**Лекція 6 Фізична організація баз даних**

***Ресурси баз даних***

Основні ідеї сучасної інформаційної технології базуються на концепції, згідно з якою дані мають бути організовані в бази даних з метою адекватного відображення реального світу, що постійно змінюється, і задоволення інформаційних вимог користувачів.

База даних - це сукупність логічно пов'язаних даних, організованих на машинному носії засобами системи управління даними (СУБД). Система управління базою даних - це комплекс програм, за допомогою якого користувачі можуть визначати, створювати і підтримувати базу даних, а також здійснювати до неї контрольований доступ.

База даних включає дані, які відображають певну логічну модель взаємопов'язаних інформаційних об'єктів, що представляють конкретну предметну область. База даних організується відповідно до моделі і структури даних, які підтримує СУБД. Організація даних у пам'яті комп'ютера характеризується двома рівнями - логічним і фізичним.

***Організація даних***

База даних передбачає певну організацію даних у пам'яті комп'ютера. Фізична організація даних визначає спосіб розміщення даних безпосередньо на машинному носії. У сучасних прикладних програмних засобах цей рівень організації забезпечується автоматично без втручання користувача. Користувач, як правило, оперує в прикладних програмах та універсальних програмних засобах уявленнями про логічну організацію даних.

Логічна організація даних на машинному носії залежить від програмних засобів організації і ведення даних, які використовуються у внутрішньо-машинній сфері. Логічна організація даних визначається типом структур даних і видом моделі, яку використовує і підтримує програмний засіб.

Основою організації бази даних є модель даних, яка визначає правила, відповідно до яких структуруються дані.

Модель даних - це сукупність взаємопов'язаних структур даних і операцій над ними.

За допомогою моделі даних описується взаємозв'язок між елементами даних. Вид моделі і типи структур даних, що використовуються в ній, відображають концепцію організації та обробки даних, яка є основою СУБД, що підтримує модель. Для розміщення тієї самої інформації у внутрішньо-машинній сфері можуть бути використані різні структури і моделі даних. їх вибір покладається на користувача, що створює інформаційну базу, і залежить від багатьох факторів, у тому числі й від технічного та програмного забезпечення, визначається складністю задач, що автоматизуються, та обсягом інформації.

Найпоширенішими є такі моделі даних: файлова, ієрархічна, мережева, реляційна.

***Файлова модель***

У файлових системах реалізується модель типу плоский файл. За цієї моделі внутрішньомашинна інформаційна база (ІБ) є сукупністю не пов'язаних між собою незалежних файлів з однотипних записів з лінійною (однорівневою) структурою. Основними елементами структур даних файлової моделі є поле, запис, файл.

*Поле* - елементарна одиниця логічної організації даних, яка відповідає окремій, неподільній одиниці інформації - реквізиту.

*Запис* - сукупність полів, що відповідають логічно пов'язаним реквізитам. Структура запису визначається складом і послідовністю полів, кожне з яких містить елементарне дане. Запис є основною структурною одиницею обробки даних і одиницею обміну між оперативною і зовнішньою пам'яттю.

*Файл* - сукупність однакових за структурою примірників записів зі значеннями окремих полів.

Організацію даних за файловою моделлю можна проаналізувати на прикладі інформаційної бази (ІБ) двох відділів фірми (реалізації і контрактів), що працюють із нерухомістю. ІБ відділу реалізації, що відповідає за оренду та продаж нерухомості, може бути представлена трьома файлами - Об'єкти\_ Нерухомості, Власники\_Об'єктів, Орендарі (у дужках наведено перелік полів кожного файла):

* Об'єкти\_нерухомості (Код\_об'єкта, Адреса, Тип\_об'єкта, Кіль-кість\_кімнат, Орендна\_плата, Код\_власника);
* Власники\_об'єктів (Код\_власника, Прізвище\_власника, Ім'я\_власника, Адреса, №\_телефону);
* Орендарі (Код\_орендаря, Прізвище\_орендаря, Ім'я\_орендаря, Адреса, №\_телефону, Тип\_нерухомості, Мах\_орендна\_плата).

