**Лекція №1. Вступ до баз даних**

*План лекції:*

1. Вступ.

2. Традиційні файлові системи

3.Основні поняття та визначення.

4. Бази даних та банки даних.

5. Принципи організації БД та основні властивості.

6. Система керування баз даних.

7. Переваги і недоліки СУБД

8. Типи інформаційних систем

**Вступ**

База даних (БД) - сукупність певним чином організованої інформації на якусь тему (в рамках деякої предметної області), впорядкований за певними правилами набір взаємопов’язаних даних.

Учені запевняють, що зберігання великих обсягів даних виправдано тільки за умови, якщо пошук потрібних даних здійснюється швидко і подаються вони в доступній для розуміння формі. Ці умови забезпечують сучасні технології зберігання даних. Основою цих технологій є комп’ютеризовані бази даних.

Перша в Україні комп’ютерна база даних була розроблена в ході робіт з проектування і експлуатації електронної обчислювальної машини «Київ» (1959 р.). ЕОМ була розроблена для обчислювального центру Академії наук УРСР Л.Н. Дашевським, К.Л. Ющенко, К.О. Шкарабарою, С.Б. Погребинським під науковим керівництвом Б.В. Гніденка та В.М. Глушкова.

В історії розвитку СУБД можна виділити три основні етапи.

**Етап 1.** Цьому етапу відповідають СУБД першого покоління. Такі системи керування базами даних мали **ієрархічну та мережну моделі даних**. Ці моделі базувались на специфікаціях CODASYL. В СУБД першого покоління інформаційні системи базувались на мейнфреймах (mainframe) – великих обчислювальних машинах (наприклад, IBM 360). СУБД першого покоління мали ряд недоліків у порівнянні з СУБД наступних поколінь.

**Етап 2.** На цьому етапі з‘являються **реляційні СУБД** (початок 70-х років ХХ століття). На сьогоднішній день реляційні бази даних є одними з найбільш популярних. В таких базах даних інформація зберігається у вигляді таблиць, які зв‘язані між собою відношеннями (relations).

**Етап 3.** Цей етап характеризується появою **об’єктно-орієнтованих СУБД**, які базуються на використанні об’єктно-орієнтованої парадигми проектування баз даних.

**Недоліки** СУБД першого покоління порівняно з СУБД другого покоління:

* СУБД першого покоління були досить складними у використанні;
* відносно сильна залежність прикладних систем від фізичної організації бази даних;
* були відсутні засоби автоматизації проектування баз даних;
* СУБД першого покоління мали високу вартість.

**Переваги** СУБД першого покоління:

* СУБД першого покоління мали низькорівневі засоби керування даними у зовнішній пам‘яті;
* мали можливість будувати вручну прикладні системи;
* мали можливість економити пам‘ять за рахунок сумісного використання об‘єктів.

**Традиційні файлові системи**

Незважаючи на те, що файлові системи баз даних давно застаріли, усе-таки є кілька причин, чому з ними варто познайомитися, оскільки розуміння проблем, які властиві файловим системам, може запобігти їх повторення в системах керування базами даних (СКБД або СУБД - системах управління БД). Знати принципи роботи файлових систем не тільки дуже корисно, але і необхідно при виконанні переходу від файлової системи до системи баз даних.

Підхід, який використовується у файлових системах БД

*Файлові системи БД* - набір програм, що виконують для користувачів деякі операції, наприклад створення звітів. Кожна програма визначає свої власні дані і керує ними.

Основні обмеження, властиві файловим системам

• Поділ і ізоляція даних.

Коли дані ізольовані в окремих файлах, доступ до них дуже ускладнений. Виконувати подібну обробку даних у файлових системах досить складно. Наприклад, якщо потрібна інформація зберігається в двох файлах, то для витягу відповідної поставленим умовам інформації програміст повинний організувати синхронну обробку двох файлів. Труднощі істотно зростають, коли необхідно витягти дані з більш ніж двох файлів. Із за децентралізованої роботи з даними у файловій системі фактично заохочується безконтрольне дублювання даних, і це, у принципі, неминуче. Безконтрольне дублювання даних небажано по наступним двом причинам:

• Дублювання даних.

