Лекція 02 Типи інформаційних систем. Класифікація баз даних

План заняття

1. Типи інформаційних систем
2. Місце баз даних в архітектурі інформаційних системах
3. Типи баз даних
4. Нові області застосування баз даних

**Типи інформаційних систем**

Сучасні **інформаційні системи** – складні комплекси апаратних і програмних засобів, технології й персоналу (автоматизовані інформаційні системи).



*Апаратне забезпечення ІС* містить у собі широкий набір засобів обчислювальної техніки, передачі даних, а також цілий ряд спеціальних технічних пристроїв (пристрою графічного відображення інформації, аудио- і відеопристрою, засобу мовного уведення й т.д.). Апаратне забезпечення є основою будь-якої ІС.

*Комунікаційне (мережне) забезпечення* містить у собі комплекс апаратних мережних комунікацій і програмних засобів підтримки комунікацій в ІС. Воно має істотне значення при створенні розподілених ІС й ІС на основі Інтернету.

*Програмне забезпечення ІС* забезпечує реалізацію функцій введення даних, їх розміщення на носіях, модифікації даних, доступ до даних, підтримку функціонування устаткування. Програмне забезпечення можна розділити на системне (яке завершує процес вибору апаратно-програмного рішення або платформи) і користувацьке (яке застосовується для вирішення завдань задоволення потреб користувача у комп'ютерному середовищі).

*Лінгвістичне забезпечення ІС* призначене для вирішення завдань формалізації змісту повнотекстової і спеціальної інформації для створення пошукового образу даних (профілю). У класичному змісті звичайно воно включає процедури індексування текстів, їхню класифікацію і тематичну рубрикацію.

У міру зростання складності і масштабів ІС важливу роль починає відігравати *організаційно-технологічне забезпечення*, що з'єднує різнорідні компоненти (апаратури, програми й персонал) у єдину систему й забезпечує процедури її керування й функціонування. Недооцінка цієї складової ІС найчастіше призводить до зриву строків впровадження системи й виведення її на виробничі потужності.

Класифікація ІС

за рівнем (сферою) діяльності:

державні

територіальні

галузеві

об’єднань підприємств (корпорацій)

технологічних процесів;

за рівнем автоматизації процесів управління:

інформаційно-пошукові

інформаційно-довідкові

інформаційно-управлінські

системи підтримки прийняття рішень;

за ступенем централізації обробки інформації:

централізовані ІС

децентралізовані ІС

ІС колективного використання;

за ступенем інтеграції функцій:

багаторівневі ІС з інтеграцією за рівнями планування

багаторівневі ІС з інтеграцією за рівнями управління.

Архітектуру інформаційної системи можна описати як концепцію, що визначає модель, структуру, виконувані функції й взаємозв'язок компонентів інформаційної системи. За способом доступу до БД інформаційні системи поділяють на локальні, файл – серверні та клієнт серверні.

В *локальних* ІС БД і СУБД знаходяться на одному комп'ютері.

Переваги - автономність (незалежність).

Недоліки - з БД працює тільки одна людина; складно обновляти при великій кількості користувачів практично неможливо “зістикувати” зміни, які вносять різні користувачі.

У *файл-серверних* СУБД файли даних розташовуються централізовано на файл-сервері. СУБД розташовується на кожному клієнтському комп'ютері (робочої станції). Доступ СУБД до даних здійснюється через локальну мережу. Синхронізація читань і оновлень здійснюється за допомогою файлових блокувань. Файл-серверна архітектура має на увазі наявність виділеного мережного ресурсу для зберігання даних. Такий ресурс називається «файловим сервером». При такій архітектурі всі функціональні компоненти системи розташовані на користувальницькому комп’ютері, що називається «клієнтом», а самі дані перебувають на сервері.

