

Лекція 15

Використання технології зберігання даних та обміну даними

Спосіб управління даними, прийшов довгий шлях протягом останніх декількох десятиліть. Сьогоднішні користувачі отримують багато переваг, які можна отримати завдяки використанню баз даних. Однак, не так давно комп'ютери спиралися на набагато менш елегантний і дорогий підхід до управління даними, який називається файловою системою.

15.1 Методи побудовані на файловій системі

Одним із способів збереження інформації на комп'ютері є зберігання її в постійних файлах. Система компанії має ряд прикладних програм; кожна з них призначена для маніпулювання файлами даних. Ці прикладні програми задовольняють вимогам користувачів, які є співробітниками організації. У міру виникнення нових потреб в інформаційну систему додаються нові застосунки. Описана система називається файловою системою.

Розглянемо традиційну банківську систему, яка використовує файлову систему для управління даними організації, показаними на рисунку 15.1. Як бачимо, у банку є різні відділи. Кожен має свої власні програми, які керують і маніпулюють різними файлами даних. Для банківських систем програми можуть бути використані для дебетування або кредитування рахунку, пошуку залишку на рахунку, додавання нового іпотечного кредиту і генерування щомісячних виписок.

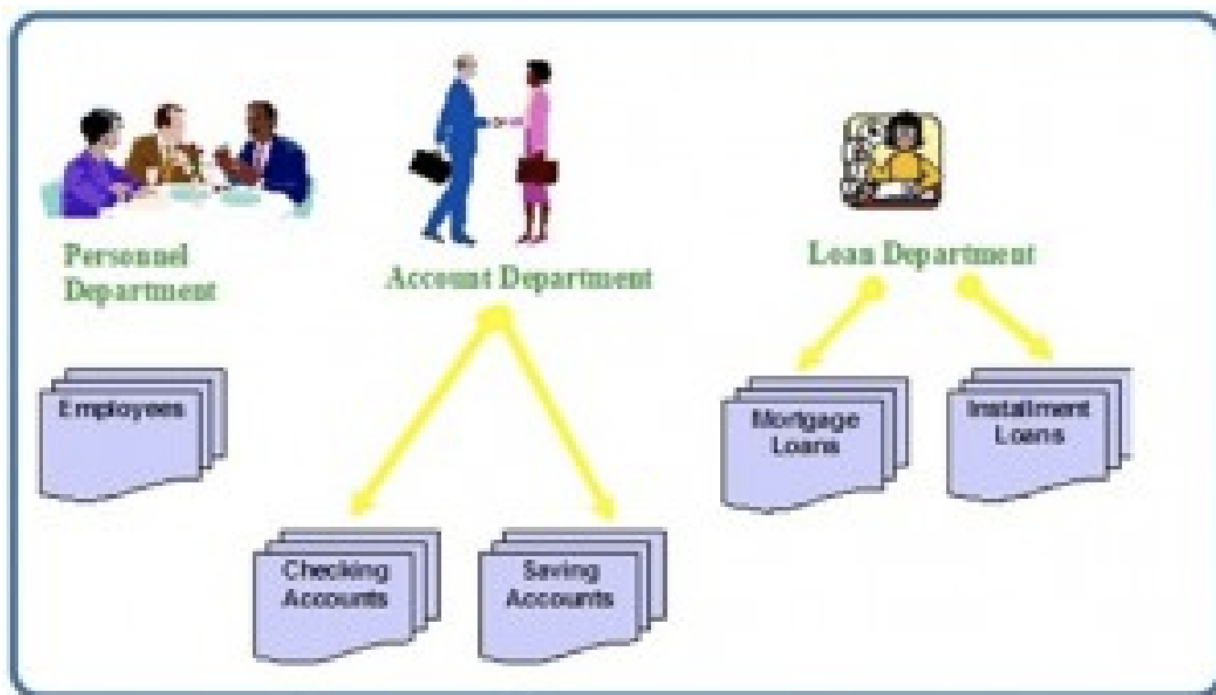


Рисунок 15.1. Приклад файлової системи, що використовується банками для управління даними.

15.2 Недоліки файлового підходу

Використання файлової системи для збереження організаційної інформації має ряд недоліків. Нижче наведено п'ять прикладів.

1 Резервування даних: часто в межах організації, файли і програми створюються різними програмістами з різних відділів протягом тривалого часу. Це може призвести до надлишковості даних, ситуації, яка виникає в базі даних, коли поле потрібно оновити більш ніж в одній таблиці. Така практика може призвести до декількох проблем, таких як:

- Невідповідність у форматі даних
- Одна і та ж інформація, що зберігається в декількох різних місцях (файлах)
- Невідповідність даних, ситуація, коли різні копії одних і тих же даних конфліктують, витрачається місце зберігання і дублюються ресурси

2 Ізольованість даних – властивість, яка визначає, коли і як зміни, внесені однією операцією, стають видимими для інших користувачів та інформаційної систем. Ця проблема виникає в умовах паралелізму, через те, що для нових програм важко отримати відповідні дані, тому що вони можуть бути збережені в різних файлах.

3 Проблема з цілісністю даних є ще одним недоліком використання файлової системи. Це стосується підтримки і впевненості в тому, що дані в базі даних правильні та послідовні. Факторами, які слід враховувати під час вирішення цього питання, є:

- Значення даних повинні задовольняти певним обмеженням узгодженості, які вказуються в програмах-застосунках.
- Важко вносити зміни в прикладні програми для того, щоб забезпечити виконання нових вимог, які виникають через зміни умов функціонування установи.

4 Безпека може бути значною проблемою за файлового підходу управління даними, оскільки:

- Існують обмеження щодо доступу до привілеїв.
- Вимоги до застосування додаються до системи таким чином, що важко забезпечити обмеження повноважень.

5 Паралельний доступ – це можливість бази даних дозволити декільком користувачам доступ до одного запису без несприятливого впливу на обробку транзакцій. Файлова система повинна управляти або перешкоджати, паралелізму прикладних програм. Зазвичай, у файловій системі, коли програма відкриває файл, цей файл заблоковано для інших програм. Це означає, що більше ніхто не має доступу до файлу в той самий час.

У системах баз даних управління паралельним доступом дозволяє отримати доступ до одного запису декільком користувачам одночасно. Це важлива відмінність між базою даних і файловими системами.

15.3 Методи побудовані на базах даних

Труднощі, що виникають за використання файлової системи, спонукали до розробки нового підходу в управлінні великими обсягами організаційної інформації – управління даними за допомогою баз даних.

Бази даних і технології розробки та ведення баз даних відіграють важливу роль в більшості областей, де використовуються комп'ютери, включаючи бізнес, освіту і медицину. Щоб зрозуміти основи систем баз даних, розглянемо деякі базові поняття в цій області.

Роль баз даних у бізнесі

Кожен якимось чином використовує базу даних, навіть якщо просто зберігати інформацію про своїх друзів або родину. Ці дані можуть бути записані або збережені на комп'ютері за допомогою текстових редакторів або можуть бути збережені в електронній таблиці. Проте найкращим способом зберігання даних є використання програмного забезпечення для управління базами даних. Це потужний програмний інструмент, який дозволяє зберігати, маніпулювати і отримувати дані різними способами.

Більшість компаній стежать за інформацією про клієнтів, зберігаючи її в базі даних. Ці дані можуть включати клієнтів, співробітників, продукти, замовлення або будь-що інше, що допомагає бізнесу в його роботі.

Значення даних

Дані є фактичною інформацією, такою як вимірювання або статистика про об'єкти і поняття. Можна використовувати дані для дискусій як факти або як результати обчислень. Даними можуть бути людина, місце, подія, дія або будь-що з множини речей реального світу. Одиничний факт є елементом даних.

Дані є інформацією, яка має форму прийнятну для обробки засобами інформатизації. Інформація – це опис частини реального світу, в якій ми працюємо. Тому важливим питанням є питання

правильного зберігання даних для їх подальшої ефективної обробки. Дані можуть зберігатися в:

- Картотеках
- Електронні таблиці
- Папках
- Бухгалтерських книжках
- Списках
- Купі паперів на вашому столі

Всі ці елементи зберігають інформацію, і так само складають базу даних. Через автоматизовану природу баз даних, вони мають приголомшливу потужність управління і обробки інформації, яку вони зберігають. Це може зробити інформацію, яку зберігають бази даних набагато кориснішою для роботи.