Дублювання даних супроводжується неощадливою витратою ресурсів, оскільки на введення надлишкових даних потрібно витрачати додаткові час і гроші. Більш того, для їхнього збереження необхідно додаткове місце в зовнішній пам'яті, що зв'язано з додатковими накладними витратами. У багатьох випадках дублювання даних можна уникнути за рахунок спільного використання файлів.

Ще більш важливим є той факт, що дублювання даних може привести до порушення їхньої цілісності. Інакше кажучи, дані, у різних відділах можуть стати суперечливими. Наприклад, є два файли для бухгалтерії та відділу кадрів, і, відповідно, дублювання даних у розрахунковому секторі бухгалтерії і відділі кадрів. Якщо співробітник переїде в інший будинок і зміну адреси буде зафіксовано тільки у відділі кадрів, то повідомлення про зарплату буде надіслано йому за старою адресою, тобто помилковою. При виявленні подібної помилки для її виправлення буде потрібно витратити додатковий час і засоби. Це демонструє протиріччя, що може виникнути при дублюванні даних. Оскільки не існує ніякого автоматичного способу відновлення даних одночасно у файлах відділу кадрів, і у файлах розрахункового сектору, неважко передбачати, що подібні протиріччя час від часу обов'язково будуть виникати. Навіть якщо співробітники розрахункового сектора після одержання повідомлень про подібні зміни будуть негайно їх вносити, усе рівно існує імовірність неправильного введення змінених даних.

• Залежність від даних.

Як уже згадувалося вище, фізична структура і спосіб збереження записів файлів даних жорстко зафіксовані в коді програм. Це значить, що змінити існуючу структуру даних досить складно.

• Несумісність файлів.

Оскільки структура файлів визначається кодом програми, вона також залежить від мови програмування цієї програми. Наприклад, структура файлу, створеного програмою мовою Pyton, може зовсім відрізнятися від структури файлу, створеного програмою мовою С++. Пряма несумісність таких файлів ускладнює процес їхньої спільної обробки.

• Фіксовані запити/швидке збільшення кількості програм.

Файлові системи багато в чому залежать від програміста, тому що всі необхідні запити і звіти повинні бути створені саме їм.

Усі перераховані вище обмеження файлових систем є наслідком двох факторів.

1. Визначення даних міститься усередині програмного коду, а не зберігається окремо і незалежно від них.

2. Крім програм не передбачено ніяких інших інструментів доступу до даних і їх обробки.

Для підвищення ефективності роботи постала необхідність використовувати новий підхід, а саме базу даних (database) і систему керування базами даних, чи СУБД (Database Management System — DBMS), також використовується абревіатура СУБД (система керування базами даних).

Основні поняття та визначення

*Інформація* – це будь-які відомості та/або дані, які можуть бути збережені на матеріальних носіях або відображені в електронному вигляді.

(ст. 1 Закону України “Про інформацію”).

*Інформація* є одиницею даних, яка підлягає обробці. *Інформація* надходить споживачеві саме у вигляді даних: таблиць, графіків, малюнків, фільмів, усних повідомлень, які фіксують у собі інформацію певної структури й типу.

*Дані* – це *інформація*, представлена в певному вигляді, яка дозволяє автоматизувати її збір, зберігання і подальшу обробку людиною або інформаційним засобом.

Банки даних та бази даних.

*База даних (БД)* – це засіб накопичення і організації великих масивів інформації про об’єкти деякої предметної області (ПО).

*Предметна область* - це цілеспрямована первинна трансформація картини зовнішнього світу у деяку картину, певна частина якої фіксується в інформаційній системі (ІС) як алгоритмічна модель фрагменту дійсності.

БД має відображати поточні дані про предметну область, накопичувати, зберігати інформацію і надавати різним категоріям користувачів швидкий доступ до даних. Для цього дані в базі мають бути структуровані відповідно до деякої моделі, що відображає основні об’єкти ПО, їх властивості і зв’язки між ними.

БД є частиною складної системи, яка називається *банком даних* або *системою баз даних* (СБД).

Відмінною рисою баз даних слід вважати те, що дані зберігаються разом з їх описом, а в прикладних програмах опис даних не міститься. Незалежні від програм користувача дані зазвичай називаються метаданими.