ІБ відділу контрактів, що відповідає за укладення угод про оренду нерухомості, може бути представлена такими трьома файлами - Угоди, Об'єкти\_нерухомості, Орендарі:

* Угоди (№\_угоди, Код\_об'єкта, Код\_орендаря, Орендна\_плата, Платіжний\_інструмент, Завдаток, Оплата, Дата\_початку\_оренди, Кінцева\_Дата\_оренди, Термін\_оренди);
* Об'єкти \_нерухомості (Код\_об'єкта, Адреса\_об'єкта, Орендна\_ плата);
* Орендарі (Код\_орендаря, Прізвище\_орендаря, Ім'я\_орендаря,

Адреса, №\_телефону, Тип\_нерухомості, Мах\_орендна\_плата).

Схему обробки файлів наведено на рис. 1.

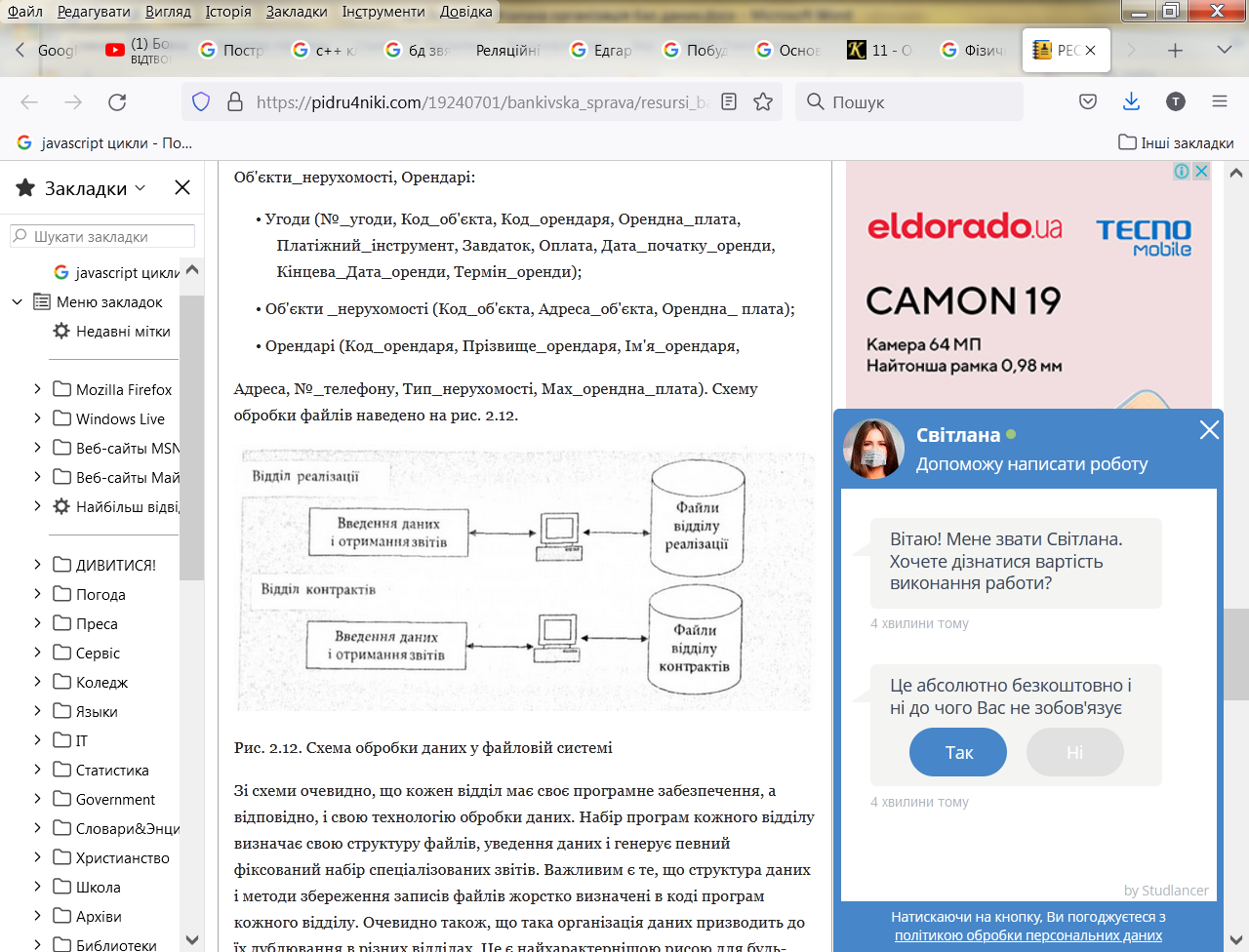


Рис. 1. Схема обробки даних у файловій системі

Зі схеми очевидно, що кожен відділ має своє програмне забезпечення, а відповідно, і свою технологію обробки даних. Набір програм кожного відділу визначає свою структуру файлів, уведення даних і генерує певний фіксований набір спеціалізованих звітів. Важливим є те, що структура даних і методи збереження записів файлів жорстко визначені в коді програм кожного відділу. Очевидно також, що така організація даних призводить до їх дублювання в різних відділах. Це є найхарактернішою рисою для будь-яких файлових систем.