База даних – це єдиний набір пов'язаних даних, що використовується для підтримки діяльності певної організації. Базу даних можна розглядати як сховище даних, яке визначається один раз, а потім доступ до нього мають різні користувачі, як показано на рис.15.2.

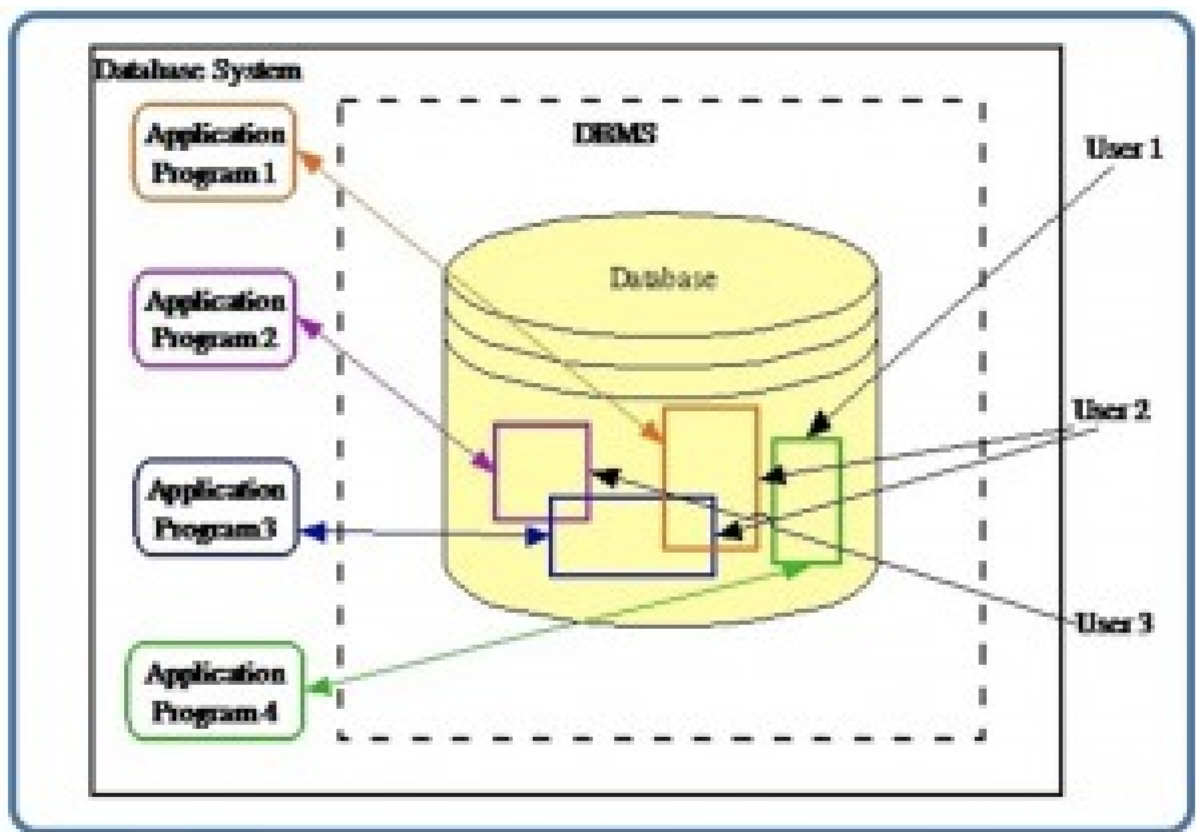


Рисунок 15.2. База даних як сховище даних

База даних має такі властивості:

- Це представлення деякого аспекту реального світу або сукупність елементів даних (фактів), що представляють інформацію про частину реального світу.
- База даних є логічною, однорідною та внутрішньо послідовною.
- Спроектowana, побудована та заповнена даними база даних є інструментом досягнення конкретної мети.
- Кожен елемент даних зберігається в певному полі бази даних.

Комбінація полів складає таблицю. Наприклад, кожне поле в таблиці співробітників містить дані про окремого працівника. База даних може містити багато таблиць. Наприклад, система про членів певної організації може містити таблицю адрес та інших особистих даних про

окремих членів, як показано на рис.2.2. Членами організації можуть бути фізичні особи, групи осіб, фірми та корпорації, які приймають участь в роботі в організації. Членство можна придбати за одно- або дворічний період, а потім поновити ще на один одно- або дворічний період. Приклад для реєстрації участі в певній організації наведено на рисунку 15.3.

Membership

ID EXPIRY DATE Prev Exp Stat Cat

Name Res

Address

City Prov Country

Notes

Cards # Members #Years

First Name	Last Name	YYMM	G	BARCODE	V	DATE	TIME	F
Mickey	Mouse	0000	M	10000001	4	20130810	10:12:29	y
Minnie	Mouse	0000	F	10000002	4	20130810	10:12:29	y
Mighty	Mouse	0000	M	10000003	4	20130810	10:12:29	y
Door	Mouse	0000	F	10000004	4	20130810	10:12:29	y
Tom	Mouse	0000	M	10000005	4	20130810	10:12:29	y
King	Rat	0000	M	10000006	4	20130810	10:12:29	y
Man	Mouse	0000	M	10000007	4	20130810	10:12:29	y
Moose	Mouse	0000	M	10000008	4	20130810	10:12:29	y

Record: 1 of 1 No Filter Search

Рисунок 15.3. Таблиця реєстрації участі в певній організації

Система управління базами даних (СУБД) – це сукупність програм, які дозволяють користувачам створювати та впроваджувати бази даних, а також контролювати доступ до даних, які в них зберігаються. Основною метою СУБД є забезпечення середовища, яке є зручним та ефективним для користувачів, які зберігають та опрацьовують інформацію даної БД.

На рисунку 15.4 наведено приклад представлення традиційної банківської системи з позиції розробки бази даних. У цьому прикладі

банк використовує відділ кадрів, відділ рахунків і відділ кредитів для доступу до спільної корпоративної бази даних.

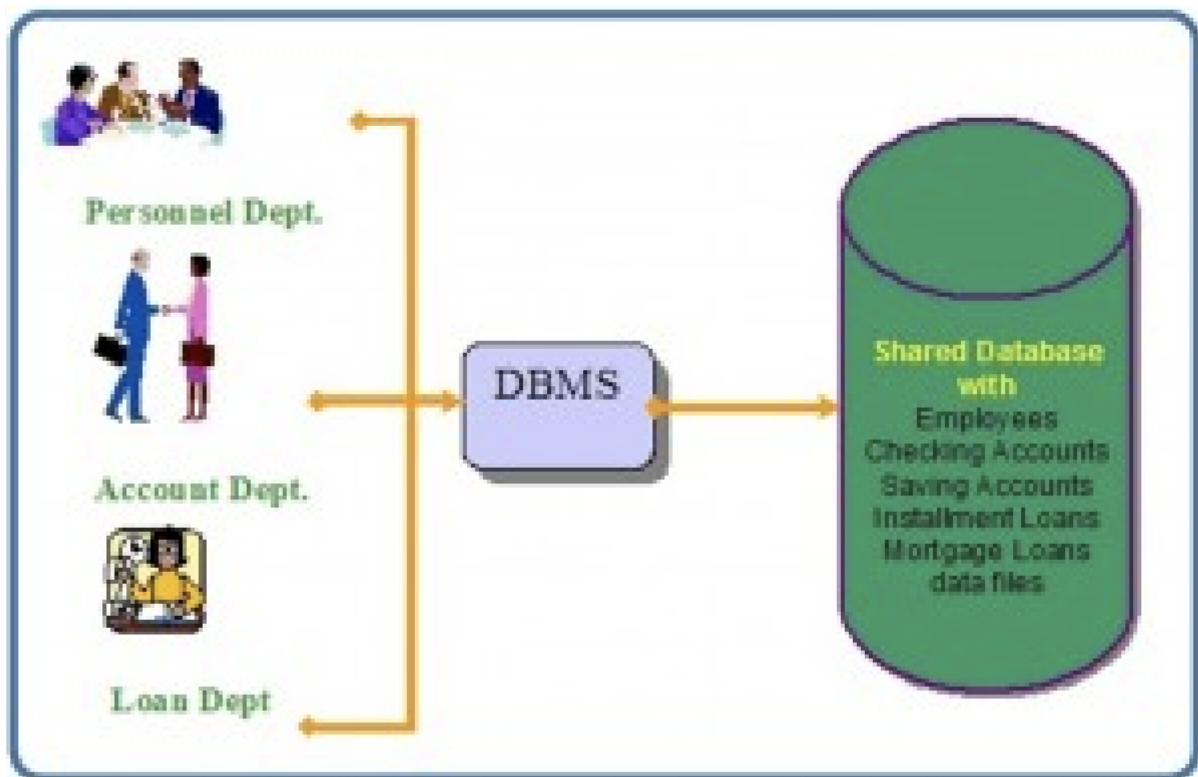


Рисунок 15.4. Система управління базою даних банку (СУБД)