Головними перевагами організації інформації у вигляді БД є:

Багаторазовість використання

Економія витрат на створення і введення

Зменшення надлишковості

Швидкість обробки

Простота і зручність внесення змін до БД

Логічна та фізична незалежність даних від прикладних програм.

Існує кілька визначень БД.

База даних (за Джорджем Мартіном) є сукупністю взаємозалежних даних, які спільно використовуються декількома додатками й зберігаються з мінімальною регульованою надлишковістю. Дані запам'ятовуються таким чином, щоб вони у міру можливості не залежали від програм.

*База даних* (БД) - це поіменована сукупність структурованих даних, які стосуються певної предметної області. (Прикладом бази даних є бібліотечний каталог, записна книжка, класні журнали, журнали обліку товарів на підприємствах та ін.)

Часто БД ще називають *набором інтегрованих записів із самоописом.* У сукупності, опис даних називається системним каталогом (system catalog), чи словником даних (data dictionary), а самі елементи опису прийняті називати мета-даними (meta-data), тобто "даними про дані". Саме наявність само опису даних у базі даних забезпечує в ній незалежність між програмами і даними (program-data independence).

Банк даних є різновидом *інформаційної системи*, в якій реалізовані функції централізованого зберігання і накопичення оброблюваної інформації, організованої в одну або декілька баз даних.

Банк даних в загальному випадку складається з наступних компонентів:

* бази (декількох баз) даних;
* системи управління базами даних;
* словника даних;
* адміністратора;
* обчислювальної системи;
* обслуговуючого персоналу.

**Принципи організації БД та основні властивості**.

Підхід, заснований на застосуванні баз даних, де визначення даних відділене від програм, дуже схожий на підхід, який використовується при розробці сучасного програмного забезпечення, коли поряд із внутрішнім визначенням об'єкта існує його зовнішнє визначення. Користувачі об'єкта бачать тільки його зовнішнє визначення і не піклуються про те, як він визначається і як функціонує. Одне з переваг такого підходу, а саме абстрагування даних (data abstraction), полягає в тім, що можна змінити внутрішнє визначення об'єкта без яких-небудь наслідків для його користувачів, за умови, що зовнішнє визначення об'єкта залишається незмінним. Аналогічним образом, у підході з використанням БД, структура даних відділена від програм і зберігається в БД. Додавання нових структур даних чи зміна існуючих ніяк не впливає на програми, за умови, що вони не залежать безпосередньо від змінюваних компонентів. Наприклад, додавання нового поля в запис чи створення нового файлу ніяк не вплине на роботу наявних програм. Однак видалення поля з використовуваного програмою файлу вплине на цю програму, а тому його також буде потрібно відповідним чином модифікувати.

Терміни, пов’язані з БД.

*Сутністю* (entity) називається окремий тип об'єкта організації (людина, місце або річ, поняття або подія), який потрібно представити в базі даних.

*Атрибутом* (attribute) називається властивість, що описує деяку характеристику описуваного об'єкта;

*Зв'язок* (relationship) — це те, що поєднує кілька сутностей.

БД представляє сутності, атрибути і логічні зв'язки між об'єктами, тобто містить логічно зв'язані дані.

Робочі характеристики БД:

* *повнота –* чим повніша база даних, тим ймовірніше, що вона містить потрібну інформацію (проте, не має бути надмірної інформації);
* *правильна організація –* чим краще структурована база даних, тим легко в ній знайти необхідні відомості;
* *актуальність –* будь-яка база даних може бути точною і повною, якщо вона постійно оновлюється, тобто необхідно, щоб база даних у кожен момент часу повністю відповідала стану об’єкту, що відображався нею;
* *зручність для використання –* база даних має бути проста і зручна у використанні і мати розвинені методи доступу до будь-якої частини інформації.

Система керування / управління баз даних (СУБД/СУБД).

Обробляє структуровані дані централізований [програмний механізм](http://ua-referat.com/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D0%9C%D0%B5%D1%85%D0%B0%D0%BD%D1%96%D0%B7%D0%BC), який називається системою управління базами даних.