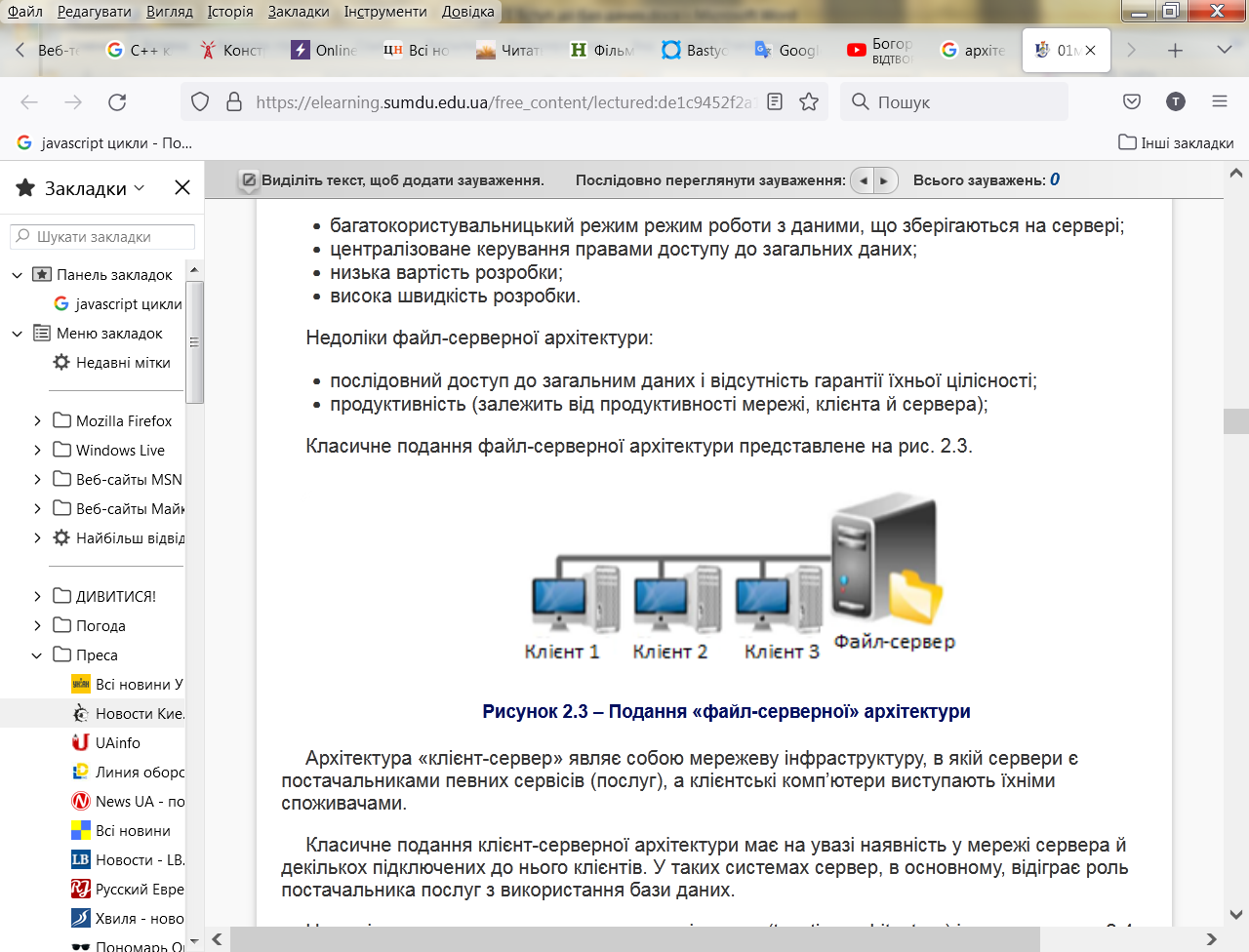


Рисунок 1 – Подання «файл-серверної» архітектури

Перевагою цієї архітектури є:.

багатокористувальницький режим роботи з даними, що зберігаються на сервері;

централізоване керування правами доступу до загальних даних;

низьке навантаження на процесор файлового сервера

низька вартість розробки;

висока швидкість розробки.

Недоліки:

потенційно високе завантаження локальної мережі;

ускладненість або неможливість централізованого управління;

ускладненість або неможливість забезпечення таких важливих характеристик як висока надійність, висока доступність і висока безпека.

послідовний доступ до загальним даних і відсутність гарантії їхньої цілісності;