15.4 Характеристики та переваги бази даних

Управління інформацією означає, її організацію та обробку в такий спосіб, щоб ця інформація була зручною з точки зору досягнення мети для якої створюється певна база даних. За використання СУБД, інформація, яку планується зберігати додається до бази даних за певним порядком та в певній формі. Така інформація стає даними, які є доступними та можуть бути інтегрованими до існуючої інформаційної системи (не обов'язково автоматизованої). Управління інформацією за допомогою бази даних дозволяє нам стати стратегічними користувачами даних.

Часто буває потрібним отримати доступ та провести повторне впорядкування даних для рішення різних завдань, наприклад:

- Створення списків розсилки
- Написання управлінських звітів
- Генерування списків певних новин
- Виявлення різних потреб клієнта

Потужність обробки бази даних дозволяє їй маніпулювати даними для реалізації наступних процедур:

- Сортувати дані
- Шукати співпадіння даних
- Створювати зв'язки між даними
- Агрегувати дані за певними параметрами або за використання певних функцій
- Обчислювати нові дані на підставі наявних
- Ранжирувати дані

Завдяки універсальності баз даних, вони можуть бути використані для реалізації проектів в будь-якій сфері людської діяльності. База даних може бути пов'язана з:

- Веб-сайтом для реєстрації користувачів
- Застосунком обліку клієнтів для організацій соціального обслуговування
- Системою медичного обліку для закладу охорони здоров'я
- Особистою адресною книгою електронної пошти
- Збіркою текстових документів
- Системою з бронювання авіа квитків

Існує ряд характеристик, які відрізняють підхід до зберігання інформації за допомогою бази даних від файлових систем.

1 Система баз даних є самостійною одиницею, оскільки вона не тільки містить саму базу даних, але й метадані, які визначають і

описують дані і взаємозв'язки між сутностями в базі даних. Ця інформація за потреби використовується програмним забезпеченням СУБД або користувачами баз даних. Таке розділення даних та інформації про дані робить систему баз даних абсолютно відмінною від традиційної файлової системи, в якій визначення даних є частиною прикладних програм.

2 Ізоляція програми від даних: у файловій системі структура файлів даних визначається в програмах так, якщо користувач хоче змінити структуру файлу, всі програми, які мають доступ до цього файлу, мають бути змінені.

3 Іншого боку, за застосування бази даних структура даних зберігається в системному каталозі а не в програмах. Тому одна зміна — це все, що потрібно для зміни структури файлу. Ця ізоляція між програмами і даними також називається програмно-інформаційною незалежністю.

3 Підтримка декількох поглядів на дані: база даних підтримує декілька представлень даних для різних типів користувачів одних інформаційних ресурсів. Таке представлення є підмножиною бази даних, яка визначена для конкретних користувачів системи. Багато користувачів системи можуть мати різні погляди на одну і ту ж саму інформацію. Кожне представлення може містити тільки дані, які цікавлять певного користувача або групу користувачів.

Кожна база даних містить інформацію про певний клас об'єктів. Сукупність таких об'єктів називається **предметною областю** бази даних (наприклад, вироби на складах, працівники фірми, транспортні засоби підприємства тощо). Інформація про ці об'єкти, представлена в різних документах (наприклад, картки складського обліку, накладні, картки обліку кадрів тощо). Сукупність цієї інформації утворює **модель предметної області**. Опис цих документів (назви документів, колонки, з яких вони складаються і зв'язок між ними) називається **схемою предметної області**.

Під час аналізу баз даних використовують три **рівні представлення предметної області**: зовнішній, концептуальний і внутрішній.

Зовнішній рівень являє частину предметної області, в якій працює конкретний користувач. Для кожного користувача може бути своя зовнішня схема використання. Наприклад, в підсистемі бухгалтерського обліку головний бухгалтер, розраховувач заробітної плати і касир по-різному використовують ту ж саму базу даних, оскільки в них різні зовнішні моделі предметної області в бухгалтерському обліку.

Концептуальний рівень об'єднує точки зору всіх користувачів на предметну область і представлений концептуальною схемою, яка містить логічний опис всіх об'єктів, інформація про які може зберігатися в базі даних, і зв'язків між ними. До складу концептуальної схеми входять: перелік типів – записів, з яких складаються файли бази даних; структура кожного типу записів – перелік полів та їх типів даних; зв'язки між файлами (таблицями).

Внутрішній рівень (фізичний) представляється внутрішньою схемою, яка будується на основі концептуальної схеми з урахуванням подання даних у пам'яті одного комп'ютера або мережі.

15.5 Обмін даними та мультикористувацька система

Сучасні системи управління базами даних призначені для одночасної роботи декількох користувачів з одними й тими ж самими даними. Тобто, вони дозволяють багатьом користувачам одночасно отримувати доступ до певних ресурсів бази даних. Цей доступ досягається за допомогою функцій, що називаються **стратегіями паралельного контролю**. Ці стратегії гарантують, що доступ до даних завжди правильний і що цілісність даних підтримується.

Проектування сучасних мультикористувацьких систем баз даних є значним вдосконаленням відносно систем, які в минулому обмежували використання бази даних однією людиною в певний момент часу та мають наступні переваги.

1 Контроль резервування даних: за використання баз даних, в ідеалі, кожен елемент даних зберігається лише в одному місці в базі даних. У деяких випадках надлишковість даних все ще існує для поліпшення продуктивності системи, але така надмірність контролюється програмними засобами і зводиться до мінімуму ще на етапі проектування бази даних.

2 Обмін даними: інтеграція всіх даних в рамках системи баз даних має багато переваг. По-перше, це дозволяє обмінюватися даними між співробітниками та іншими користувачами, які мають доступ до системи. По-друге, це дає користувачам можливість генерувати більше інформації з певного обсягу вже існуючих даних, ніж це було б можливо без інтеграції.

3 Дотримання обмежень цілісності: системи управління базами даних повинні забезпечити можливість визначення та застосування певних обмежень, щоб користувачі могли вводити актуальну інформацію та підтримувати цілісність даних. Обмеження бази даних – це обмеження або правило, яке визначає те, що може бути введено або відредаговано в таблиці, наприклад, поштовий код з використанням певного формату або додавання існуючої у місті адреси.

Існує багато типів обмежень даних. Тип даних поля бази даних, наприклад, визначає тип даних, які можна вносити в поле (наприклад тільки цілі числа). Унікальність даних, така як первинний ключ, гарантує, що дублікати не вводяться. Обмеження можуть бути простими (на онові полів) або складними (засобами програмування).

4 Обмеження несанкціонованого доступу. Не всі користувачі системи баз даних матимуть однакові привілеї доступу. Наприклад, один користувач може мати доступ тільки для читання (тобто можливість читання файлу, але не внесення змін), в той час як інший може мати привілеї читання, запису та редагування, тобто зміни файлу бази даних. З цієї причини система управління базами даних повинна забезпечити підсистему безпеки для створення і контролю різних типів облікових записів користувачів і обмеження несанкціонованого доступу.

5 Незалежність даних. Ще однією перевагою системи керування базами даних є те, як вона дозволяє отримати незалежність від даних. Іншими словами, системні описи даних – метадані або дані, що описують дані є відокремленими від програм-застосунків. Це можливо через те, що зміни в структурі даних обробляються системою управління базами даних і не вбудовуються в саму програму.