*Система управління базами даних* (СУБД) - це програмний механізм, призначений для запису, пошуку, [сортування](http://ua-referat.com/%D0%A1%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), обробки (аналізу) та друкування інформації, що міститься в базі даних.

Функції СУБД:

– управління буферами оперативної пам'яті;

– управління транзакціями;

– захист від відмов і відновлення (журнализація);

– забезпечення різних рівнів доступу до даних.

Можливості СУБД:

* дозволяє визначати базу даних за допомогою мови визначення даних;
* дозволяє вставляти, оновлювати, видаляти, витягувати інформацію з бази даних за допомогою мови управління даними;
* надає контрольований доступ до бази даних за допомогою системи забезпечення безпеки та системи підтримки цілісності даних;
* забезпечує додатковий рівень безпеки;
* надає механізм налаштування зовнішнього інтерфейсу бази даних;
* дозволяє зберігати зовнішній інтерфейс бази даних несуперечливим і незмінним навіть при внесенні змін до її структуру.

Основні компоненти СУБД:

* апаратне забезпечення;
* програмне забезпечення;
* дані;
* процедури;
* користувачі.

СУБД - це програмне забезпечення, за допомогою якого користувачі можуть визначати, створювати і підтримувати базу даних, а також здійснювати до неї контрольований доступ. СУБД - це програмне забезпечення, що взаємодіє з прикладними програмами користувача і базою даних і має наведеними нижче можливості:

* Дозволяє визначати базу даних, що звичайно здійснюється за допомогою мови визначення даних (DDL - Data Definition Language). Мова DDL надає користувачам засоби призначати типи даних і їх структури, а також засоби завдання обмежень для інформації, збереженої в базі даних.
* Дозволяє вставляти, обновляти, видаляти і витягати інформацію з бази даних, що звичайно здійснюється за допомогою мови керування даними (DML - Data Manipulation Language). Наявність централізованого сховища всіх даних і їх описів дозволяє використовувати мову DML як загальний інструмент організації запитів, який іноді називають мовою запитів (query language). Наявність мови запитів дозволяє усунути властивим файловим системам обмеження, при яких користувачам приходиться мати справа тільки з фіксованим набором запитів чи постійно зростаючою кількістю програм, що породжує інші, більш складні проблеми керування програмним забезпеченням.

Існує два різновиди мов DML - процедурні (procedural) і непроцедурні (non-procedural) мови, - які відрізняються між собою способом витягу даних. Основна відмінність між ними полягає в тім, що процедурні мови звичайно обробляють інформацію в базі даних послідовно, запис за записом, а непроцедурні оперують відразу цілими наборами записів. Тому за допомогою процедурних мов DML звичайно вказується, *як* можна одержати бажаний результат, тоді як непроцедурні мови DML використовуються для опису того, *що* варто одержати. Найбільш розповсюдженим типом непроцедурної мови є мова структурованих запитів (Structured Query Language — SQL), що у даний час визначається спеціальним стандартом і *фактично* є обов'язковою мовою для будь-яких реляційних СУБД. (SQL вимовляється або по буквах "S-Q-L", або як мнемонічне ім'я See-Quel").

Надає контрольований доступ до бази даних за допомогою засобів:

* системи забезпечення безпеки, що запобігають несанкціонований доступ до бази даних з боку користувачів;
* системи підтримки цілісності даних, що забезпечують несуперечливий стан збережених даних;
* системи керування рівнобіжною роботою програм, що контролюють процеси їх спільного доступу до бази даних;
* системи відновлення, що дозволяють відновити базу даних до попереднього несуперечливого стану, порушеного в результаті збою апаратного чи програмного забезпечення;
* доступного користувачам каталогу, що містить опис збереженої в базі дані інформації.

На відміну від варіанта з файловою системою, фізична структура і спосіб збереження даних у цьому випадку контролюються за допомогою СУБД.