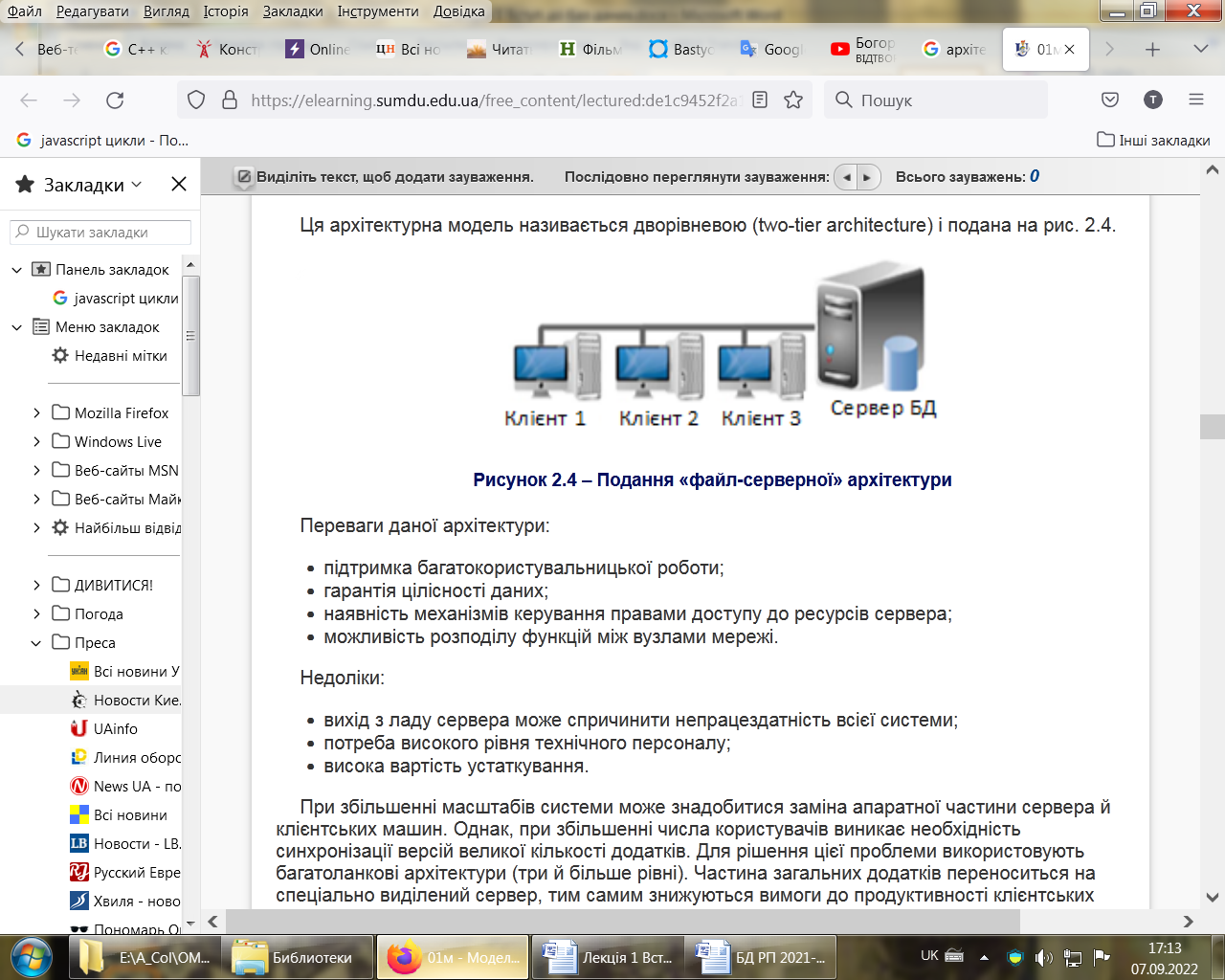
продуктивність (залежить від продуктивності мережі, клієнта й сервера).

Застосовуються найчастіше в локальних додатках, які використовують функції управління БД; в системах з низькою інтенсивністю обробки даних і низькими піковими навантаженнями на БД. виділення однієї з машин мережі в якості центральної (сервер). На такій машині зберігається спільно використовувана централізована БД. Усі інші машини мережі виконують функції робочих станцій, за допомогою яких підтримується доступ користувальницької системи до централізованої бази даних. Файли бази даних відповідно до призначених для користувача запитів передаються на робочі станції, де в основному і проводиться обробка. При великій інтенсивності доступу до одних [і тих же даних](http://compi.com.ua/metod-ta-zasib-heshuvannya-danih-na-osnovi-modelej-matematichn.html) [продуктивність](http://compi.com.ua/metod-ta-zasib-heshuvannya-danih-na-osnovi-modelej-matematichn.html) ІСпадає.

*Клієнт-серверна* СУБД розташовується на сервері разом з БД і здійснює доступ до БД безпосередньо, в монопольному режимі. Всі клієнтські запити на обробку даних обробляються клієнт-серверної СУБД централізовано.

Архітектура «клієнт-сервер» являє собою мережеву інфраструктуру, в якій сервери є постачальниками певних сервісів (послуг), а клієнтські комп’ютери виступають їхніми споживачами.

Класичне подання клієнт-серверної архітектури має на увазі наявність у мережі сервера й декількох підключених до нього клієнтів. У таких системах сервер, в основному, відіграє роль постачальника послуг з використання бази даних. Ця архітектурна модель називається дворівневою (two-tіer archіtecture) і подана на рис. 2.

 Рисунок 2. Подання «клієнт-серверної» архітектури

Переваги:

потенційно більш низьке завантаження локальної мережі;

зручність централізованого управління;

зручність забезпечення таких важливих характеристик як висока надійність, висока доступність і висока безпека.