6 Ефективна обробка транзакцій: система управління базами даних повинна включати підсистеми управління паралелізмом. Ця функція гарантує, що дані залишаються послідовними і актуальними під час обробки транзакцій, навіть якщо кілька користувачів оновлюють одну і ту ж інформацію одночасно.

7 Надання різних видів даних. За своєю природою СУБД дозволяє багатьом користувачам мати доступ до певної бази даних як індивідуально, так і одночасно. Користувачам не потрібно знати, як і де зберігаються дані, з якими вони працюють. Резервне копіювання та відновлення даних, є методами, які дозволяють уникати втрат даних за мультикористувацької обробки. Система баз даних забезпечує окремий процес, з резервного копіювання мережі, для резервного копіювання та відновлення даних. Якщо жорсткий диск не функціонує, а, отже, база даних, що зберігається на жорсткому диску, є недоступною, єдиний спосіб відновити базу даних – з резервної копії.

Якщо комп'ютерна система зазнає невдачі під час виконання процесу оновлення, підсистема відновлення відповідає за те, щоб базу даних буде відновлено до її початкового стану.

15.6 Типи моделей даних

На певних етапах проектування баз даних дані представлені з використанням певної моделі даних. **Модель даних** – це сукупність концепцій або нотацій для опису даних, зв'язків між даними, семантики даних та обмежень даних. Більшість моделей даних також включають набір базових операцій для маніпулювання даними в базі даних.

Розглянемо типи моделей даних, які дозволяють впроваджувати наведені вище переваги.

1 Концептуальна модель високого рівня

Концептуальні моделі даних високого рівня забезпечують загальні представлення даних в термінах близьких до тих, якими люди сприймають дані. Типовим прикладом є модель відносин сутності, яка використовує основні поняття, такі як сутності, атрибути і відносини. Сутність представляє об'єкт реального світу, такий як працівник або проект. Сутність має атрибути, які представляють такі властивості, як ім'я працівника, адреса і дата народження. Відносини є асоціаціями між суб'єктами; наприклад, працівник працює над багатьма проектами. Між працівником і кожним проектом існує взаємозв'язок.

2 Логічні моделі даних на основі записів

Логічні моделі даних на основі записів забезпечують концепції, які користувачі можуть зрозуміти, але не надто далекі від того, як дані зберігаються в комп'ютері. Три відомі моделі даних цього типу — це реляційні моделі даних, мережеві моделі даних та ієрархічні моделі даних.

В **реляційній моделі даних** предметну область представляють у вигляді таблиць. Найменування стовпців і зв'язки між ними утворюють схему предметної області, а вміст таблиць — її модель.

Подання інформації у вигляді таблиць, що складаються з заданих стовпців, які розташовані в певному порядку зліва направо, у математиці називається відношенням. Звідси і назва моделі — реляційна (від англ. relation – відношення).

В реляційній базі даних таблиці розміщуються в файлах. Кожна таблиця відповідає одному інформаційному об'єкту предметної області.

Наприклад, в системі учасників в організації Science World, кожна група має багато учасників (рис.2.2). Ідентифікатор, дата закінчення та адресна інформація є полями учасника. Окремі учасники є

екземплярами сутності «учасник». Кожен запис, є екземпляром об'єкта «учасник» або рядком таблиці «учасник».

Мережева модель даних – логічна модель даних, що є розширенням ієрархічного підходу, суворе математична теорія, що описує структурний аспект, аспект цілісності і аспект обробки даних в мережевих базах даних. Мережева модель представляє дані як типи записів. Ця модель також є визначенням типу одного з множини відношень, які називаються множиною типів, як показано на рис. 15.5.

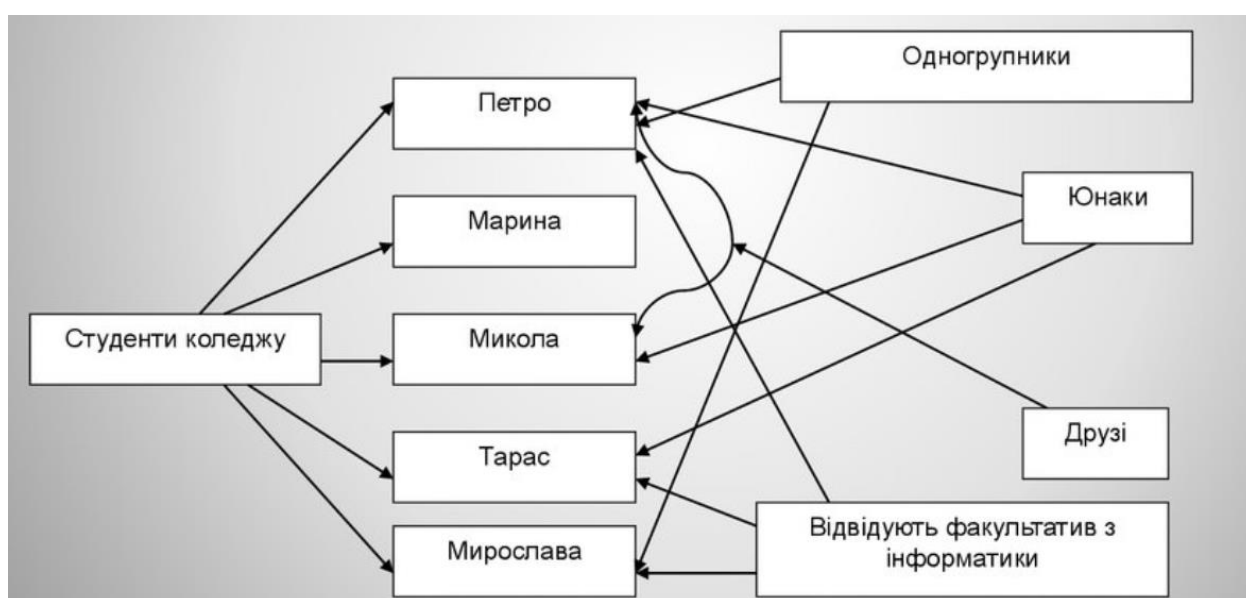


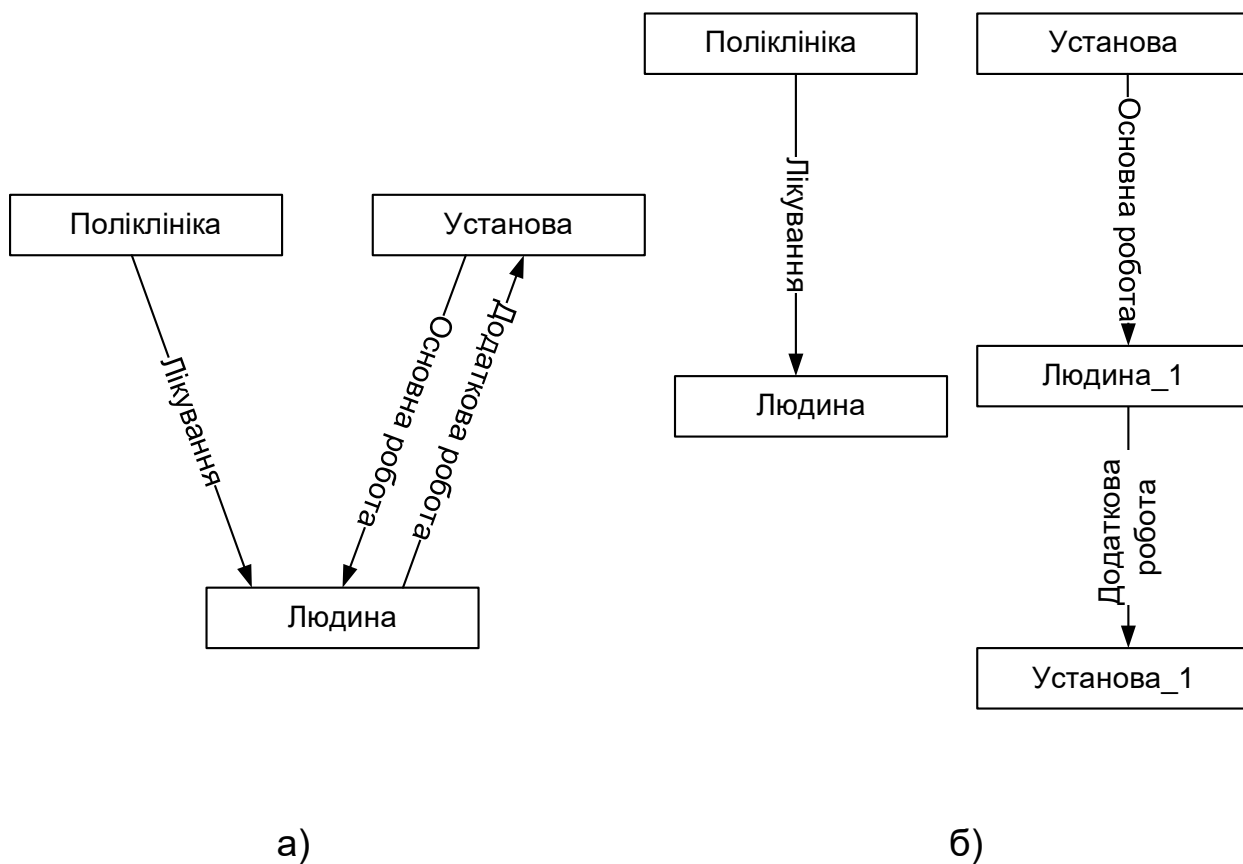
Рисунок 15.5. Мережева модель представляє даних

