1. **Поняття бази даних. Основні властивості.**

База даних (БД) -- це сукупність взаємозв’язаних даних, що зберігаються разом. Основними та невід’ємними властивостями БД є такі:

- для даних допускається така мінімальна надлишковість, яка сприяє їх оптимальному використанню в одному чи кількох застосуваннях;

- незалежність даних від програм;

- для пошуку та модифікації даних використовуються спільні механізми;

- як правило, у складі БД існують засоби для підтримки її цілісності та захисту від неавторизованого доступу

1. **Інформаційна модель концептуального рівня. ER-модель.**

База даних (БД) -- це сукупність взаємозв’язаних даних, що зберігаються разом. Основними та невід’ємними властивостями БД є такі:

- для даних допускається така мінімальна надлишковість, яка сприяє їх оптимальному використанню в одному чи кількох застосуваннях;

- незалежність даних від програм;

- для пошуку та модифікації даних використовуються спільні механізми;

- як правило, у складі БД існують засоби для підтримки її цілісності та захисту від неавторизованого доступу

**3. Класифікація запитів.**

* А(O)=? - по заданому атрибуту і об’єкту знайти відповідне значення. Приклад: Який виторг у кіоска № 2 у березні. Такий запит називають прямим.
* А(?) = V - по заданному атрибуту та значенню знайти множину об’єктів. Які кіоски у березні мали виторг у 2000 грн. Такий запит називають інвертованим.
* ?(О)=V - знайти імена атрибутів, що мають вказане значення по специфікованому об’єкту. У якому місяці виторг кіоска № 2 дорівнював 2000 грн.
* ?(О)= ? - знайти всі значення по специфікованому об’єкту разом з відповідними іменами атрибутів. Для кіоска № 2 по кожному місяцю знайти значення виторгу.
* А(?)= ? - знайти всі значення по вказаному атрибуту разом з специфікаціями відповідних об'єктів. Знайти значення виторгу за лютий місяць для всіх кіосків.
* ?(?) = V - знайти специфікації об'єктів з іменами потрібних атрибутів, де мається вказане значення. Знайти номери кіосків та назви місяців, коли виторг дорівнював 2000 грн.

1. **Класифікація АІС за структурами даних.**

Структури даних, що підтримуються в системі бази даних, - це важливий фактор, що впливає, як на виразову потужність, так і на ефективність функціонування. Для систем з ієрархічною структурою базовою структурою даних є дерево; як правило, вони мають найвищу ефективність функціонування, але виразові можливості їх відносно низькі. Системи з структурами даних типу мережа мають значно кращі виразові можливості, але дещо програють у ефективності функціонування, точніше, від користувача вимагається значно вищий рівень кваліфікації для ефективної експлуатації таких систем. В останні десятиріччя найбільшого розповсюдження (особливо для персональних ЕОМ) зазнали СУБД реляційного типу, для яких характерно щонайпростіша структура даних (плоский файл), але одночасово суттєво підвищений рівень мов маніпулювання даними, що максимально употужнює виразові можливості та знижує ефективність функціонування, тому для таких систем потрібні потужні комп’ютери, і вони значно чутливіші (порівняно з попередниками) до росту об’ємів даних.

**5. Реляційна алгебра Кодда, основні операції.**

Алгебраїчна пара : основна множина і сигнатура. Операнди і результати операцій належать основній множині. Реляційна алгебра – алгебра в строгому розумінні. Елементи основної множини – реляції. Сигнатура складається з 8-ми операцій.

Теоретико-множинні операції ∩,U,\ – частково визначені (визначені тільки для сумісних реляцій з однаковою структурою).

Реляції R 1 (A 1 , … , A n ) і R 2 (B 1 , … , B k ) називаються сумісними, якщо:

1. у них однакова кількість атрибутів, тобто k = n;

2. кожному атрибуту першої реляції можна поставити у взаємно однозначну відповідність атрибут другої реляції, тобто існує таке бієктивне відображення

, що

Операція *декартового добутку* теж відноситься до теоретико-множинних операцій, але на відміну від попередніх є всюди визначеною.

Операція вилучення дублів пов’язана з сортуванням, тому є досить складною в сенсі виконання.