підтримка багатокористувальницької роботи;

гарантія цілісності даних;

наявність механізмів керування правами доступу до ресурсів сервера;

можливість розподілу функцій між вузлами мережі.

Недолік клієнт-серверних СУБД полягає

в підвищених вимогах до сервера

вихід з ладу сервера може спричинити непрацездатність всієї системи;

потреба високого рівня технічного персоналу;

висока вартість устаткування.

У цій концепції мається на увазі, що крім зберігання централізованої БД центральна машина (сервер бази даних) повинна забезпечувати виконання основного обсягу обробки даних. Запит на дані, який видається клієнтом (робочою станцією), породжує пошук і вилучення даних на сервері. Витягнуті дані (але не файли) транспортуються по мережі від сервера до клієнта.

При збільшенні масштабів системи може знадобитися заміна апаратної частини сервера й клієнтських машин. Однак, при збільшенні числа користувачів виникає необхідність синхронізації версій великої кількості додатків. Для рішення цієї проблеми використовують багатоланкові архітектури (три й більше рівні). Частина загальних додатків переноситься на спеціально виділений сервер, тим самим знижуються вимоги до продуктивності клієнтських машин. Клієнти з низькою обчислювальною потужністю називають «тонкими клієнтами», а з високою продуктивністю – «товстими клієнтами». При багатоланковій архітектурі з виділеним сервером додатків існує можливість використання портативних пристроїв. Багатоланкова архітектура показана на рис. 3.