Суть притаманних обмежень файлових систем:

* розподіл та ізоляція даних - дані ізольовані в окремих файлах, що ускладнює їх синхронну обробку;
* дублювання даних - децентралізована робота з даними фактично стимулює безконтрольне дублювання даних, що, у принципі, неминуче (в ІБ відділу реалізації та ІБ відділу контрактів дублюються дані про об'єкти нерухомості та орендарів);
* залежність від даних, фізична структура і спосіб збереження записів файлових даних жорстко зафіксовані в коді програм застосувань. Це означає, що змінити структуру даних досить складно. Наприклад, щоб збільшити у файлі Об'єкти\_нерухомості довжину поля Адреса об'єкта з 35 до 36 символів (незначна зміна його структури), треба написати програму спеціального призначення (та одноразового використання) для перетворення файлу Об'єкти нерухомості в новий формат;
* несумісність форматів файлів - оскільки структура файлів визначається кодом програми, вона також залежить від мови програмування. Наприклад, структура файлу, що створена мовою Cobol, може зовсім відрізнятись від структури файлу, створеного мовою С. Пряма несумісність форматів таких файлів ускладнює процес їх сумісної обробки.

Така організація даних є неефективною і вимагає пошуку інших рішень, нових підходів. На рис. 2 наведено приклад реалізації підходу до обробки даних з використанням СУБД.