Операція θ - з’єднання (θ - join). Введемо поняття Θ - порівнюваності для кортежів. На двох реляціях R(A1, … , Ak) і S(B1, … , Bn) візьмемо два кортежі D1⊗D2 ⊗ … ⊗ Dm: r1 = (d1, d2, … , dm), r2 = (d1’, d2’, … , dm’) Θ = {=, ≠, <, ≤, >, ≥, …} Два кортежі r1 і r2 Θ - порівнювані: r1 Θ r2 ⟺ di Θ di | i = 1,m (m ≤ k, m ≤ n)

Операція еквіз’єднання називається природним з’єднанням, якщо в результат проходить лише один стовпчик з двох.

Приклад використання операції ділення. Знайти прізвища програмістів, які знають мови А і Ф одночасно. R[мова ≑ мв]S

**6. Реляційне числення. Мова ALPHA. Теза Кодда про повноту.**

З теоретичної точки зору РЧ є аналогом численням предикатів 1-го порядку і, таким чином, створює чітке та математично строге підґрунтя для вище розташованої надбудови. З прагматичної точки зору РЧ є основою для мов запитів дуже високого рівня.

Спираючись на РЧ як аналог числення предикатів 1-го порядку Кодд запропонував поняття реляційної повноти, тобто мова запитів є реляційно повною, якщо вона дає можливість описати певну множину формул (стосовно даних, які зберігаються в базі), інтерпретація яких дає можливість одержати будь-які потрібні дані з бази у вигляді відповідної реляції. Він це виразив як тезу про повноту, пізніше більшість фахівців стали називати її тезою Кодда: реляційне числення є реляційно повним.

Альфа-мова - розширений діалект мови програмування Algol 60. Мова була доповнена типом «комплексний», над яким можна було виконувати всі арифметичні операції. У мові Альфа з'явилася можливість роботи з масивами цілком, для чого було введено поняття багатовимірного значення і багатовимірної змінної і ряд інших розширень, покликаних наблизити Алгол до природної математичної нотації. Приклад:

Знайти прізвища постачальників таких, що для них існують поставки з кодом КП такі, що для них існують деталі червоного кольору.

RANGE Д X

RANGE ОПД Y

GET W(П.прізвище): ∃ X ∃ Y (Y.КП = П.КП & X.КД = Y.КД & X.колір = червоний)

**7. Поняття квазіключа, первинного ключа та «чужого» ключа.**

Квазіключем (деякої реляції R) називається список атрибутів, який задовольняє умовам :

Первинний ключ — атрибут або набір атрибутів, який однозначно ідентифікує кортеж даного відношення. Первинний ключ обов'язково унікальний. У реляційних базах даних первинний ключ обирають серед одного з потенційних ключів або ж генерують сурогатний ключ.

*Чужий ключ відношення* – це мінімальний набір атрибутів, що відповідає можливому ключу іншого, але зв'язаного з ним відношення.

**8. Функціонально повна залежність. 2НФ. 2НФп.**

функціонально повно залежить (ФПЗ) від , якщо

1. R. → R.

2. ∀А⊂ (власної підмножини ) ∃В⊆: ¬(R.А→R.В)

Кажуть, що реляція знаходиться в 2-ій нормальній формі (2НФ), якщо вона знаходиться в першій нормальній формі і кожний вторинний атрибут функціонально повно залежить від кожного квазіключа.

Достатні умови знаходження реляції в 2НФ.

1) всі атрибути первинні;

2) кожен квазіключ має один атрибут.

Кажуть, що реляція знаходиться в 2-ій нормальній формі посиленій(2НФп)(або у 2-ій нормальній формі Бойса-Кодда (Boyce-Codd)), якщо вона знаходиться в першій нормальній формі і кожен атрибут функціонально повно залежить від кожного квазіключа. Іншими словами знімається вимога вторинності атрибута.

**9. Теорема Хіза. Декомпозиція реляцій.**

Якщо R.M 1 → R.M 2 , тоді R можна декомпонувати на дві реляції з можливістю відновлення без втрат. Природне з’єднання відбувається по атрибутам М 1 .

*Доведення:* Беремо реляцію R[ R \(M 2 \M 1 )] (вона має ту ж саму кількість кортежів, що і R) і доповнюємо її елементами з М 2 , які однозначно визначаються на основі функціональної залежності від М 1 , в результаті отримуємо повну реляцію R.