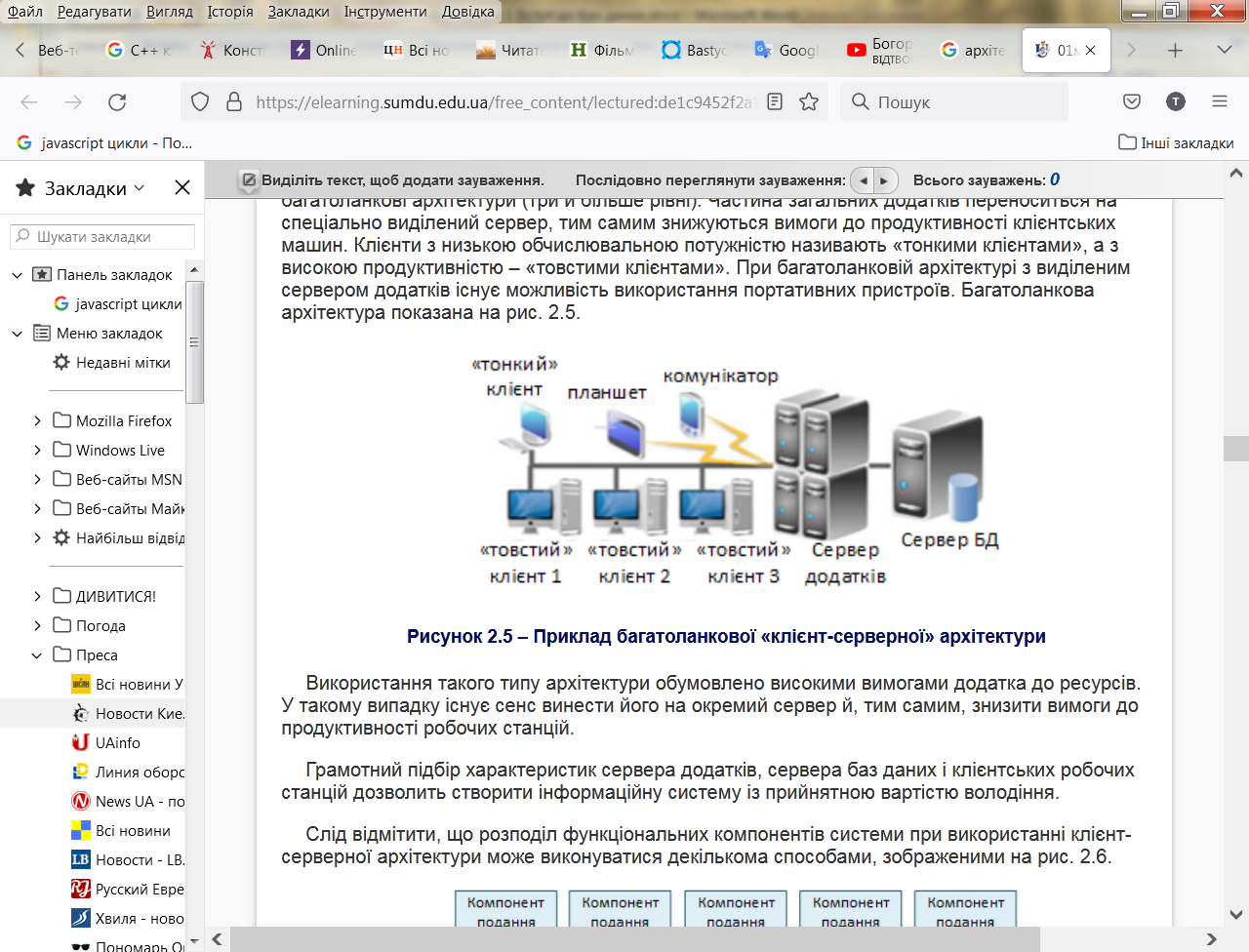


Рисунок 3. Приклад багатоланкової «клієнт-серверної» архітектури

Використання такого типу архітектури обумовлено високими вимогами додатка до ресурсів. У такому випадку існує сенс винести його на окремий сервер й, тим самим, знизити вимоги до продуктивності робочих станцій.

Грамотний підбір характеристик сервера додатків, сервера баз даних і клієнтських робочих станцій дозволить створити інформаційну систему із прийнятною вартістю володіння.

Типи баз даних

Класифікація БД *за моделлю даних*:

табличні

ієрархічні

мережеві

реляційні

об'єктні

об'єктно-орієнтовані

об'єктно-реляційні

хмарні

Класифікація БД за ступенем розподіленості:

централізовані (зосереджені)

розподілені

В *табличній БД* дані представлені у вигляді однієї таблиці. Таке буває дуже рідко. Таблична база даних, як випливає з назви, - це база даних, структурована в табличній формі. Він розташовує елементи даних у вертикальні стовпці та горизонтальні рядки. Кожна комірка утворюється перетином стовпця та ряду. Кожен рядок і стовпчик має унікальну нумерацію, щоб зробити його впорядкованим та ефективним.

Табличні бази даних містять такі ключові властивості:

Вони поділяють однаковий набір властивостей на запис. Це означає, що кожен рядок має однаковий набір стовпців. Кожному стовпцю зазвичай присвоюється метадані як його заголовок. Якщо в одному з рядків відсутні дані для певного стовпця, відсутнє значення, пов'язане з метаданими цього стовпця, буде збережено у цій комірці.

Вони отримують доступ до записів через ідентифікатори. Кожна таблиця в табличній базі даних містить певний набір пов’язаної інформації, яка пов'язана з темою бази даних через ключові поля, в яких описується кожен запис (рядок), щоб у разі запиту система змогла швидко знайти запис.

Кількість полів в табличних базах даних визначається розробником і не може змінюватися користувачем.

Будь-які поля повинні мати унікальне ім'я. Поля можуть мати різний тип:

рядок символів (довжиною до 255 символів);

дійсне число (з дробовою частиною);

ціле число;

грошова сума;

дата, час, дата і час;

логічне поле (істина або неправда, так або ні);

багатосторінковий текст (МЕМО);

рисунок, звук або інший об'єкт (об'єкт OLE).

Поля можуть бути обов'язковими для заповнення або ні. Таблиці можуть містити скільки завгодно записів (ця кількість обмежується тільки об'ємом диску); записи можна добавляти, вилучати, редагувати, сортувати, шукати.

Термінологія.

*Ключове* *поле* – це стовпець таблиці, що дозволяє встановити зв’язок із записами в іншій таблиці. Значення ключового поля однозначно визначає кожен запис у таблиці. Первинний ключ не допускає значень Null і завжди повинен мати унікальний індекс.

Існує три типи ключів:

Поле лічильника (Тип даних «Лічильник»). Тип даних поля в базі даних, у якому для кожного запису, що додається в таблицю, у полі автоматично заноситься унікальне числове значення.

Простий ключ. Якщо поле містить унікальні значення, такі як коди або інвентарні номери, то це поле можна визначити як первинний ключ. У якості ключа можна визначити будь-яке поле, що містить дані, якщо це поле не містить повторювані значення або значення Null.

Складений ключ. У випадках, коли неможливо гарантувати унікальність значень кожного поля, існує можливість створити ключ, що складається з декількох полів.

Прикладом табличної БД є табличний процесор *MS Excel.*

Ієрархічні бази

Однією з перших моделей даних була ієрархічна модель, запропонована компанією IBM в кінці 1960-х рр. в СУБД IMS (Information Management System). Така модель даних представляється набором дерев, пов'язаних один з одним за принципом побудови ієрархічних структур. В сучасних інформаційних системах такими ієрархічними моделями даних користуються при складанні структур в форматі XML, застосовуючи при передачі даних і роботі з даними в інтернет- системах.