Володіння зазначеними вище функціональними можливостями перетворює СУБД у надзвичайно корисний інструмент. Однак, оскільки для кінцевих користувачів неважливо, наскільки проста чи складна внутрішня організація системи, можна почути заперечення, що СУБД ускладнює роботу, надаючи користувачам набагато більшу кількість даних,чим їм дійсно потрібно. Для рішення проблеми щодо доступу користувачів до різних розрізів даних в СУБД пропонується **механізм** - створення **представлень** (view) - які дозволяє будь-якому користувачу **мати** свій власний погляд на базу даних. Мова DDL включає засобу визначення представлень, кожне **з** який є деякою підмножиною бази даних.

Крім спрощення роботи за рахунок надання користувачам тільки дійсно потрібнихїм даних, представлення мають деякі інші достоїнства:

* Забезпечують додатковий рівень безпеки. Представлення можуть створюватися з метою виключення тих даних, що не повинні бачити деякі користувачі. Наприклад, можна створити деяке представлення, що дозволить менеджерам відділень і співробітникам розрахункового сектора бухгалтерії переглядати всі дані про персонал, включаючи зведення про їх зарплату. У той же час для організації доступу до даних інших користувачів можна створити ще одне представлення, з якого всі зведення про зарплату будуть виключені.
* Надають механізм настроювання зовнішнього інтерфейсу бази даних. Наприклад, співробітники відділу контрактів можуть працювати з полем Monthly Rent (Щомісячна орендна плата), використовуючи для нього більш коротке і просте ім'я - Rent.
* Дозволяють зберігати зовнішній інтерфейс бази даних несуперечливим і незмінним навіть при внесенні змін у її структуру - наприклад, при додаванні чи видаленні полів, зміні зв'язків, розбивці файлів, їх реорганізації чи перейменуванні. Якщо у файл додаються чи з нього віддаляються поля, які не використовуються в деякім представленні, то всі ці зміни на даному представленні ніяк не відіб'ються. Таким чином, представлення забезпечує повну незалежність програм від реальної структури даних, що дозволяє усунути найважливіший недолік файлових систем.

Приведені вище міркування мали трохи загальний характер. Насправді реальний обсяг функціональних можливостей, пропонованих у деякої конкретний СУБД, відрізняється від продукту до продукту. Наприклад, у СУБД для персонального комп'ютера може не підтримуватися рівнобіжний спільний доступ, а керування режимом безпеки, підтримка цілісності даних і відновлення будуть присутні тільки в дуже обмеженій ступені. Однак сучасні потужні СУБД для багатьох користувачів пропонують усі перераховані вище функціональні можливості і багато чого іншого. Сучасні системи являють собою надзвичайно складне програмне забезпечення, що складається з мільйонів рядків коду і багатьох томів документації. Такий результат прагнення одержати програмне забезпечення, що могло б задовольняти вимогам усе більш загального характеру. Більш того, у даний час використання СУБД припускає майже 100-процентну надійність і готовність навіть при збоях в апаратному і програмному забезпеченні. Програмне забезпечення СУБД постійно удосконалюється і повинне усе більше і більше розширюватися, щоб задовольняти все новим вимогам користувачів. Наприклад, у деяких програмах потрібно зберігати графіку, відео, звук і т.д. Для охоплення цієї частини ринку СУБД повинна еволюціонувати, причому згодом їй, імовірно, буде потрібно виконувати якісь нові функції, а тому функціональна частина СУБД ніколи не буде статичною. Більш докладно основні функції СУБД будемо розглядати в ході вивчення предмету.

**Переваги і недоліки СУБД**

Переваги:

* контроль за надмірністю даних;
* несуперечність даних;
* більше корисної інформації при тому ж обсязі збережених даних;
* спільне використання даних;
* підтримка цілісності даних;
* підвищена безпека;
* застосування стандартів;
* підвищення ефективності зростанням масштабів системи;
* можливість знаходження компромісу при суперечливих вимогах;
* підвищення готовності даних до роботи;
* підвищення доступності даних;
* покращення показників продуктивності;
* спрощення супроводу системи за рахунок незалежності від даних;
* покращене керування паралельністю;
* розвинені служби резервного копіювання та відновлення.

Недоліки:

* складність;
* розмір;
* вартість СУБД;
* додаткові витрати на апаратне забезпечення;
* витрати на перетворення;
* продуктивність;
* серйозні наслідки при виході системи з ладу.