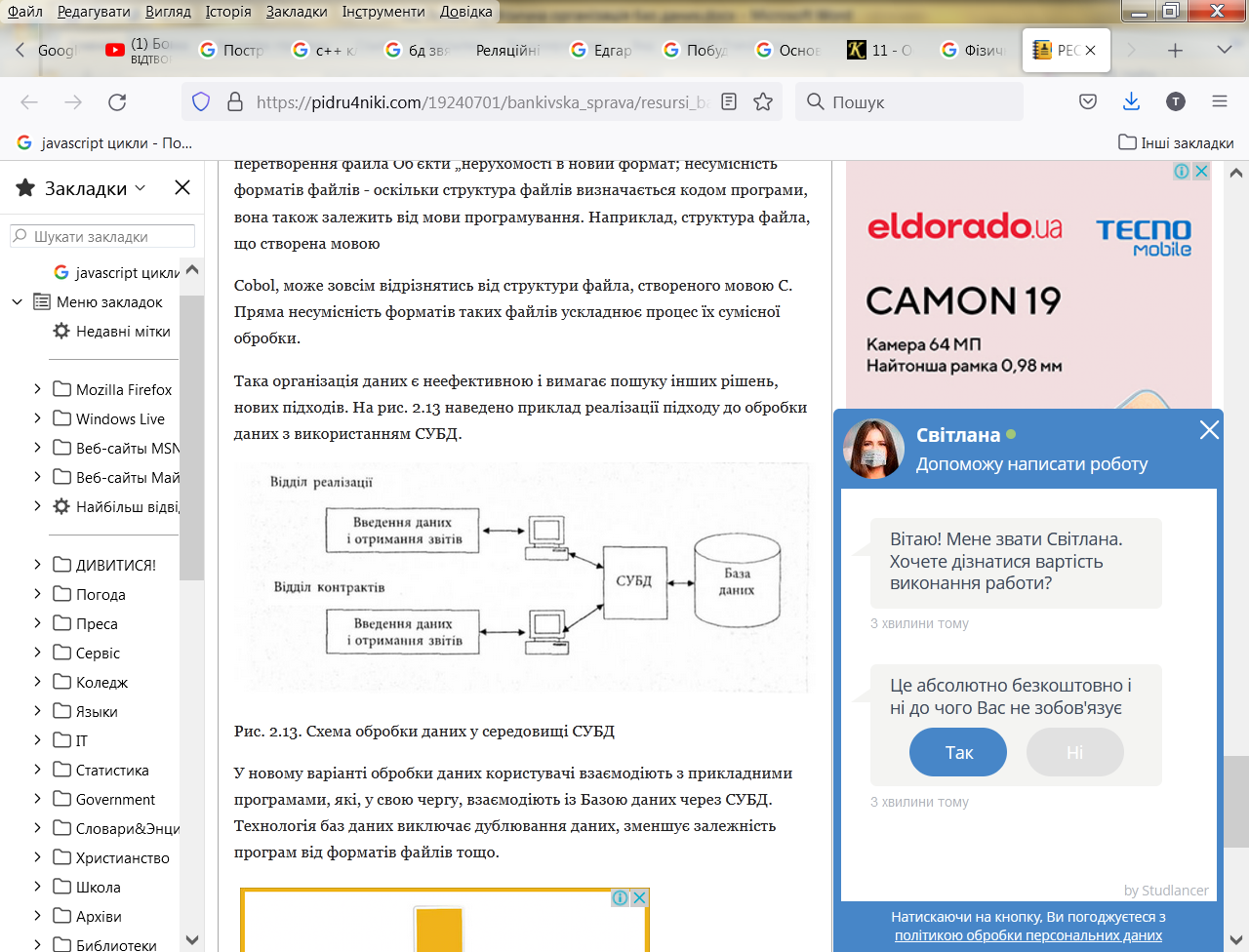


Рис. 2. Схема обробки даних у середовищі СУБД

У новому варіанті обробки даних користувачі взаємодіють з прикладними програмами, які, у свою чергу, взаємодіють із Базою даних через СУБД. Технологія баз даних виключає дублювання даних, зменшує залежність програм від форматів файлів тощо.

Фізична організація даних відповідає за їх зберігання, управління, форми представлення і структури даних.

*Фізичне проектування* являє собою процес визначення характеристик сховища даних і доступу до них в БД. Властивості сховища даних залежать від пристроїв зберігання, засобів доступу до даних, що підтримуються системою і від СУБД. На етапі фізичного проектування визначається місцезнаходження даних на пристроях зберігання, загальна продуктивність системи.

Фізична організація складається з 3 складових: організація інформаційних масивів; організація пошукових структур (індексів); організація обміну інформації в системі.

БД на рівні ОС розглядається як набір файлів. Кожен файл має структуру. Крім опису файлу створюється системний каталог, в якому зберігається інформація про таблицю, обмеження в таблиці та ін.

Програма користувача звертається до БД за допомогою диспетчера файлів та диспетчера дисків.

Процес пошуку і представлення даних користувачу виконується в декілька етапі:



Рис. 3. Організація доступу до даних

Диспетчер файлів-розглядає БД як набір сторінок фіксованого розміру з унікальним ідентифікатором.

Диспетчер дисків-визначає фізичне місцезнаходження сторінки.

Для організації зберігання даних застосовується технологія *кластерізації -* фізично близьке розташування записів у просторі пам'яті середовища зберігання БД.

Організація інформаційних масивів.

1.послідовна (записи послідовно)

запис

3

2

n

2.організація у вигляді двосполученого списку

0

0

0

Є показник на попередній та наступний запис.

3.організація у вигляді зв’язаних блоків записів

*Читання і запис інформації.* Виконується блоками, вони мають фіксовану довжину ,блоки ще називаються сторінками. Для кожного файлу даних зберігається мета-інформація, де є дані про структуру файлу. На рівні БД існує словник даних в якому зберігається інформація про всі атрибути, таблиці та їх обмеження.

Організація інформаційних масивів: послідовна, у вигляді двосполученого списку, у вигляді зв’язаних блоків записів. Читання і запис інформації на вінчестер виконується саме блоками (сторінками). Блоки мають фіксовану довжину.

Вилучення інформації розрізняється на фізичне і логічне. Фізичне – інформація фізично вилучається із зовнішнього носія; логічне – інформація на зовнішньому носії, але до неї не можна отримати доступ.. Для кожного файлу даних зберігається мета-інформація, де є дані про структуру файлу. На рівні БД існує словник даних, в якому зберігається інформація про всі таблиці, атрибути та їх обмеження.

*Пам'ять сторінок* - організація простору зовнішньої або віртуальної пам'яті в БД, яка передбачає поділ простору пам'яті на сторінки фіксованого розміру. Для розміщення сторінок, що викликаються, в оперативній пам'яті використовуються спеціальні області - *буфери.* Оновлений в буферах зміст сторінок повертається у зовнішню пам'ять. Розмір сторінок, кількість буферів, алгоритми вибору буферів для розміщення сторінок суттєво впливають на ефективність роботи системи.