Ієрархічнібази даних 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 головного (об'єкт більш близький до кореня) до нащадку (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-нащадку обов'язково тільки один предок. Об'єкти, що мають загального головного, називаються близнюками.

Організація даних в ієрархічної моделі передбачає обов'язкове поєднання батьківського і дочірнього об'єктів даних. Ці об'єкти даних характеризуються такими структурними елементами:

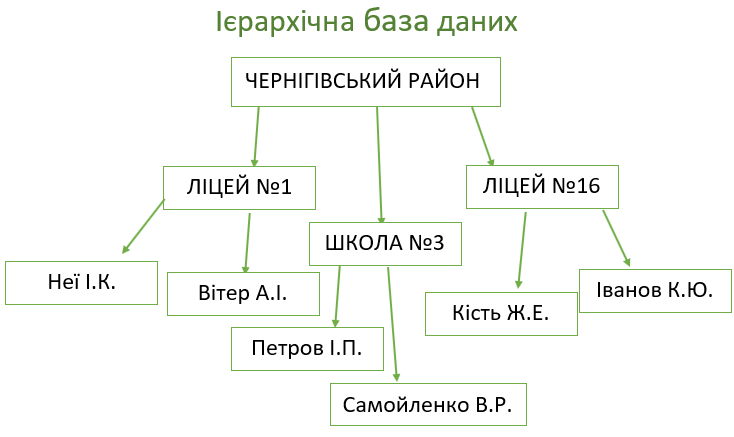
• атрибут - представляється найменшою одиницею елемента даних, що позначає функціональну сутність описуваного властивості об'єкта;

• запис - представляється групою атрибутів, що позначає конкретний екземпляр описуваного об'єкта;

• групове відношення- представляється зв'язком між записами різних типів, яка відображає взаємодію батьківської і дочірньої записів;

• ключовий елемент - представляється атрибутом, який може містити тільки унікальні значення для кожного запису.

З огляду на унікальність значень ключових елементів в ієрархічній моделі (рис. 1), для об'єкта даних некорневого рівня вона містить ключовий елемент, який визначається унікальністю значень в рамках групового відносини, що дозволяє однозначно виділити унікальну запис дочірнього об'єкта в межах конкретної записи батьківського об'єкта.



*Мал. 1.* Приклад ієрархічної моделі

Для групових відносин характерно важлива властивість, яка передбачає, що для зберігання будь-якої некорневой записи обов'язково повинна існувати батьківська запис. При цьому якщо видаляється батьківська запис, то автоматично повинні бути видалені всі дочірні записи. Ці властивості накладають дуже суттєве обмеження на використання моделі в реальних інформаційних системах. Наприклад, коли потрібно тільки накопичувати дані, то ієрархічна модель може бути варіантом подання відповідної структури. Також ієрархічна модель використовується при наявності ієрархічно організованих даних.

Прикладом такої організації може бути інтернет-сторінка або структура переданого повідомлення, що найчастіше організовується за допомогою ієрархічної схеми XML.

Поряд з істотними перевагами ієрархічної моделі виділяється кілька *недоліків*, серед яких відзначимо наступні:

- В реалізації ієрархічної моделі виникає дублювання відомостей (створення парних записів), при цьому в моделі не передбачена підтримка відповідності таких записів;

- В ієрархічній моделі реалізується тільки зв'язок типу 1: JV (один - до - багатьох), причому тільки в напрямку від батьківського елемента до дочірньому, котрі дають можливість створення зв'язків із зазначенням декількох батьківських елементів.

Так, наприклад, якщо виникне необхідність вказівки варіанта, коли договір складають кілька виконавців і один виконавець формує кілька договорів, тобто між договором і виконавцем використовується зв'язок *N: M* (багато - до - багатьох), то в ієрархічній моделі необхідно буде повторювати відомості про договори в якості дочірнього елемента до виконавця, що істотно ускладнює саму модель і не дає можливості коректно обробляти подану інформацію.

Ієрархічну БД можна перетворити в табличну форму

Мережеві бази даних

*Мережеві* бази даних 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 даних відносяться: рівень, елемент (вузол), зв'язок. Вузол - це сукупність атрибутів даних, що описують деякий об'єкт. На схемі ієрархічного дерева вузли представляються вершинами графа. У мережній структурі кожен елемент може бути пов'язаний з будь-яким іншим елементом.