Ієрархічна модель представляє дані у вигляді ієрархічної деревоподібної структури, в якій кожна гілка ієрархії представляє ряд пов'язаних записів. На рисунку 15.6 показано схему ієрархічного представлення даних.



Рисунок 15.6. Ієрархічна модель представляє даних

Різниця між ієрархічною моделлю даних і мережевою полягає в тому, що в ієрархічних структурах запис-нащадок повинен мати в тільки одного предка, а в мережевій структурі даних у нащадка може бути будь-яка кількість предків. Різниця між мережевою та ієрархічною моделями наведена на рисунку 15.7.



а – мережева модель; б – ієрархічна модель

Рисунок 15.7. Різниця між мережевою та ієрархічною моделями

15.7 Моделювання даних

Моделювання даних є *першим етапом* в процесі проектування баз даних. Цей етап іноді вважається високорівневою або абстрактною фазою проектування, яку також називають концептуальним дизайном. Метою цього етапу є опис:

- Даних, які містяться в базі даних (наприклад, сутностей: студент, викладач, курс, факультет)
- Взаємозв'язків між елементами даних (наприклад, студенти навчаються на факультетах; викладачі викладають курси)
- Обмеження на дані (наприклад, номер студента повинен мати вісім цифр; курс може складати від 3 до 12 кредитів)

На *другому етапі* елементи даних, відношення та обмеження виражаються з використанням концепцій, передбачених моделлю даних високого рівня. Через те, що ці концепції не включають деталі реалізації, результатом процесу моделювання даних є напівформальне представлення структури бази даних. Результат цього етапу проектування легко зрозуміти кінцевому користувачеві БД, тому він використовується як посилання для того, щоб переконатися, що всі вимоги користувача виконані.

Третій етап – проектування баз даних, який можна розділити на два підетапи: перший називається "логічний дизайн бази даних", який визначає базу даних в моделі даних конкретної СУБД, а другий – "фізичний дизайн бази даних", який визначає внутрішню структуру зберігання даних, організацію файлів або методи індексації. Ці два підетапи реалізують створення бази даних та інтерфейсу операцій та користувача.

15.8 Рівні абстрагування даних

Розглянемо процес проектування бази даних з точки зору деталізації специфіки даних. Так само, як будь-який дизайн починається на високому рівні і поступово рухається в бік зростання рівня деталізації, так само як і розробка баз даних. Наприклад, під час побудови будинку ви починаєте з того, скільки спальень і ванних кімнат буде у будинку, чи будуть вони розташовані на одному поверсі або декількох. Наступним кроком є отримання архітектором дизайну будинку з більш структурованої перспективи. Цей рівень стає більш детальним описом фактичних розмірів кімнати, розміщення сантехніки, розведення електричної системи тощо. Останній крок – найняти підрядника для будівництва будинку. Це погляд на дизайн від високого рівня абстракції до зростаючого рівня деталізації.

Дизайн бази даних дуже схожий на наведений вище. Він починається з того, що користувачі ідентифікують бізнес-правила; потім розробники баз даних і аналітики створюють дизайн бази даних; а потім адміністратор бази даних реалізує дизайн за допомогою певної СУБД.

Наведемо етапи розробки БД в порядку зменшення рівня абстракції даних.

1 Зовнішні моделі

- Представлення погляду користувача бази даних
- Об'єднання кілька різних зовнішніх поглядів
- Визначення зв'язків з предметами та явищами реального світу, з якими працює кожен користувач бази даних

2 Концептуальні моделі

- Забезпечення гнучких можливостей структурування даних
- Відображення сукупного погляду всіх користувачів: логічна структура всієї бази даних
- Розміщення даних, що мають зберігатися в базі даних

- Відображення взаємозв'язків між даними в тому числі: обмежень, семантичної інформації (наприклад, правила ведення бізнесу), інформації про безпеку та цілісність даних.

- Представлення бази даних як сукупності сутностей (об'єктів) різних видів

- Створення основи для ідентифікації та високорівневого опису основних об'єктів даних за уникнення деталей

3 Внутрішні моделі

Трьома найвідомішими моделями даного виду є реляційна модель даних, мережева модель даних та ієрархічна модель даних. За застосування внутрішніх моделей слід брати до уваги такі особливості:

- База даних розглядається як сукупність записів фіксованого розміру

- Такі моделі знаходяться ближче до фізичного рівня або структури файлів

- Такі моделі є представленням бази даних, як їх бачать СУБД.

- Такі моделі вимагають від дизайнера відповідності характеристик і обмежень концептуальної моделі до характеристик обраної моделі реалізації

- Такі моделі є переходом від відображення сутностей в концептуальній моделі до таблиць в реляційній моделі

4 Фізичні моделі

- Є фізичним представленням бази даних

- Мають найнижчий рівень абстракцій

- Визначають як саме зберігаються дані в пам'яті комп'ютера, а відповідно визначають наступні параметри: швидкість часу виконання операцій з даними, ступінь стиснення та кількість ресурсів для зберігання даних, організацію файлів і методи доступу до даних, шифрування даних.

- Визначають операційну систему для управління базою даних
- Визначають деталі того, як дані зберігаються в пам'яті комп'ютера

На практичному прикладі можна побачити, як різні моделі працюють разом. Розглянемо це питання з позиції зовнішньої моделі.

Зовнішня модель – це погляд кінцевого користувача на дані. Зазвичай база даних є корпоративною системою, яка обслуговує потреби декількох підрозділів. Проте, дані, які потрібні для роботи одного відділу не мають бути досяжними для працівників інших відділів (наприклад, відділ кадрів не повинен мати доступу до даних відділу продажів). Таким чином, одне представлення користувача буде відрізнятися від представлення іншого користувача.

Зовнішня модель забезпечує розподіл розробником набору вимог та обмежень на функціональні модулі, які можуть бути розглянуті в рамках їх зовнішніх моделей (наприклад, людські ресурси в порівнянні з продажами).

Розробник бази даних має розуміти всі дані так, щоб можна було побудувати базу даних, яка є корисною для функціонування всього підприємства (фірми, установи тощо). Концептуальна модель є першим представленням бази даних, яке враховує потреби одразу всіх відділів.

На цьому етапі концептуальна модель не залежить ні від програмного забезпечення, ні від апаратного забезпечення. Вона не залежить від програмного забезпечення СУБД ані від апаратного забезпечення, яке використовується для реалізації моделі. Зміни в апаратному або програмному забезпеченні не впливають на дизайн бази даних на концептуальному рівні.

Після вибору СУБД можна реалізувати внутрішню модель. На етапі створення внутрішньої моделі слід створити всі таблиці, обмеження, ключі, правила тощо. Таку модель часто називають логічною конструкцією.

Фізична модель – це просто спосіб зберігання даних на диску. Кожен постачальник баз даних має свій спосіб зберігання даних.