Компоненти системи БД, які призначені для зберігання даних, мають різну ємність і різну швидкість доступу до даних. Організація ієрархії пам'яті комп'ютера має наступний вигляд:

Кеш-пам’ять - це швидка буферна пам’ять не великого обсягу, яка зберігає останні дані, до яких виконувався доступ. Для роботи з даними в БД застосовується віртуальна пам’ять, яка моделюється за допомогою апарату і програмних засобів. Диск поділяється на логічні одиниці - блоки. Дані між оперативною пам’яттю і дисковою передаються блоками. Ці блоки на рівні операційної системи називаються сторінками. Архітектура пам’яті визначається наступними чинниками: яким чином дані переміщуються з одного рівня на інший; кешування- тимчасовим зберіганням копій даних на певному рівні.

Кеш-пам’ять

Оперативна пам’ять

Дискова пам’ять

Третична пам’ять(пам’ять на магнітних стрічках)

Hot Date

Cold Date

OnLine

OffLine

Керовані дані дозволяють: прискорити процес обробки інформації; мінімізувати число блокувань на сервері БД.

***Захист баз даних***

***Захист бази даних*** – забезпечення захищеності бази даних проти будь-яких навмисних або ненавмисних загроз за допомогою різних комп'ютерних та некомп'ютерних засобів.

***Небезпека*** – будь-яка ситуація або подія, навмисна або ненавмисна, яка здатна несприятливо вплинути на систему і, як наслідок, на всю організацію.

Потенційні небезпеки:

* викрадення та фальсифікація даних;
* втрата конфіденційності;
* порушення недоторканості особових даних;
* втрата цілісності;
* втрата доступності.

**Комп’ютерні засоби контролю:**

* авторизація користувачів;
* представлення;
* створення резервних копій та відновлення;
* підтримка цілісності;
* шифрування;
* допоміжні процедури.

***Авторизація*** – надання прав (або привілеїв), що дозволяють їх власнику мати законний доступ до системи або її об'єктів.

***Аутентифікація*** – механізм визначення того, чи є користувач тим, за кого себе видає.

***Представлення*** – це динамічний результат однієї або декількох реляційних операцій з базовими відношеннями з метою створення деякого іншого відношення.

***Резервне копіювання*** – періодично виконувана процедура отримання копії бази даних і її файлу журналу на носії, збереженому окремо від системи.

***Ведення журналу*** – процедура створення та обслуговування файлу журналу, що містить відомості про всі зміни, внесені в базу даних з моменту створення останньої резервної копії, та призначеного для забезпечення ефективного відновлення системи у випадку її відмови.

***Контрольна точка*** – момент синхронізації між станом бази даних та станом журналу виконання транзакцій.

***Підтримка цілісності*** – запобігання переходу даних у неузгоджений стан.

***Шифрування*** – кодування даних з використанням спеціального алгоритму, внаслідок чого дані стають недоступними для читання будь-якою програмою, яка не має ключа дешифрування.

***Допоміжні процедури*** для захисту даних у середовищі СУБД:

* аудит;
* встановлення нового прикладного програмного забезпечення;
* встановлення або модернізація системного програмного забезпечення.

**Некомп’ютерні засоби контролю:**

* заходи забезпечення безпеки та планування захисту від непередбачених обставин;
* контроль за персоналом;
* захист приміщень та сховищ;
* гарантійні угоди;
* договори про супровід;
* контроль за фізичним доступом.

***Захист ПК***

Способи організації захисту персонального комп’ютера:

* використання індивідуальних ідентифікаторів користувачів та відповідних особистих паролів;
* спеціальні сховища для захищеного зберігання будь-яких необхідних даних, програмного забезпечення та комп'ютерного обладнання;
* особиста відповідальність кожного користувача за створення копій даних та зберігання програмного забезпечення і комп'ютерного обладнання;
* чіткі вказівки про процедури, які повинні застосовуватися до будь-якого програмного забезпечення, перед тим як воно буде допущено до встановлення системи.

***СУБД та захист у Web***

Заходи захисту, пов'язані з використанням СУБД у середовищі Web:

* проксі-сервери;
* брандмауери;
* використання цифрового підпису;
* алгоритми створення дайджесту повідомлень цифрового підпису;
* цифрові сертифікати;
* використання церберів;
* використання технологій SSL та SHTTP;
* концепції мов програмування для захисту Web-додатків;
* засоби захисту технології ActiveX.