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

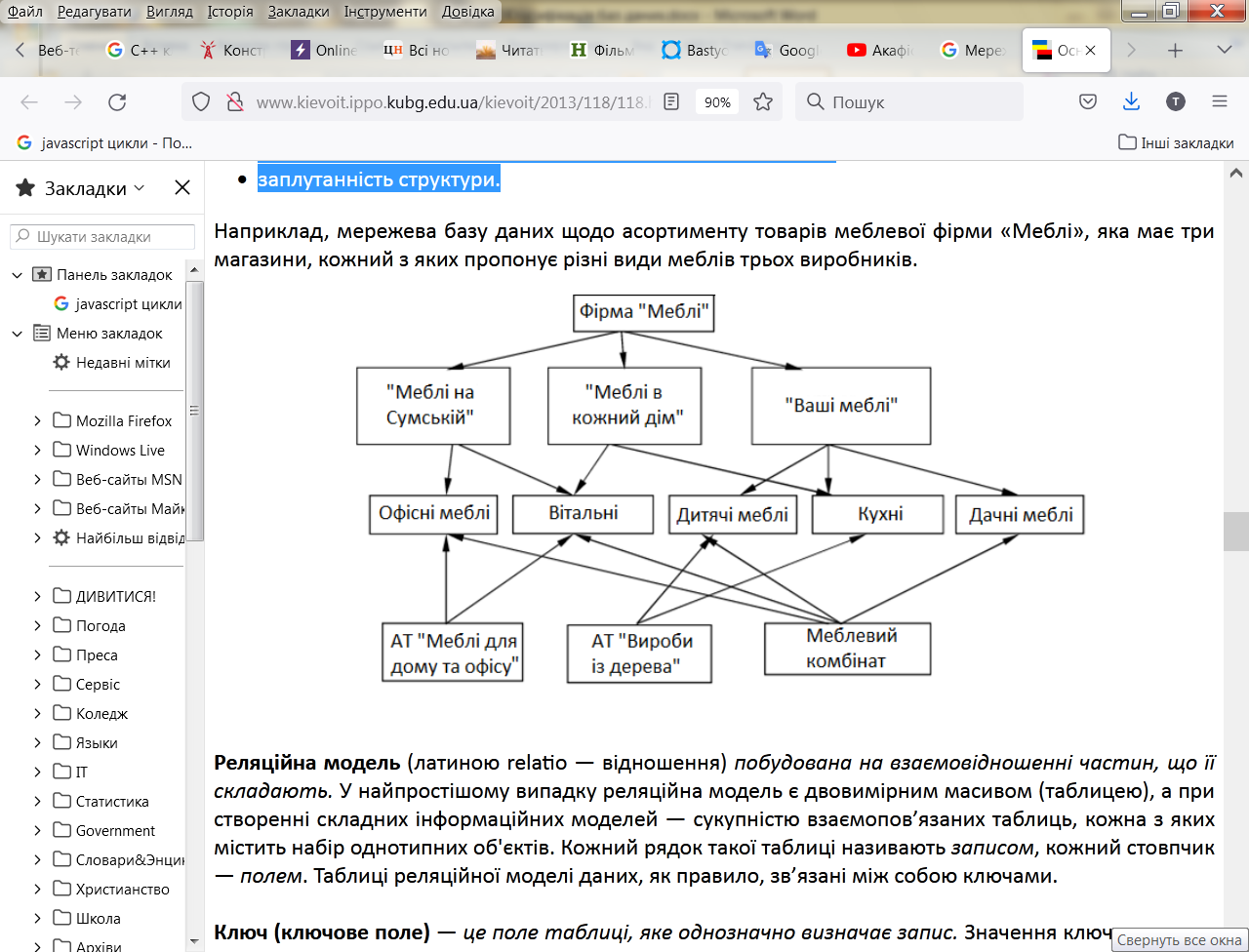
Також, оскільки логіка процедури вибірки даних залежить від фізичної організації цих даних, то ця модель не є повністю незалежною від програми. Іншими словами, якщо необхідно змінити структуру даних, то потрібно змінити і додаток.

Переваги мережевої моделі: найповніше відображення структури деяких задач. наприклад, мережевого планування в економіці.

Недоліки мережевої моделі

складність зберігання й пошуку інформації про всі зв'язки;

заплутанність структури.



*Мал. 1.* Приклад мережевої БД

*Реляційні бази даних*

Реляційна модель (латиною rela­tio — відношення) *побудована на взаємовідношенні частин, що її складають.* У найпростішому випадку реляційна модель є двовимірним масивом (таблицею), а при створенні складних інформаційних моделей — сукупністю взаємопов’язаних таблиць, кожна з яких містить набір однотипних об'єктів. Кожний рядок такої таблиці називають *записом*, кожний стовпчик — *полем*. Таблиці реляційної моделі даних, як правило, зв’язані між собою ключами.