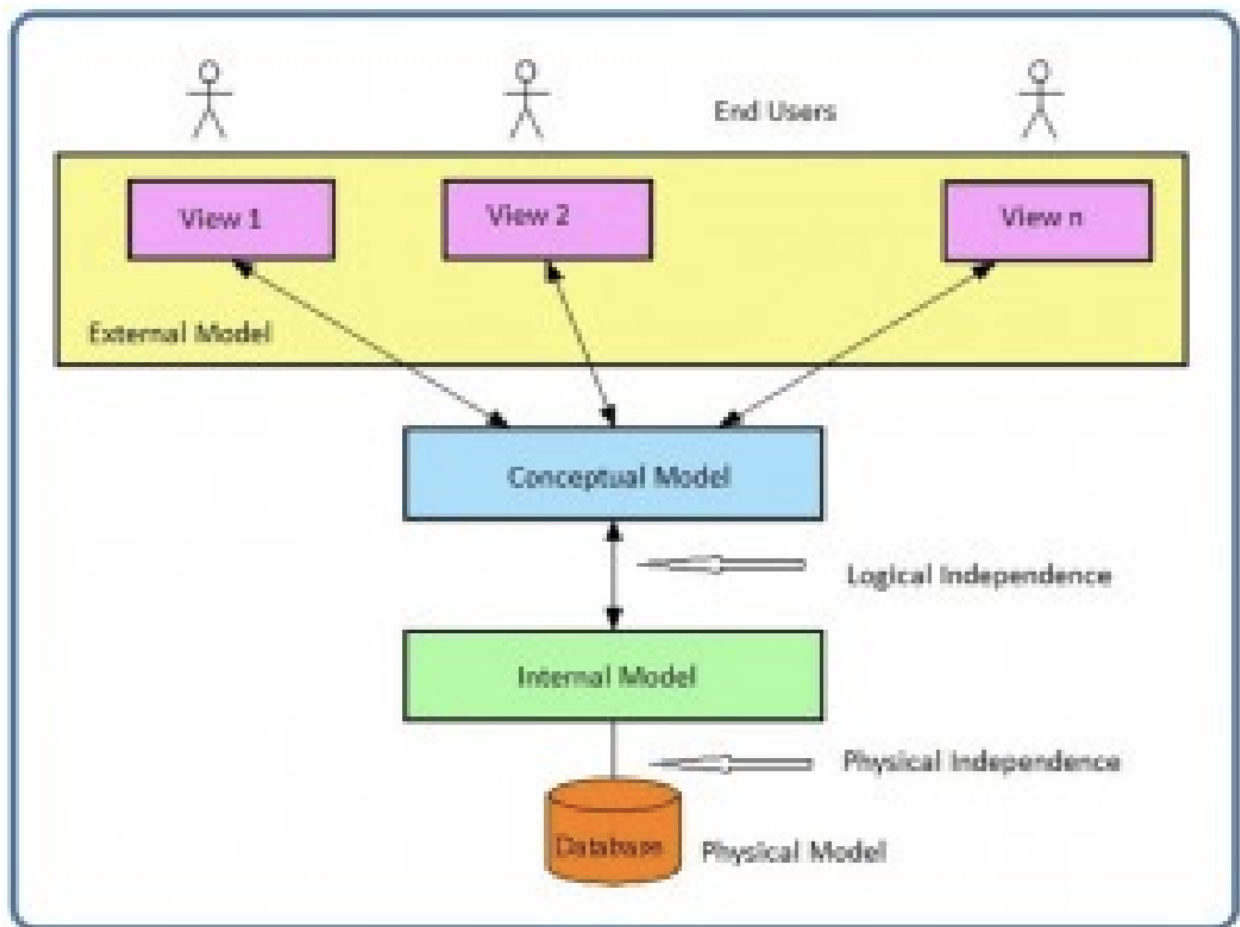


Рисунок 15.8. Рівні абстрагування даних

15.9 Схеми даних

Схеми є загальним описом бази даних, і зазвичай вони представлені діаграмою взаємозв'язку сутностей (ERD). Існує багато схем, які представляють зовнішні моделі і таким чином відображають зовнішнє представлення даних. Нижче наведено список елементів, які слід розглядати під час процесу проектування бази даних.

- Зовнішні схеми, яких має бути декілька
- Схеми відображення різних зовнішніх представлень даних

- Концептуальна схема, яка має бути лише одна. Ця схема включає елементи даних, взаємозв'язки і обмеження, які представлені в ERD.

- Фізична схема, яка має бути лише одна

Незалежність даних відображає стійкість користувацьких застосунків до змін, яким можуть піддаватися визначення та організація даних. Абстракції даних відбиває лише ті елементи, які є важливими або корисними для користувача. Складність обробки даних приховується від користувача бази даних.

Незалежність даних та незалежність операцій разом формують особливість абстракції даних. Існує два типи незалежності даних: логічна та фізична.

Логічна схема є концептуальним проектом бази даних, виконаним на папері або дошці, подібно до архітектурних креслень будинку. Можливість зміни логічної схеми, без зміни зовнішньої моделі або представлення користувача, називається логічною незалежністю даних. Наприклад, додавання або видалення нових сутностей, атрибутів або зв'язків до концептуальної схеми повинно бути можливим без необхідності зміни існуючих зовнішніх схем або переписування існуючих прикладних програм.

Іншими словами, зміни в логічній схемі (наприклад, зміни структури бази даних, такі як додавання стовпця або інших таблиць) не повинні впливати на функцію застосунку (зовнішнє представлення).

Фізична незалежність даних стосується стійкості внутрішньої моделі щодо змін фізичної моделі. Логічна схема залишається незмінною, навіть якщо внесені зміни в організацію файлів або структуру їх зберігання, пристрої зберігання даних або стратегію індексації. Фізична незалежність даних включає в себе приховування деталей структури зберігання від користувацьких застосунків. Прикладні застосунки не повинні бути задіяні в питаннях організації зберігання даних на конкретних пристроях, оскільки операції обробки даних не мають залежати від фізичної моделі даних.

15.10 Хмарне зберігання даних

Сьогодні «хмарам» довіряють свою інформацію мільйони користувачів по всьому світу.

Хмарне сховище даних (англ. Cloud storage) – модель онлайн-сховища, в якому дані зберігаються на численних розподілених в мережі серверах, що надаються в користування клієнтам, в основному, третьою особою. На відміну від моделі зберігання даних на власних виділених серверах, придбаних або орендованих спеціально для подібних цілей, кількість або будь-яка внутрішня структура серверів, за загальним правилом, клієнту не відома. Дані зберігаються і обробляються в так званій «хмарі», яка представляє собою, з точки зору клієнта, один великий віртуальний сервер. Фізично ж такі сервери можуть розташовуватися віддалено один від одного географічно.

Хмарне сховище даних – це свого роду віртуальний носій інформації, який зберігає і обробляє дані на численних серверах, розкиданих у всесвітній павутині.

Хмарні шлюзи – технологія, яка може бути використана для більш зручного надання «хмари» клієнту. Наприклад, за допомогою відповідного програмного забезпечення, сховище в «хмарі» може бути надано для клієнта як локальний диск на комп'ютері. Таким чином, робота з даними в «хмарі» для клієнта стає абсолютно прозорою. За наявності гарного, швидкого зв'язку з «хмарою» клієнт може навіть не помічати, що працює не з локальними даними у себе на комп'ютері, а з даними, що зберігаються, можливо, за багато сотень кілометрів від нього.

Причин для розміщення даних в хмарі може бути досить багато, і для різних користувачів вони можуть мати різний пріоритет. Наприклад, для приватних осіб важливіша буде можливість доступу до даних з різних місць інтернету та з різних пристроїв, а для корпоративних користувачів більш істотними можуть виявитися надійність і вартість зберігання, а також трансформація капітальних витрат в операційні діяльності (використання таксі замість покупки службового автомобіля).

15.11 Типи хмарних сховищ

Існує три типи хмарних сховищ даних, кожен з яких пропонує унікальні переваги і власні приклади використання.

1. *Об'єктне сховище*. Для застосунків, розроблених в хмарі, як правило, потрібні такі переваги об'єктного сховища, як широкі можливості масштабування і характеристики метаданих. Об'єктні сховища, наприклад Amazon Simple Storage Service (S3), ідеально підходять для розробки сучасних застосунків з нуля, коли потрібна гнучкість і можливість масштабування. Крім того, ці сховища можна використовувати для імпорту даних з існуючих сховищ з метою аналітики, резервного копіювання або архівації.