***Проксі-сервер*** – це комп'ютер, який розміщується між Web-браузером та Web-сервером. У загальному випадку проксі-сервери призначені для вирішення задач підвищення продуктивності та фільтрації запитів.

***Брандмауер*** – це система, призначена для захисту від несанкціонованого доступу до (або з) закритої мережі. Брандмауери можуть бути програмними, апаратними або комбінованими.

Основні типи брандмауерів:

* пакетний фільтр;
* шлюз додатка;
* шлюз на рівні ланцюга;
* проксі-сервер.

***Цифровий підпис*** використовується для перевірки істинності походження даних та складається з двох частин: рядка бітів, який обчислений на основі "підписаних" даних, та закритого ключа приватної особи або організації, яка "підписала" ці дані.

***Цифровий сертифікат*** – це доповнення електронного повідомлення, яке використовується з метою забезпечення безпеки, – найчастіше для того, щоб довести, що відправник повідомлення дійсно є тим, за кого себе видає.

***Цербер*** – це сервер захищених облікових імен та паролів користувачів. Цербер ідентифікує і встановлює справжність користувача.

***Протокол Secure Sockets Layer*** (SSL) – це протокол, який створює безпечне з'єднання між клієнтом та сервером, по якому може безпечно передаватися будь-яка кількість даних.

***Протокол Secure НТТР*** (S-HTTP) – це протокол, який призначений для безпечної передачі окремих повідомлень.

***Концепція "пісочниця"*** гарантує заборону на доступ до системних ресурсів несанкціонованого і, можливо, зловмисного додатку. Для реалізації концепції "пісочниці" використовуються завантажувач класів, верифікатор байт-коду та менеджер захисту.

***ActiveX*** – це фреймворк для визначення програмних компонентів, придатних до використання з програм, написаних на різних мовах програмування.

***Управління транзакціями***

Основні функції сучасної СУБД:

* механізми підтримки транзакцій;
* служби управління паралельністю;
* засоби відновлення баз даних.

Засоби відновлення та механізм управління паралельністю доступу призначені для захисту баз даних від переходу даних в неузгоджений стан або їх втрати.

**Підтримка транзакцій**

***Транзакція*** – це дія або серія дій, що виконуються одним користувачем або прикладною програмою і здійснюють доступ бази даних або змінюють її вміст.

1. Транзакція є логічною одиницею роботи, що виконується в базі даних і може бути представлена окремою програмою, частиною алгоритму програми або окремою командою та включати довільну кількість операцій, які виконуються в базі даних.
2. Будь-яка транзакція завжди повинна переводити базу даних з одного узгодженого стану в інший, хоча допускається, що узгодженість стану бази буде порушуватися під час виконання транзакції.
3. Будь-яка транзакція завершується одним з двох можливих способів: COMMIT (фіксація) або ROLLBACK (відкат).
4. Жодна СУБД не володіє внутрішньою можливістю встановити, які саме зміни повинні бути сприйняті як єдине ціле, що утворить одну логічну транзакцію.
5. ***COMMIT*** – це процес фіксації результатів транзакції в базі даних у разі її успішного завершення.
6. ***ROLLBACK*** – це процес відновлення бази даних, тобто приведення її в той узгоджений стан, в якому вона перебувала до початку виконання транзакції, що відбулося невдало.

***Компенсуюча транзакція*** – це транзакція, яка скасовує дії, виконані зафіксованою помилковою транзакцією.

Для позначення границь окремих транзакцій використовуються наступні оператори: BEGIN TRANSACTION, COMMIT і ROLLBACK (або їх еквіваленти). Якщо ці обмежувачі не були використані, вся виконувана програма розцінюється як єдина транзакція і СУБД автоматично виконає команду СОММIТ при нормальному завершенні цієї програми. Аналогічно, у разі її аварійного завершення в базі даних автоматично буде виконана команда ROLLBACK.

Основні властивості транзакцій:

* *Атомарність* – будь-яка транзакція є неподільною одиницею роботи, яка може бути або виконана вся цілком, або не виконана зовсім;
* *Узгодженість* – кожна транзакція повинна переводити базу даних з одного узгодженого стану в інший узгоджений стан;
* *Ізольованість* – усі транзакції виконуються незалежно одна від одної;
* *Тривалість* – результати зафіксованої транзакції повинні зберігатися в базі даних постійно і не повинні бути втрачені в результаті наступних збоїв.

