*, код якої наведений нижче:

*private void btViewBook\_Click(object sender, EventArgs e)*

*{*

*if (book != null)*

*new BackgroundTask("Displaying book ... ", DisplayBook);*

*}*

Ця фунція дозволяє відкрити доданий до програми файл у текстовому редакторі за замовчуванням.

Також існує можливість додати додаткову інформацію про документ (группу документів). Для цього використовується функція *btSetBook\*\*\* ,* де \*\*\* - це назва параметру, який необхідно встановити [26]. Усі функції виду *btSetBook\*\*\** мають аналогічну реалізацію. Для прикладу наведемо лістинг функції *btSetBookTitle:*

*private void btSetBookTitle\_Click(object sender, EventArgs e)*

*{*

*if (book != null)*

*{*

*tbTitle.Text = tbTitle.Text.Trim();*

*if (tbTitle.Text != book.Title)*

*{*

*if (tbTitle.Text != string.Empty)*

*{*

*book.Title = tbTitle.Text;*

*booksListOperations.ChangeBookTitle();*

*}*

*else*

*tbTitle.Text = book.Title;*

*}*

*}*

*}*

Після заповнення усієї необхідної інормації, вікно програми має вигляд (рис. 3.8):

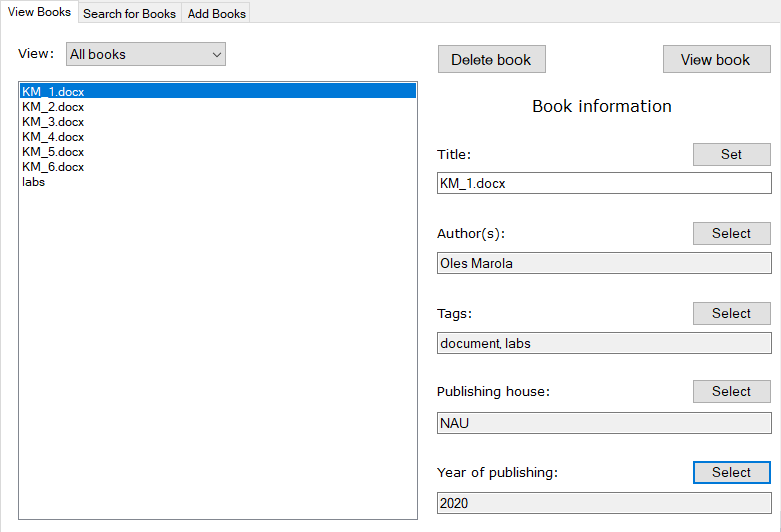


Рис. 3.8. Головне вікно програми після додавання даних про документ

## 3.7. Додавання нового документа до бази документів

Вікно додавання нового документа має вигляд (рис. 3.9):

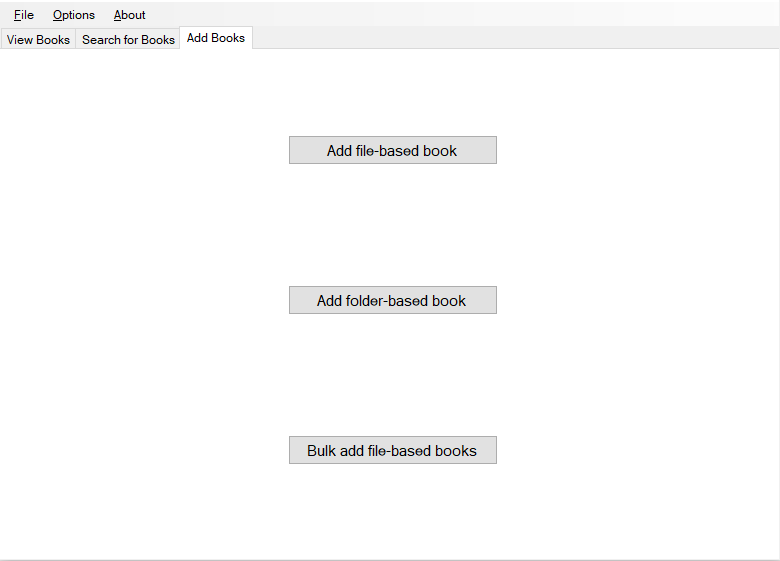


Рис. 3.9. Вікно додавання нового документа

Існує декілька можливостей додати документи до бази документів. Перший - це додати один файл до бази документів (кнопка *Add file-based book*). Лістинг функції, яка відповідає за додавання файлу до бази документів наведена нижче:

*private void btAddFileBasedBook\_Click(object sender, EventArgs e)*

*{*

*OpenFileDialog fileDialog = new OpenFileDialog();*

*fileDialog.CheckPathExists = true;*

*fileDialog.CheckFileExists = true;*

*fileDialog.Filter = String.Format("{0}|{1}|{2}|{3}|{4}|{5}|{6}|{7}|{8}",*

*"All files (\*.\*)|\*.\*",*

*"Compiled HTML files (\*.chm)|\*.chm",*

*"HTML files (\*.htm, \*.html)|\*.htm;\*.html",*

*"Microsoft PowerPoint files (\*.ppt, \*.pptx, \*.pps)|\*.ppt;\*.pptx;\*.pps",*

*"Microsoft Word files (\*.doc, \*.docx)|\*.doc;\*.docx",*

*"Open Document Format files (\*.odf)|\*.odf",*

*"Rich Text Format files (\*.rtf)|\*.rtf",*

*"Portable Document Format files (\*.pdf)|\*.pdf",*

*"Text files (\*.txt)|\*.txt");*

*fileDialog.InitialDirectory = Properties.Settings.Default.BooksFolder;*

*fileDialog.RestoreDirectory = true;*

*fileDialog.SupportMultiDottedExtensions = true;*

*fileDialog.Title = "Select the book file:";*

*DialogResult fileDialogResult = fileDialog.ShowDialog();*

*if (fileDialogResult == DialogResult.Cancel)*

*return;*

*ExtractPathInfo(fileDialog.FileName, out rootPath, out path);*

*Properties.Settings.Default.BooksFolder = rootPath;*

*new BackgroundTask("Adding the book to the database ... ",*

*AddFileBasedBook);*

*PopulateViewTab();*

*}*

Аналогічним чином до бази документів можна додати папку з документами, яка буде зберігатись як папка, і відкриватись як папка у провіднику операційної системи (див. рис. 3.10).

Це дозволяє користувачам більш гнучко користуватися системою, та надає можливість вибору [27].

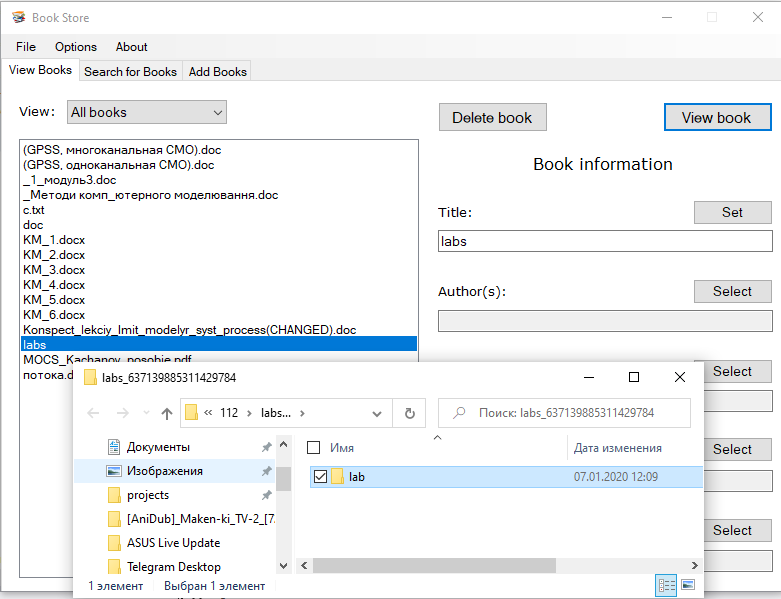


Рис. 3.10. Вікно перегляду папки, яка збережна у базі документів

## 3.8. Видалення документа з бази документів

Для того, щоб видалити документ/папку з документами з бази документів, була реалізована функція *btDeleteBook*, лістинг якої наведено нижче:

*private void btDeleteBook\_Click(object sender, EventArgs e)*

*{*

*if (book != null)*

*{*

*string deleteConfirmMessage;*

*if (book.Authors.Count > 0)*

*deleteConfirmMessage = String.Format("{0}{1}{2}{3}{4}",*

*"Are you sure you want to delete the book \"",*

*book.Title,*

*"\", written by \"",*

*book.Authors[0],*

*"\" from the database?");*

*else*

*deleteConfirmMessage = String.Format("{0}{1}{2}",*

*"Are you sure you want to delete the book \"",*

*book.Title,*

*"\" from the database?");*

*DialogResult result = MessageBox.Show(*

*deleteConfirmMessage,*

*"Confirm Book Deletion",*

*MessageBoxButtons.YesNoCancel,*

*MessageBoxIcon.Question*

*);*

*if (result == DialogResult.Yes)*

*{*

*new BackgroundTask("Deleting book ... ", DeleteBook);*

*booksListOperations.DeleteBook();*

*ClearDataAndFields();*

*book = null;*

*}*

*}*

*}*

Вікно видалення документа зображено на рисунку 3.11:

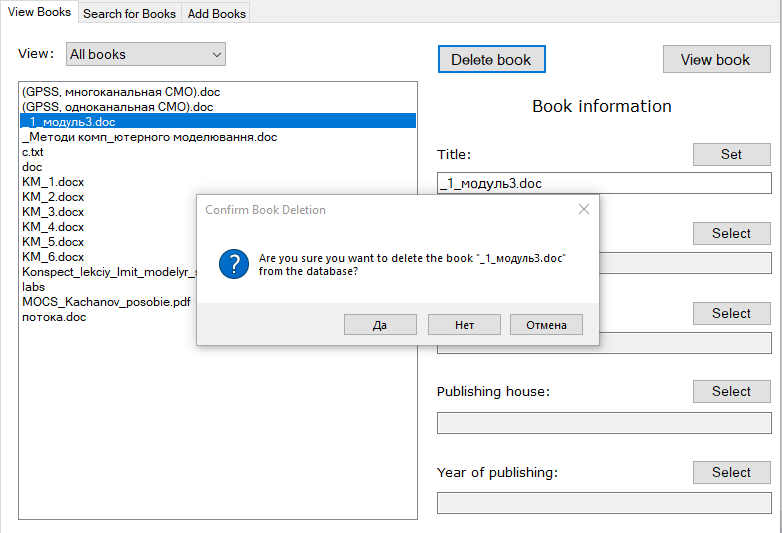


Рис. 3.11. Вікно видалення документа з бази документів

## 3.9. Організація швидкого доступу до документів

Для організації швидкого доступу до документів було використано механізми гнучкого пошуку, який дозволяє організувати відображення документів, які підпадають під умови пошуку [28]. Вікно пошуку по категоріям зображено на рисунку 3.12:



Рис. 3.12. Вікно пошуку по категоріям

Після вибору декількох параметрів вікно матиме вигляд (рис. 3.13):

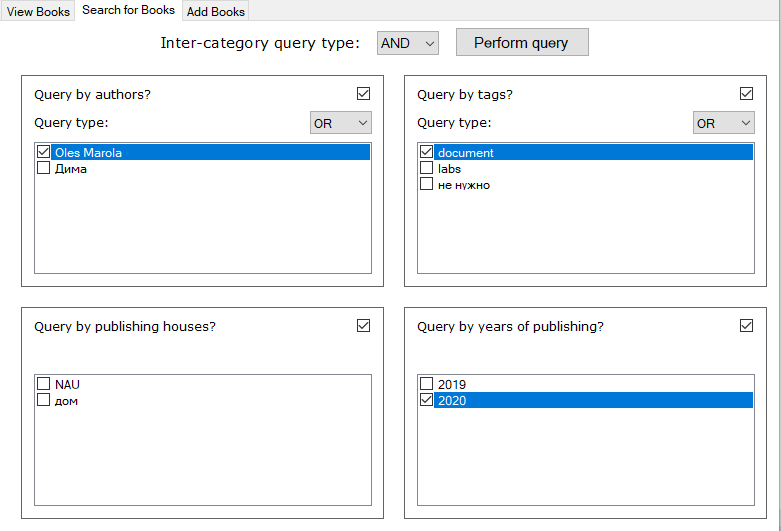


Рис. 3.13. Вікно пошуку після вибору параметрів

Після натискання кнопки *Perform query* відбувається пошук у системі, після чого у каталозі зміниться пункт *View* і відобразяться документи, які відповідають критеріям пошуку (рис. 3.14). Функція, яка виконує пошук по документам має наступну реалізацію:

*public class ORQueryType : IQueryType*

*{*

*public List<Book> PerformQuery(List<List<Book>> books)*

*{*

*List<Book> requestedBooks = new List<Book>();*

*foreach (List<Book> currentBook in books)*

*requestedBooks = SetOperations<Book>.Union(requestedBooks, currentBook);*

*return requestedBooks;*

*}*

*}*

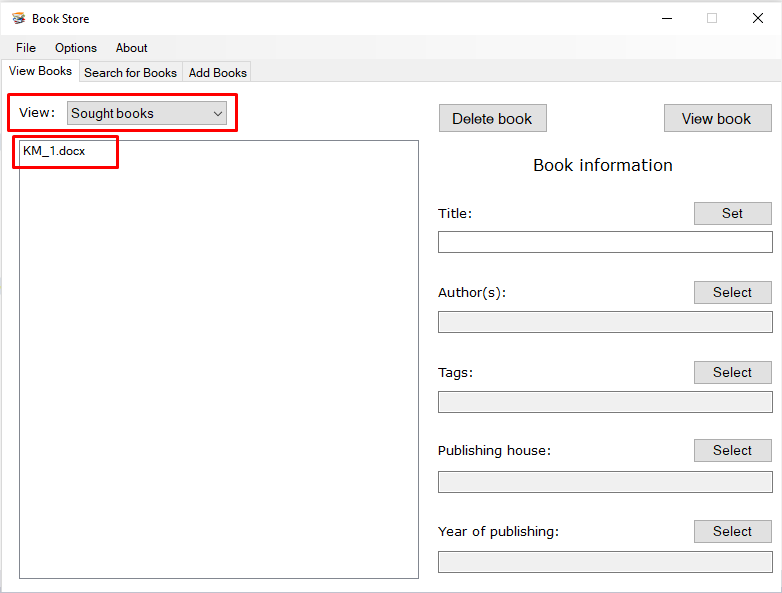


Рис. 3.14. Головне вікно програми з результатами пошуку

## 3.10. Висновки до розділу

У третьому розділі описано структуру та роботу розробленої системи збереження та швидкого доступу до електронних документів.

Основною перевагою розробленої системи є висока захищеність даних, так як вона зберігається локально і доступ до неї може отримати лише розроблена система. Також легкість використання та швидкість роботи системи каже про те, що вона може бути використана на будь-яких персональних комп’ютерах на операційній системі *Windows*. Система також не потребує доступу до мережі Інтернет, що додає до системи певних переваг (за умови, якщо база документів не була створена на видаленому сервері).

Описана робота системи, основні функції, які були використані при написанні додатку.

Представлено великий список кількісних характеристик, а саме:

* можливість створення бази документів в одному файлі локально;
* можливість додавання одного документа;
* можливість додавання папки з документами;
* можливість додавання групи документів;
* можливість перегляду документів по категоріям;
* можливість додавання додаткової даних для подальшого швидкого доступу до документів;
* можливість видалення документів;
* можливість пошуку по документах;
* можливість швидко відобразити потрібні категорії.

Приведено короткий опис, як необхідно користуватися розробленою системою. Отже можна зробити висновок, що дана система є унікальною та доступною. Якщо взяти до уваги рішення конкурентів, то вони, переважно, пропонують свої рішення за деякі кошти або за підпискою.

Розробивши систему і провівши деякі самостійні тести стабільності ми можемо вважати розробку даної системи доцільною.

# **ВИСНОВКИ**

Метою дипломної роботи є розробити систему збереження документів користувачів для швидкого та зручного отримання доступу до електронних документів. В першу чергу була розглянута проблема організації роботи з документами. Сформульовані характеристики якими повинен володіти інструмент збереження та обміну файлами для користувачів.