2. *Файлове сховище*. Деяким програмам потрібно отримувати доступ до передачі файлів, отже, їм необхідна файлова система. Даний тип сховища часто підтримується сервером сховищ, підключеним до мережі. Рішення файлових сховищ, наприклад Amazon Elastic File System (EFS), ідеально підходять для таких прикладів використання, як великі репозиторії контенту, середовища розробки, мультимедійні сховища або особисті каталоги користувачів.

3. *Блочне сховище*. Інші корпоративні застосунки, наприклад бази даних або системи планування ресурсів підприємства Enterprise Resource Planning (ERP), часто потребують виділеного сховища з низькими затримками для кожного з вузлів. Таке сховище працює аналогічно до сховища з прямим підключенням Direct-attached storage (DAS) або мережі зберігання даних Storage Area Network (SAN). Рішення хмарних сховищ на основі блоків, такі як Amazon Elastic Block Store (EBS), виділяють сховище для кожного віртуального сервера і забезпечують наднизьку затримку для робочих навантажень, що вимагають високої продуктивності.

Розміщувати в інтернеті можна різні дані, по-різному організовані, тому і сховища теж можуть бути різними.

Але щоб перейти до розгляду типів інтернет-сховищ потрібно зробити одне застереження: дані в них можуть зберігати не тільки користувачі, але і застосунки, якими, в свою чергу, користуються люди або підприємства. Наприклад, багато програм для зв'язку - Skype,

WhatsApp, Facebook Messenger та інші - зберігають контакти користувачів в своїх інтернет-сховищах.

З точки зору користувача, інтернет-сховище може виглядати як додатковий локальний диск або папка для розміщення довільних файлів. Останній варіант добре знайомий численним користувачам сервісів Dropbox, OneDrive, Яндекс.Диск, Google.Drive.

Але є типи інтернет-сховищ, з якими приватні особи зазвичай не стикаються. Сховища цих типів використовуються додатками. Наприклад, таке сховище може мати давно звичний SQL-інтерфейс, але при цьому не буде вимагатися будь-яке системне адміністрування SQL-сервера.

Найбільш сучасним і універсальним типом хмарних сховищ є об'єктне сховище, яке дозволяє працювати з даними найбільш узагальненим способом. Для більшості прикладних користувачів мережевий диск або мережева папка – поняття близькі, якщо не ідентичні. Але в частині системного адміністрування вони істотно розрізняються.

15.12 Хмарні диски

Зараз дуже часто хмарними дисками називають сервіси типу Dropbox, OneDrive. Ці ресурси надаються користувачам або через веб-інтерфейс, або у вигляді папок на робочому столі. В обох випадках мова не може йти про диски. Наприклад, їх не можна відформатувати в потрібну файлову систему.

Але дисковий простір сховища дійсно може бути представлено якомусь комп'ютеру через інтернет так, щоб воно сприймалося як локальний диск, з яким будуть доступні всі дискові операції.

Для надання дискового простору через інтернет існують спеціальні протоколи, наприклад, Internet Small Computer Systems Interface (iSCSI), Internet Fibre Channel Protocol (iFCP) або Fibre Channel over IP (FCIP). На їх основі створюються, так звані, мережі зберігання даних (SAN, Storage Area Network). За цими протоколами дискові пристрої представляються серверу, до якого вони підключаються, на найнижчому – блоковому – рівні, і тому є універсальними.

Однак взаємодія на такому рівні вимагає інтенсивного обміну даними між сервером і дисковим пристроєм. Якщо такий обмін йде по недостатньо швидким каналах зв'язку, ефективність дискового пристрою падає. Якщо зв'язок високошвидкісний і високонадійний, хмарні диски можуть стати зручною і універсальною частиною корпоративної інформаційної системи.

15.13 Хмарні папки

Саме їх зараз часто називають хмарними дисками (Google Drive). Хмарна папка – корисний і широко затребуваний ресурс, особливо серед приватних осіб. Наприклад, він дає можливість використовувати одні і ті ж файли з різних пристроїв, підключених до інтернету. Також можливе спільне використання папки для групової роботи з файлами.

Покласти файл в хмарну папку можна або через браузер і веб-інтерфейс, або через спеціальну локальну папку, яка автоматично синхронізується з хмарним сховищем. У другому випадку на комп'ютер користувача потрібно встановити відповідну утиліту.

Крім доступу до файлів з різних пристроїв, хмарні папки забезпечують зберігання резервних копій цих файлів. Як правило, хмарна інфраструктура будується на обладнанні з дуже високою надійністю.

15.14 Об'єктні сховища

Концепція об'єктного сховища дозволила забезпечити універсальну обробку даних довільного типу.

Файли в такому сховищі супроводжуються додатковими відомостями - даними - які дозволяють обробляти ці файли як прикладні об'єкти: бухгалтерські документи, відеофільми, проекти, товари, фотографії тощо.

Ідея об'єктного зберігання даних виникла і опрацьовувалася ще в 70-і та 80-і роки. Але тоді вона розвивалася в рамках окремих замкнутих проектів, а зараз це – широко затребувана послуга, яка використовується в найрізноманітніших застосунках.

Для взаємодії з хмарним об'єктним сховищем використовується програмний інтерфейс Application Programming Interface (API). Тобто

запис об'єкта в сховище відбувається не перетягуванням файлу з однієї папки в іншу, а за допомогою спеціальних команд, які видаються застосунками.

Сучасні хмарні об'єктні сховища забезпечують високий рівень надійності зберігання даних, гнучкість їх розміщення і опису, високу масштабованість і низьку питому вартість зберігання.

В об'єктному сховищі прикладної користувач може ефективно зберігати як безліч відносно невеликих об'єктів, так і величезні, наприклад, відеофільми.

15.15. Способи використання хмарних сховищ

Створення резервних копій та відновлення критично важливі для забезпечення захисту та доступності даних, однак дотримання відповідності зростаючим потребам в області ресурсів може стати постійною проблемою. Хмарне сховище забезпечує низьку вартість, високу надійність і практично безмежні можливості масштабування для рішень резервного копіювання та відновлення. Вбудовані політики управління даними, наприклад Amazon S3 Object Lifecycle Management, можуть виконувати автоматичну міграцію даних на більш економічні рівні сховища на підставі частотних або часових параметрів, за рахунок чого можна створювати архівні сховища, що дозволяють полегшити дотримання юридичних або нормативних вимог. Ці переваги надають широкі можливості масштабування в галузях фінансових послуг, охорони здоров'я та ЗМІ, де постійно створюються великі обсяги даних з необхідністю тривалого зберігання.

Середовища тестування і розробки програмного забезпечення часто вимагають створення, використання і подальшого видалення окремих, незалежних і дублюючих середовищ зберігання. Крім тимчасових витрат, з цими процесами можуть бути пов'язані серйозні початкові капіталовкладення.

Деякі з найбільших та найприбутковіших компаній світу змогли створити застосунки в рекордно швидкий термін завдяки гнучкості, продуктивності та низькій вартості хмарного сховища. Навіть роботу найпростіших статичних веб-сайтів можна поліпшити з мінімальними

витратами. Розробники з усього світу звертаються до рішень для зберігання даних з оплатою за фактом використання, які позбавляють їх від проблем з управлінням і масштабуванням.

Доступність, надійність і фінансові переваги хмарного зберігання можуть здаватися дуже привабливими для власників компаній, однак для відповідальних фахівців традиційних ІТ-напрямків (наприклад, адміністраторів систем зберігання даних, систем резервного копіювання, мереж, систем безпеки, а також керівників з питань дотримання вимог) перенесення великої кількості даних в хмару часто представляється серйозною проблемою. Сервіси міграції даних в хмару, такі як AWS Import / Export Snowball, дозволяють спростити міграцію сховища в хмару і вирішують завдання, пов'язані з високими витратами на використання мережі, тривалою передачею даних і питаннями безпеки.

Традиційні локальні рішення для зберігання даних можуть виявитися непередбачуваними в питаннях вартості, продуктивності і масштабованості, особливо з плином часу. Проекти, пов'язані з великими даними, вимагають наявності великомасштабних, доступних і надійних пулів сховищ даних з високою доступністю. Часто подібні пули називають «озерами даних».