**Управління паралельністю**

***Управління паралельністю*** – це процес організації одночасного виконання в базі даних різних операцій, що гарантує виключення їх взаємного впливу один на одного.

***Протокол управління паралельністю*** – це протокол, який реалізується в СУБД для запобігання небажаного впливу призначених для користувача процесів один на одного.

Потенційні проблеми, які можуть мати місце при паралельному виконанні транзакцій:

* проблема втраченого оновлення;
* проблема залежності від нефіксованих результатів;
* проблема неузгодженої обробки.

**Впорядкованість і відновлення БД**

***Впорядкованість*** – засіб, здатний допомогти у виявленні тих транзакцій, які гарантовано не викличуть порушення узгодженості даних при одночасному виконанні.

***Графік*** – послідовність запуску операцій багатьох паралельно виконуваних транзакцій, що зберігає черговість виконання операцій в кожній окремій транзакції.

***Послідовний графік*** – графік, в якому операції кожної з транзакцій виконуються строго послідовно і не можуть чергуватися з операціями, що виконуються в інших транзакціях.

***Непослідовний графік*** – графік, в якому чергуються операції з деякого набору одночасно виконуваних транзакцій.

Суть впорядкування полягає у знаходженні таких непослідовних графіків, які дозволять транзакціям виконуватися паралельно, але без надання взаємовпливу одна на одну, і привести базу даних в стан, який може бути досягнуто при використанні послідовного графіка.

Порядок виконання операцій читання і запису даних задля досягнення впорядкованості:

* якщо дві транзакції тільки зчитують деякий елемент даних, вони не будуть конфліктувати між собою і порядок їх виконання не має значення;
* якщо дві транзакції зчитують або записують абсолютно незалежні елементи даних, вони не будуть конфліктувати між собою і порядок їх виконання не має значення;
* якщо одна транзакція записує елемент даних, а інша транзакція цей самий елемент даних зчитує або записує, порядок їх виконання має істотне значення.

Типи впорядкування:

* конфліктне впорядкування;
* впорядкування за переглядом.

***Конфліктно впорядкований графік*** – це графік, в якому порядок виконання будь-яких конфліктуючих операцій відповідає їх розміщенню в послідовному графіку. Для перевірки конфліктної впорядкованості можна використовувати граф передування.

***Впорядкований за переглядом графік*** – це графік, еквівалентний за переглядом деякому послідовному графіку.

Кожен конфліктно впорядкований графік в той же час є впорядкованим за переглядом, проте зворотнє твердження невірне.

***Впорядковані графіки*** – це графіки, які дозволяють зберегти узгодженість бази даних в припущенні, що жодна з транзакцій цього графіка не буде скасована.

***Відновлюваний графік*** – графік, в якому для кожної пари транзакцій *Тi* і *Tj* виконується правило: якщо транзакція *Тi* зчитує елемент даних, попередньо записаний транзакцією *Tj*, то фіксація результатів транзакції *Тi* повинна виконуватися до фіксації результатів транзакції *Tj*.

Методи управління паралельністю підтримують консервативні та оптимістичні підходи.

***Консервативні (песимістичні) методи*** – це методи, які відкладають виконання транзакцій, здатних в майбутньому в той чи інший момент часу увійти в конфлікт з іншими транзакціями.

***Методи блокування*** – це методи, які зазвичай усувають можливі конфлікти за допомогою переведення транзакцій в стан очікування.

***Блокування*** – процедура, яка використовується для управління паралельним доступом до даних і дозволяє відхилити спроби отримання доступу до даних з боку інших транзакцій у разі, коли деяка транзакція вже отримала до них доступ.

Фундаментальний принцип методів блокування: транзакція повинна вимагати виконати блокування для читання або для запису деякого елемента даних перед тим, як вона зможе виконати в базі даних відповідну операцію читання або запису.

***Протокол двофазного блокування (2PL)*** – це протокол, який визначає моменти встановлення та зняття блокування для кожної з транзакцій і використовується для забезпечення впорядкованості. Фундаментальний принцип двофазного блокування: транзакція виконується по протоколу двофазного блокування, якщо в ній всі операції блокування передують першій операції розблокування.

***Методи з використанням часових міток*** – це методи, які не передбачають будь-якого очікування – залучені в конфлікт транзакції просто скасовуються, після чого запускаються заново.