Розглянуті способи збереження файлів. Можна зробити висновок, що персональні засоби збереження документів, такі як жорсткі диски і флеш накопичувачі не відповідають вимогам організацій. Ця проблема потребує більш комплексного рішення, яке інтегрує дані користувача у єдину систему, яка повинна мати функціонал структурування та захисту великих обсягів документів.

Досліджені відкриті програмні рішення не відповідають сформульованим вимогам. Організації постачальники пропонують платні програмні системи, спеціалізовані на рішенні даної проблеми, які не підходять малим компаніям або одиночним користувачам, які не мають достатньої кількості ресурсів.

Також були досліджені технології, які використовуються для побудови сучасних систем збереження та швидкого доступу до даних. Розглянуто основні типи фізичних моделей які використовуються для побудови апаратних сховищ документів. Розглянуто використання прийомів віртуалізації для поділу програмних засобів управління розподіленим сховищем та апаратних засобів збереження документів.

В якості основи для побудови програмного продукту вибрані програмно-визначені сховища оскільки вони реалізують більшу частину необхідного функціоналу із управління розподілом сховищем та добре підходять для поставленої задачі.

Розглянуто 3 основних типи моделей документів розподілених сховищ. Для реалізації поставленої задачі обрана об’єктна модель документів оскільки вона є оптимальною для вирішення завдання збереження великих обсягів неструктурованих документів.

Розглянуто можливості використання метадокументів для оптимізації роботи із сховищем. Можливість об’єктного сховища додавати метадані до файлів можна використовувати для структурування документів. Метадані файлів можна зберігати разом у вигляді пошукових індексів, що дає можливість обробляти складні пошукові запити.

Як результат було спроектовано архітектуру програмної системи. Була проведена об’єктна декомпозиція та представлення взаємозв’язків між підсистемами.

Було розглянуто методи та алгоритми реалізації пошуку документів для подальшого швидкого доступу до них.

Описана робота системи, основні функції, які були використані при написанні додатку.

Представлено великий список кількісних характеристик, а саме:

* створення бази документів в одному файлі локально;
* додавання одного документа;
* додавання папки з документами;
* додавання групи документів;
* перегляд документів по категоріям;
* додавання додаткової даних для подальшого швидкого доступу до документів;
* видалення документів;
* пошук по документах;
* можливість швидко відобразити потрібні категорії.

Приведено короткий опис, як необхідно користуватися розробленою системою.

В результаті роботи був реалізований прототип програмної системи. Реалізовану програмну систему можна використовувати для вирішення задач збереження та обміну файлами як одиночним користувачам, так і малим підпиємствам.

# **СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Браун М. Методы поиска информации в Интернете./ М. Браун – М., 2005.

– 344 с.