**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.

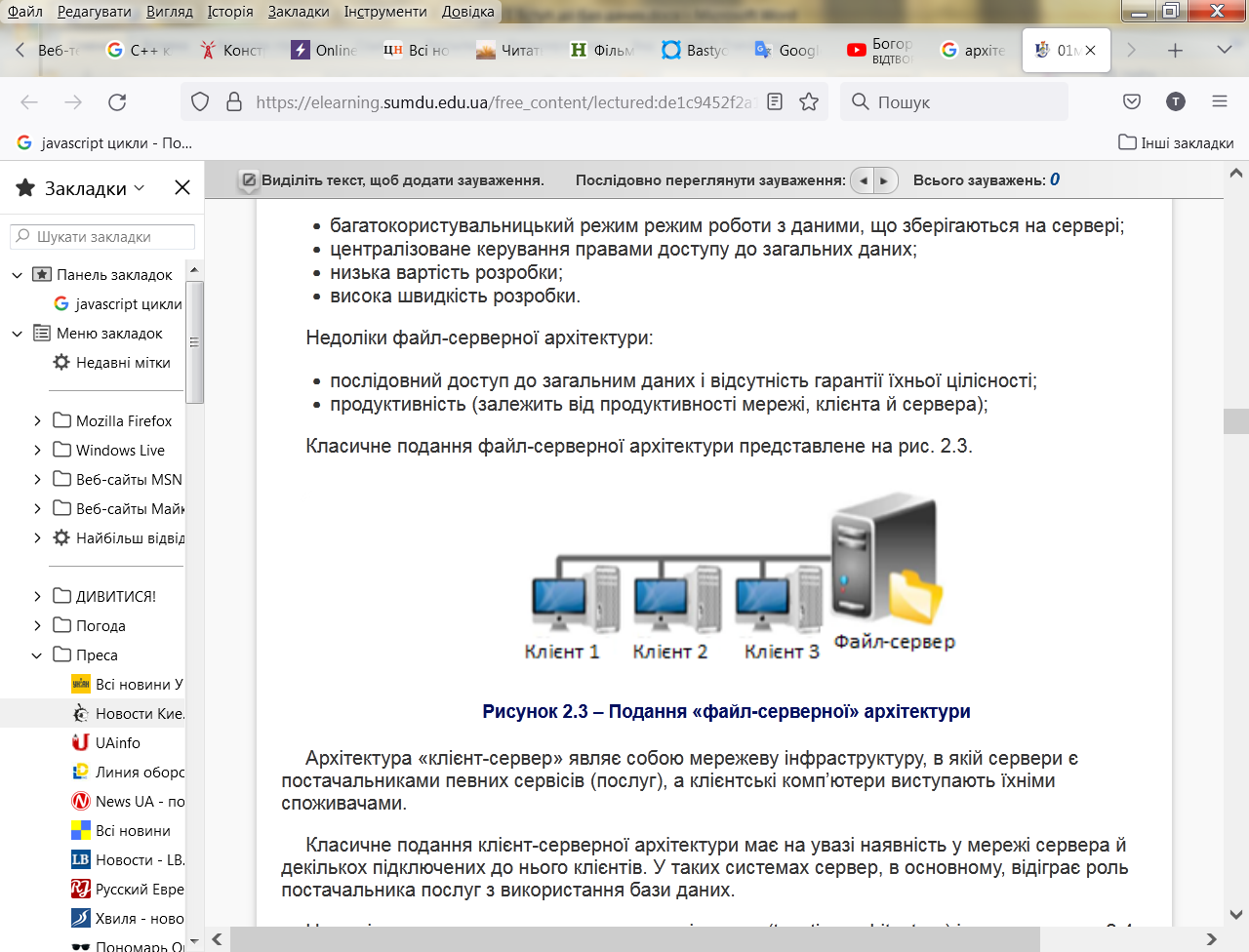
****

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:

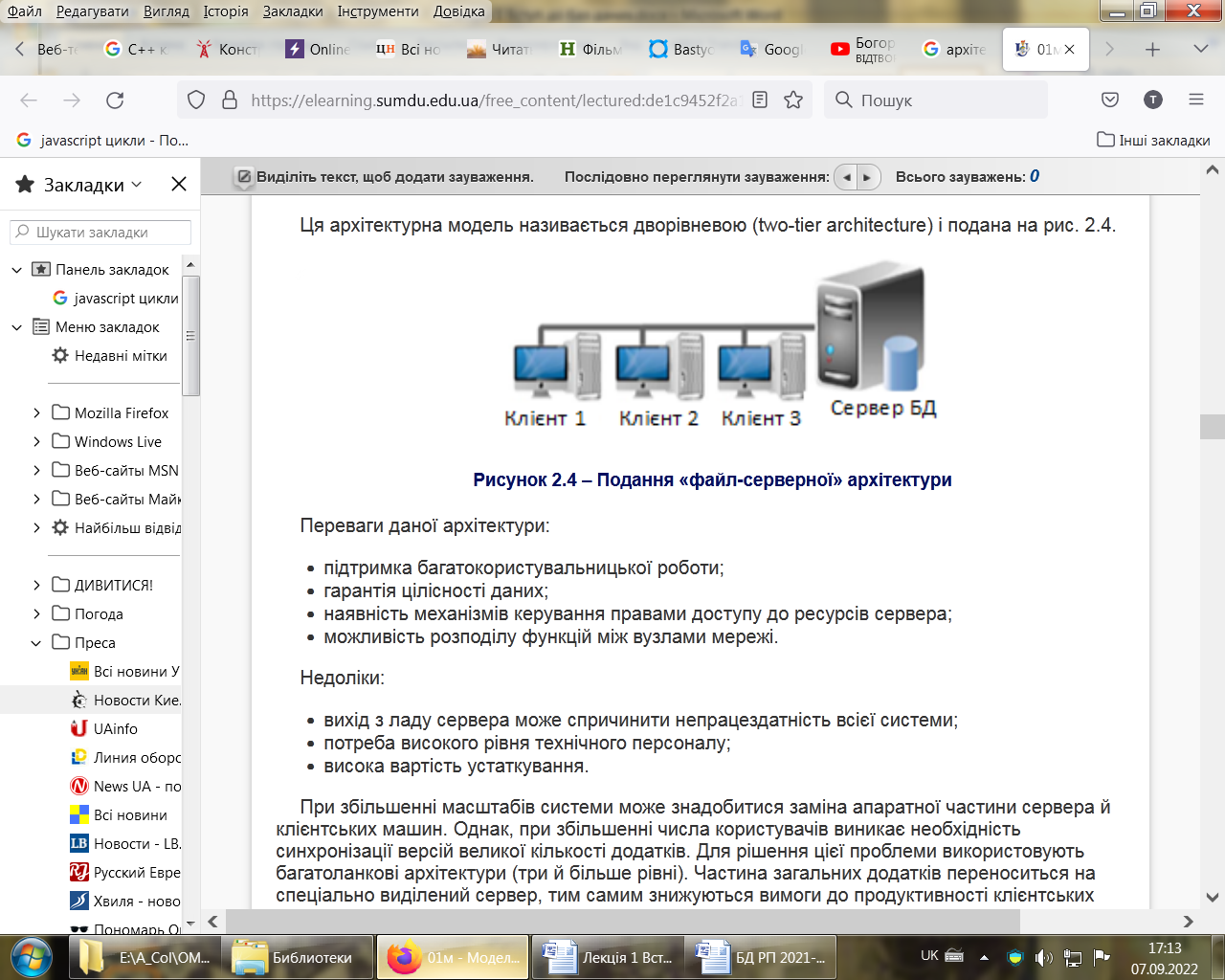
* потенційно високе завантаження локальної мережі;
* ускладненість або неможливість централізованого управління;
* ускладненість або неможливість забезпечення таких важливих характеристик як висока надійність, висока доступність і висока безпека.
* послідовний доступ до загальним даних і відсутність 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.