Озера даних, побудовані на базі об'єктних сховищ, зберігають інформацію в початковій формі і містять розширені метадані, що дозволяють вибірково витягувати і використовувати дані з метою аналізу. Хмарні озера даних можуть бути розташовані в центрі будь-якого типу систем зберігання і обробки великих даних і аналітичних движків, наприклад Amazon Redshift, Amazon RDS, Amazon EMR, Amazon DynamoDB, що дозволить Вам виконати наступний проект швидше і з більшим ступенем релевантності.

Приклади використання хмарних сховищ:

- Системи резервного копіювання даних і серверів
- Цифрові бібліотеки
- Системи дистанційної освіти
- Системи відеонагляду

- Відеотеки
- Архіви корпоративних і офіційних документів
- Публічні бази даних
- Ігрові платформи

15.16 Сутність технології

Принцип роботи будь-якого «хмарного» сховища приблизно наступний: на персональний комп'ютер або ноутбук ставиться програма-клієнт «хмарного» сховища, прописується шлях до папок розташованих на жорсткому диску, які планується помістити в «хмару». Програма-клієнт копіює інформацію з зазначених папок в сховище, і в подальшому відстежує будь-які зміни в цих папках і автоматично вносить корективи в «хмарне» сховище даних.

Якщо Ви вирішили змінити файл, що зберігається в «хмарі», програма внесе правки в копії файлів на Вашому комп'ютері. Такий підхід дозволяє мати актуальний набір файлів на будь-якому з Ваших пристроїв (смартфоні, комп'ютері, планшеті тощо). Єдина умова, яка потрібна для безперебійної роботи сховища з файлами комп'ютера - повна синхронізація.

Після включення ПК Ви також повинні дочекатися, поки пройде синхронізація даних. Швидкість здійснення даного процесу багато в чому залежить від швидкості з'єднання з інтернетом. Якщо вимкнути пристрій передчасно, можлива помилка синхронізації даних хмарного сховища.

Вимоги, які висуваються до хмарного сховища.

Зберігання даних в хмарі може поставити питання про регулювання і відповідність вимогам, особливо якщо дані вже знаходяться в системах сховищ, обмежених певними вимогами. Хмарні засоби забезпечення відповідності вимогам, наприклад Amazon Glacier Vault Lock, гарантують впевненість в тому, що Ви легко виконайте розгортання і включіть в роботу засоби забезпечення відповідності вимогам для окремих сховищ даних за допомогою певних політик. Питання забезпечення надійного зберігання, безпеки та доступності

критично важливих корпоративних даних мають першорядну важливість. При розгляді варіанту зберігання даних в хмарі існує кілька фундаментальних вимог.

1 Надійність. Дані повинні зберігатися з надмірністю. В ідеалі вони повинні бути розподілені між кількома об'єктами і кількома пристроями в рамках кожного з об'єктів. Стихійні лиха, людський фактор або механічні несправності не повинні призводити до втрати даних.

2 Доступність. Всі дані повинні бути доступними в разі необхідності, але існує різниця між виробничими даними і архівами. Ідеальне хмарне сховище пропонує оптимальне поєднання між часом отримання даних і вартістю.

3 Безпека. В ідеалі всі дані повинні шифруватися – як ід час зберігання, так і під час передачі. Дозволи та контроль доступу повинні працювати в хмарі так само, як і в локальних сховищах даних. Ви можете вказати спеціальні параметри, наприклад одноразовий запис з багаторазовим читанням (WORM), щоб заблокувати дані від подальших змін. Рішення для роботи з журналами аудиту, наприклад AWS CloudTrail, допоможуть вирішити всі питання в області забезпечення відповідності вимогам для хмарних сховищ і систем архівації.

Переваги технології:

- Можливість доступу до даних з будь-якого комп'ютера, що має вихід в Інтернет.
- Можливість організації спільної роботи з даними.
- Висока ймовірність збереження даних навіть у разі апаратних збоїв.
- Сукупна вартість володіння. Завдяки хмарному сховищу Вам не потрібно купувати устаткування, виділяти ресурси для сховища або витрачати кошти на те, що «коли-небудь стане в нагоді». Ви можете додавати або видаляти ресурси на вимогу, швидко змінювати продуктивність та термін зберігання. І при цьому Ви будете платити тільки за ресурси, якф використовуєте. Дані, які використовуються не так часто, можна автоматично переміщати

на більш економічні рівні за певними правилами, дія яких легко контролюється. Це дозволяє забезпечити економію обсягу ресурсів. При цьому, клієнт платить тільки за те місце в сховищі, яке фактично використовує, але не за оренду сервера, всі ресурси якого він може і не використовувати.

- Клієнту немає необхідності займатися придбанням, підтримкою і обслуговуванням власної інфраструктури зі зберігання даних, що в решті решт, зменшує загальні витрати виробництва.
- Хмарне сховище підходить, практично, під всі операційні системи, як ПК, так і мобільних (проте є й обмеження у використанні ПО).
- Доступ і синхронізація файлів хмари з файлами Вашої операційної системи залежить від самого клієнта хмари (під час вибору хмари зверніть увагу на готовність хмари до десктопних застосунків).
- Управління інформацією. Централізоване сховище в хмарі створює величезні можливості для нових прикладів використання. Використовуючи політики управління життєвим циклом в хмарному сховищі, можна вирішувати важливі завдання, пов'язані з управлінням інформацією, включаючи автоматичний розподіл за рівнями або блокування даних з метою дотримання вимог.
- Час до розгортання. Коли команди розробників готові до запуску проектів, інфраструктура не повинна обмежувати їх. Хмарне сховище дозволяє ІТ-фахівцям швидко виділяти необхідний простір для зберігання даних саме тоді, коли це потрібно. В результаті ІТ-фахівці можуть зосередитися на вирішенні складних проблем, пов'язаних з застосунками, а не на питаннях управління системами зберігання даних. Всі процедури з резервування та збереження цілісності даних реалізуються провайдером «хмарного» центру, яка не втягує в цей процес клієнта.

Недоліки технології:

Безпека за зберігання та пересилання даних є одним з основних питань під час роботи із «хмарою», особливо щодо конфіденційних і приватних даних. Так, наприклад, провайдер має можливість переглядати дані

клієнта (якщо вони не захищені паролем), які також можуть потрапити в руки хакерів, які зуміли зламати системи захисту провайдера.

Надійність, своєчасність отримання і доступність даних в «хмарі» дуже сильно залежить від багатьох проміжних параметрів, таких як: канали передачі даних на шляху від клієнта до «хмари», надійність останньої милі, якість роботи інтернет-провайдера клієнта, доступність самого «хмари» в даний момент часу. Якщо ж сама компанія, що надає онлайнове сховище, буде ліквідована, клієнт може втратити всі свої дані.

Загальна продуктивність при роботі з даними в «хмарі» може бути нижче, ніж при роботі з локальними копіями даних.

Абонентська плата за додаткові можливості (збільшений обсяг зберігання даних, передача великих файлів тощо).

15.17 Популярні сервіси

Якщо поглянути на доступні сьогодні хмари, то можна нарахувати з десяток таких сервісів, але лише кілька з них ми б виділили для використання. Для детальнішого розгляду хмар було вибрано кілька з них з міркувань якості самого сервісу, його захисту, що підтримується програмним забезпеченням і особистої довіри до його розробників, а також до компаній з підтримки.

Найпопулярнішими серед сервісів хмарного зберігання даних є такі:

iDrive:	https://www.idrive.com/
OpenDrive:	https://www.opendrive.com/
Syncplicity:	https://www.syncplicity.com/
MediaFire:	https://www.mediafire.com/
Amazon Web Services:	https://aws.amazon.com/ru/?nc2=h_lg
4shared:	https://www.4shared.com/
Google Drive:	https://www.google.com/intl/uk/drive/
iCloud Drive:	https://www.icloud.com/
Mega:	https://mega.nz/
Box.net:	https://www.box.com/

SpiderOak:		https://spideroak.com/
Adrive:		http://www.adrive.com/
SkyDrive:		https://onedrive.live.com/about/uk-ua/
Copy:		https://copy.com/?r=O78BHX
4Sync:		https://en.4sync.com/
Cubby:		https://www.cubby.com/
Hubic:		https://hubic.com/en/
Yunpan	360:	https://eyun.360.cn/?src=old
SugarSync:		https://www.sugarsync.com/
Bitcasa:		http://www.bitcasa.com/