Ключ (ключове поле) — *це поле таблиці, яке однозначно визначає запис.* Значення ключа не може повторюватись у таблиці. Ключі у базах даних відіграють украй важливу роль, адже саме за їх допомогою СКБД ідентифікує об'єкти.

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

кожен елемент таблиці - один елемент даних;

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 слідування рядків у таблиці може бути довільним і може характеризуватися кількістю полів, кількістю записів, типом даних.

Над цією моделлю бази даних зручно виконувати такі дії:

* сортування даних (наприклад за алфавітом);
* вибірка даних за групами (наприклад класами);
* пошук записів (наприклад за прізвищами) і т. д.

*Об'єктна* СУБД

*Об'єктна* СУБД ідеально підходить для інтерпретації складних даних, на відміну від реляційних СУБД, де додавання нового типу даних досягається ціною втрати продуктивності або за рахунок різкого збільшення термінів і вартості розробки додатків. Об'єктна база, на відміну від реляційної, не вимагає модифікації ядра при додаванні нового типу даних. Новий клас і його екземпляри просто надходять у зовнішні структури бази даних. Система управління ними залишається без змін.

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

*Об'єктно-орієнтовані бази даних*

Об'єктно-орієнтовані бази даних звичайно рекомендовані для тих випадків, коли потрібна високопродуктивна обробка даних, 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 даних і високорівневі непроцедурні мови запитів.

***Хмарна база даних*** — база даних, що, зазвичай, працює на платформі хмарних обчислень. Є дві поширені моделі розгортання: користувачі можуть запускати бази даних на хмарі незалежно, використовуючи віртуальну машину, або вони можуть отримати доступ до сервісу бази даних, що підтримується провайдером хмарних БД. З баз даних, доступних на хмарі, деякі базуються на SQL-основі, інші використовують [NoSQL](https://uk.wikipedia.org/wiki/NoSQL) модель даних.

*За технологією обробки даних* бази даних поділяються на централізовані й розподілені.

*Централізована* база даних зберігається у пам'яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом мережі ЕОМ, можливий розподілений доступ до такої бази. Такий спосіб використання баз даних часто застосовують у локальних мережах ПК.

*Розподілена* база даних 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 даних (СУРБД).

*За способом доступу* до даних бази даних поділяються на бази даних з локальним доступом і бази даних з віддаленим (мережевим) доступом

Області застосування баз даних

документографічні й документальні застосовуються у всіх базах органів влади та управління;

бази даних з промислової, будівельної та сільськогосподарської продукції;

бази даних з економічної та кон'юнктурної інформації (статистична, кредитно-фінансова, зовнішньоторговельна);

фактографічні бази соціальних даних, які включають відомості про населення і про соціальні середовища;

бази даних транспортних систем; літаків і поїздів, адреси та телефони громадян і організацій);

ресурсні бази даних, що включають фактографічну інформацію про природні ресурси (земля, вода, надра, біоресурси, гідрометеорологія, вторинні ресурси і відходи, екологічна обстановка);

фактографічні бази і банки наукових даних, щоб забезпечити фундаментальні наукові дослідження;

фактографічні бази даних у галузі культури і мистецтва;

лінгвістичні бази даних, тобто машинні словники різного типу і призначення.

**Нові області застосування баз даних**

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

*Інтелектуальний аналіз даних.*Ідея інтелектуально аналізу даних (data mining), тобто добування інформації з величезних масивів даних, накопичених зовсім для інших цілей.

Сховища даних.

У сховищі даних накопичуються дані з однієї або більше баз даних.

Репозитарії.

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

*Для самостійного вивчення*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література*.

1. *Дейт К.* Введение в системы баз данных : пер. с англ. / К. Дейт. – 8-е изд. – М., СПб.: Вильямс, 2005. – 1328 с.

2. *Мейер Д.* Теория реляционных баз данных / Д. Мейер. – М., Мир, 1987. – 608 с.

3. *Кузнецов С.Д.* Основы баз данных / С.Д. Кузнецов. – 2-е изд. – Москва: Бином, 2007. – 251 с.

*4. Васкевич Д.* Стратегии клиент/сервер / Д. Васкевич. – Киев: Диалектика, 1996. – 384 с.

*Запитання для самоперевірки.*

1. Визначте типи інформаційних систем
2. Визначте основні компоненти ІС
3. Які типи архітектурних систем ви знаєте?
4. Надайте опис багатоланкової «клієнт-серверної» архітектури
5. Які типи баз даних вам відомі.
6. Як поділяютьсябази даних *за технологією обробки даних?*
7. *Які нові області застосування БД ви можете визначити?*