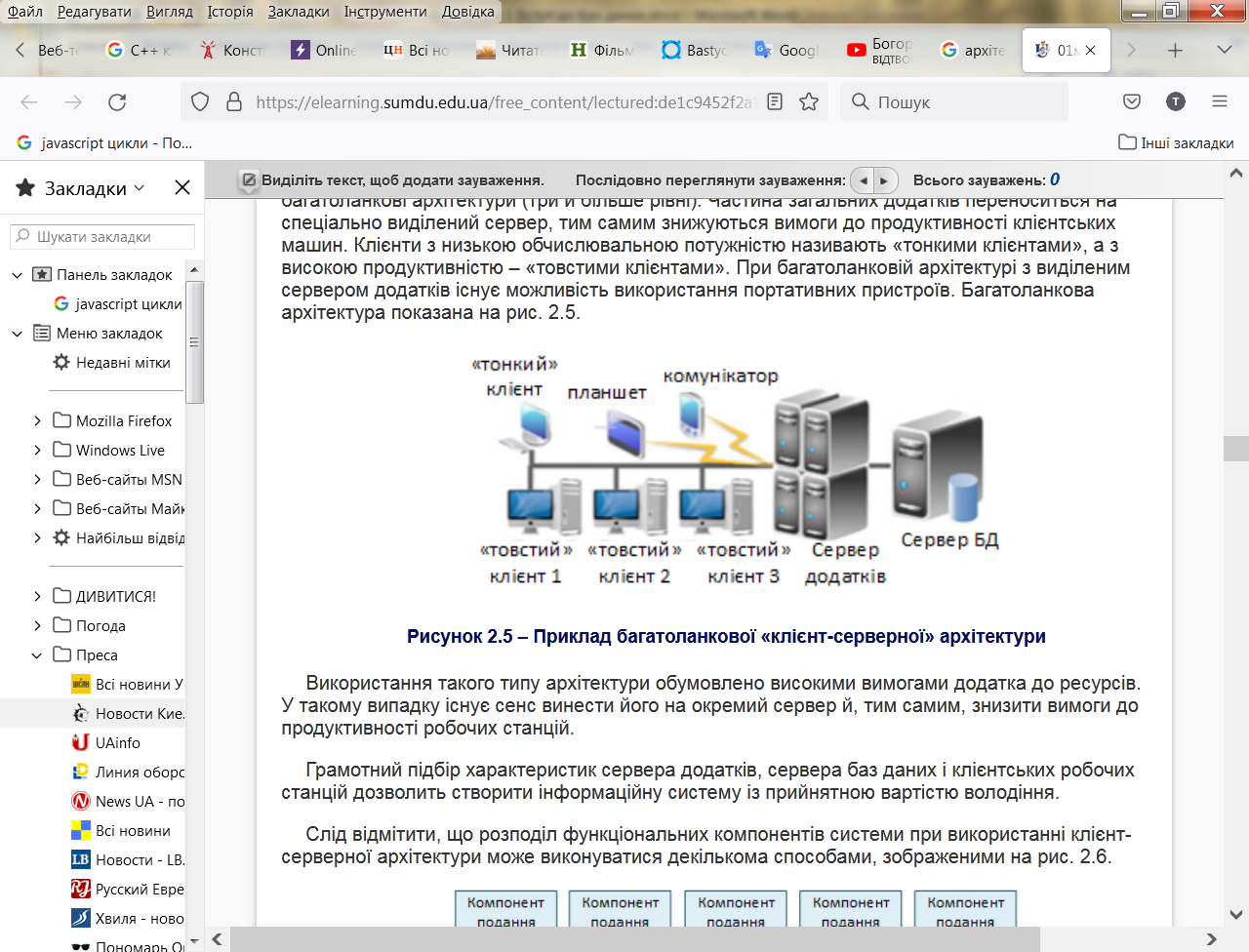


Рисунок 3. Приклад багатоланкової «клієнт-серверної» архітектури

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.

*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. Роз’яснити значення наступних термінів: «дані», «база даних», «система управління базами даних», «незалежність від даних», «представлення», «цілісність» та «безпека».

2. Описати підхід, що використовується в файлових системах. Вказати основні недоліки даного підходу.

3. Описати основні характеристики підходу, який базується на використанні бази даних..

4. Які основні компоненти СУБД вам відомі?

5. Пояснити ролі різних користувачів СУБД.

6. Назвати основні переваги СУБД.

7. Назвати основні недоліки СУБД.