1. Головач В.В. Дизайн пользовательского интерфейса. / Головач В.В. – М. : Наука, 2003. – 752 с.
2. Когаловский М. Р. Перспективные технологии информационных систем. / М. Р. Когаловский. – М. : ДМК Пресс, 2003. – 288 с.
3. Маннинг К. Введение в информационный поиск./ К.Маннинг, П. Рагхаван, Ч. Шютце. . – М.:, 2011. – 528 с.
4. *Microsoft Visual Studio* [Електронний ресурс]. – Режим доступу до ресурсу: *https://uk.wikipedia.org/wiki/Microsoft\_Visual\_Studio*
5. Створення сховищ документів. Технології *OLAP* та *Data Mining* [Електронний ресурс]. – Режим доступу до ресурсу: *https://pidruchniki.com/ olap\_data\_mining/*
6. Сховище документів [Електронний ресурс]. – Режим доступу до ресурсу: *https://pidruchniki.com/74246/informatika/shovische\_danih*
7. Принцип работы поисковых систем [Електронний ресурс]. – Режим доступу до ресурсу: *https://works.doklad.ru/view/4yqZXQ0Pous.html*
8. Сховище документів [Електронний ресурс]. – Режим доступу до ресурсу: *https://uk.wikipedia.org/wiki/%D0%A1%D1%85%D0%BE*
9. Сховища документів, архітектура, моделі та основи їх створення [Електронний ресурс]. – Режим доступу до ресурсу: *https://studopedia.su/16\_177900\_shovishcha-danih-arhitektura-modeli-ta-osnovi-ih-stvorennya.html*
10. Технологія сховищ документів *Data Warehousing* [Електронний ресурс]. – Режим доступу до ресурсу: *https://studfile.net/preview/5118185/page:31/*
11. Порівняння і синхронізація дозволів баз документів *SQL Server* [Електронний ресурс]. – Режим доступу до ресурсу: *https://autosyncdb.com/ua/porivnyannya-i-synkhronizatsiya-dozvoliv-baz-danykh-sql-server/*
12. Аналіз найактуальніших серверних систем управління базами документів [Електронний ресурс]. – Режим доступу до ресурсу: *http://vlp.com.ua/files/14\_22.pdf*
13. Порiвняння *sql* та *nosql* баз документів [Електронний ресурс]. – Режим доступу до ресурсу: *http://www.tech.vernadskyjournals.in.ua/journals/2018/6\_2018/part\_2/7.pdf*
14. *Kapadia, A. Implementing cloud storage with Openstack swift: Design, implement, and successfully manage your own cloud storage cluster using the popular OpenStack swift software. / Kapadia, A., Varma, S., Rajana, K. // United Kingdom: Packt Publishing*, 2014 – pp. 134 – 136.
15. *Arnold, J. OpenStack swift: Using, administering, and developing for swift object storage. / Arnold, J. // O’Reilly Media*, 2014 – *pp*. 254 – 256.
16. *Gormley, Elasticsearch: The definitive guide. / Gormley, C., Tong, Z.*
17. *Toigo, J.W. The holy Grail of enterprise data storage. / Toigo, J.W. // Prentice-Hall,* 1999 – *pp*. 64 – 69.
18. *Ali Babar. Guidelines for Building a Private Cloud Infrastructure. / Ali Babar. // IT-Universitetet i København*, 2012 – *pp*. 274 – 276.
19. *Girish L S. Building Private Cloud using OpenStack. / Girish L S., Dr. H. S. Guruprasad. // International Journal of Emerging Trends & Technology in Computer Science*, 2014 – *pp*. 134 – 138.
20. *Marc Farley. Rethinking Enterprise Storage: A Hybrid Cloud Model. / Marc Farley. // Microsoft Press*, 2013 – *pp*. 24 – 28.
21. *Kapadia, Amar, Kris Rajana, Sreedhar Varma. OpenStack Object Storage (Swift) Essentials. / Kapadia, Amar, Kris Rajana, Sreedhar Varma // Packt Publishing*, 2015 – *pp*. 184 – 187.
22. *Beach Daniel. Building a Search Server with Elasticsearch. / Beach Daniel. // Packt Publishing*, 2015 – *pp*. 267 – 269.
23. *Metz, Sandi. Practical Object-oriented Design in Ruby: An Agile Primer. / Metz, Sandi. // Addison-Wesley*, 2016 – *pp*. 150 – 154.
24. *Hartl, Michael. Ruby on Rails Tutorial: Learn Web Development with Rails. / Hartl, Michael. //Addison-Wesley*, 2016 – *pp*. 194 – 196.
25. Хорстманн, Кей С., Корнелл, Гарi. *Java 2* Бібліотека професіонала, том 1. Основи. 8-е видання.: Видавничий дім «Вільямс», *ISBN 978-5-8459-1378-4,* 2012. – 813 с.
26. Хорстманн, Кей С., Корнелл, Гарi. *Java 2* Бібліотека професіонала, том 2. Тонкощі програмування. 8-е видання.: Видавничий дім «Вільямс», *ISBN 978-5-8459-1378-4,* 2012. – 813 с.
27. «*W3schools*» [Електронний ресурс] – Режим доступу до ресурсу: *https://www.w3schools.com /xml/xpath\_syntax.asp*
28. ДСТУ 3008-95: Документація. Звіти у сфері науки і техніки Структура і правила оформлення
29. ДСТУ 8302-2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».
30. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.