***Часова мітка*** – унікальний ідентифікатор, який створюється СУБД з метою позначення відносного моменту часу запуску транзакції.

***Метод використання часових міток*** – протокол управління паралельністю, основна мета якого – встановлення глобальної черговості виконання транзакцій, при якій транзакції з меншим значенням часової мітки мають більш високий пріоритет при розв’язанні виникаючих конфліктів.

***Правило запису Томаса*** – це модернізований базовий протокол впорядкування за часовими мітками, який дозволяє досягти менш жорсткої конфліктної впорядкованості за рахунок скасування застарілих операцій запису.

***Правило ігнорування застарілого запису*** – це ситуація, коли новіша транзакція вже перезаписала значення елемента даних і значення, яке старіша транзакція збирається записати в даний елемент, було обчислено на основі застарілого вихідного значення даного елемента, тому операція запису цілком безпечно може бути проігнорована.

***Оптимістичні методи*** – це методи, які будуються на припущенні, що ймовірність конфлікту невисока, і тому допускають асинхронне виконання транзакцій, а перевірка на наявність конфлікту відкладається на момент їх завершення і фіксації в базі даних.

Фази оптимістичного підходу: читання, перевірки, запису.

Оптимістичні технології ефективні в разі незначної кількості конфліктів в системі, але можуть викликати відкат окремих транзакцій, при цьому виникнення каскадного відкату виключається.

***Відновлення бази даних*** – це процедура повернення бази даних до деякого коректного (узгодженого) стану, що вживається в разі руйнування системи.

Причини руйнування системи:

* відмови обладнання;
* програмні помилки;
* збої носіїв інформації.

Функції відновлення бази даних:

* механізм резервного копіювання;
* засоби ведення журналу;
* створення контрольних точок;
* менеджер відновлення.

***Контрольна точка*** – момент синхронізації між базою даних і журналом реєстрації транзакцій.

Методи відновлення:

* метод відновлення з використанням відкладеного поновлення;
* метод відновлення з використанням негайного поновлення;
* метод тіньових сторінок.

**Покращені моделі транзакцій**

***Модель вкладених транзакцій*** – це модель, яка розглядає транзакцію, як набір пов'язаних підзадач (субтранзакції), кожна з яких також може складатися з довільної кількості субтранзакцій.

***Хроніка*** – це послідовність (пласких) транзакцій, які можуть чергуватися з іншими транзакціями.

***Модель багаторівневих транзакцій*** – це модель, яка вимагає, щоб завершення роботи субтранзакцій виконувалося від низу до верху, в напрямку транзакції верхнього рівня.

***Модель відкритих вкладених транзакцій*** – це модель, в якій атомарность порушується і часткові результати виконання субтранзакцій можуть бути доступні поза транзакцією.

***Динамічна реструктуризація*** – це модель, яка об’єднує операції розбиття і об'єднання транзакцій

***Робочий потік*** – це вид діяльності, який передбачає координоване виконання множини завдань, що здійснюються різними суб'єктами обробки.

Запитання

1. Пояснити призначення та область застосування поняття «захист бази даних».

2. Перелічити шість типів небезпек, яким можуть піддаватися системи з базами даних, і вказати для кожної з них можливі засоби контролю та протидії.

3. Навести приклади зв’язку між можливими варіантами порушень системи захисту та їх наслідками.

4. Пояснити наступні поняття: авторизація користувачів, резервне копіювання, шифрування, захист від непередбачених обставин, контроль за персоналом, гарантійні угоди, недоторканість особових даних, захист особистих даних.

5. Які засоби контролю належать до комп’ютерних?

6. Які засоби контролю належать до некомп’ютерних?

7. Які заходи захисту, пов'язані з використанням СУБД у середовищі Web, вам відомі?

8. Що таке транзакція?

9. Які властивості транзакцій вам відомі?

10. Що таке протокол управління паралельністю?

11. Які потенційні проблеми можуть мати місце при паралельному виконанні транзакцій?

12. Що таке впорядкованість?

13. Яким має бути порядок виконання операцій читання і запису даних задля досягнення впорядкованості?

14. Які типи впорядкування вам відомі? В чому їх особливості?

15. Які методи управління паралельністю вам відомі?

16. Що таке відновлення бази даних?

17. Які функції відновлення типової СУБД вам відомі?

18. Які вам відомі покращені моделі транзакцій? В чому їх особливості?