Процедура отримання реляцій у 2НФ наступна: потрібно дослідити залежності заданої реляції і визначити, чи є порушення 2НФ. Якщо порушень немає, то процедура завершена і ми маємо 2НФ. Якщо ж порушення є, то потрібно взяти залежність, яка створює це порушення, і застосувати теорему Хіза. В результаті отримаємо дві реляції, стосовно яких знову застосуємо попередню процедуру. Оскільки кількість як залежностей, так і атрибутів у початковій реляції скінченна, то рано чи пізно вийдемо на 2НФ. Цей процес називається *декомпозицією* реляції до 2НФ, або нормалізацією (2НФ).

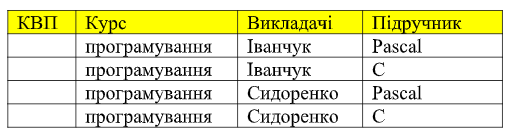
**10. Транзитивна залежність в сенсі реляційного підходу. 3НФ. 3НФп.**

транзитивно залежить від якщо

Кажуть, що реляція знаходиться в 3НФ, якщо вона в 2 НФ і не має транзитивної залежності вторинних атрибутів від кожного квазіключа.

Кажуть, що реляція знаходиться в 3НФп, якщо вона в 2 НФп і не має має транзитивної залежності кожного атрибута від кожного квазіключа.

**11. Багатозначні залежності. 4НФ.**



Розглянемо таблицю КВП. Курс програмування ведуть два викладачі і для курсу використовується два підручники. Легко бачити, що функціональних залежностей між атрибутами цієї таблиці немає; разом з тим якісь залежності є – певний курс можуть вести не всі викладачі, а тільки деяка їх підмножина, подібні ж стосунки між курсом та підручниками, а також викладачами та підручниками. Залежності такого виду будемо називати багатозначними (multivalued).

Багатозначна залежність називається тривіальною, якщо вона дублюється функціональною, інакше вона є нетривіальною.Реляція знаходиться в 4НФ, якщо вона знаходиться в 3НФ і не має нетривіальної багатозначної залежності, або

4НФ: (А – квазіключ)

**12. Мінімальна структура функціональних залежностей.**

Кажуть, що множина функціональних залежностей F є мінімальною, якщо виконуються 3 властивості:

* Кожна права частина має один атрибут
* Ні для якої функціональної залежності не буде виконуватись

(X → А) є F , F\ {X → A} !~ F антинадлишковість залежностей, тобто з структури F не можна вилучити жодної залежності, не втративши еквівалентності з F.

* Антинадлишковість лівої частини. Ні для якої функціональної залежності не буде виконуватись (X → А) є F & Z С Х, F \ {X → A} U {Z → A} !~ F тобто з лівої частини залежності не можна вилучити жодного атрибута, не втративши еквівалентності з F.

**13. Поняття про транзакцію. ACID.**

Транзакція - це логічна одиниця роботи, що передбачає узгодженість окремих операцій СУБД, це перехід з одного узгодженого стану б\д в інший узгоджений стан, причому в проміжних точках стани можуть бути неузгоджені.

Транзакції мають 4 властивості: атомарність, узгодженість, ізоляція, довговічність – АУІД.

* Атомарність. Транзакції атомарні (все або нічого) (Atomicity).
* Узгодженість. (Concurrency) Транзакції захищають базу даних узгоджено, тобто переводять б/д з одного узгодженого стану в інший без обов’язкової підтримки узгодженості у всіх проміжних точках.
* Ізоляція. (Isolation) Транзакції відокремлені одна від іншої. Якщо буде стартовано кілька конкуруючих транзакцій, то будь-яке оновлення виконане однією транзакцією буде приховане від інших, принаймні до завершення даної транзакції.
* Довговічність. (Durability) Якщо транзакція виконана, то її оновлення зберігаються, навіть якщо у наступний момент буде мати місце збій системи.

**14. Бази знань. Моделі представлення знань.**

Три основні частини системи бази знань:

* база знань (БЗ);
* механізм прийняття рішень (МР);
* інтерфейс (І).

4 ознаки знань:

* внутрішня інтерпретованість; (ця ознака є в б/д, але тут додаються ще й програми)
* структурованість; (теж є в б/д)
* наявність зв’язків; (не тільки в структурному плані, а і відображення відношень типу: причина – наслідок; факт – подія – процес)
* активність; (виявлення суперечностей та неповноти знань)

Класифікація знань.

* поняття (математичні і нематематичні);
* факти;
* правила, залежності, закони,зв’язки;
* алгоритми, процедури.

4 види моделей представлення знань.

* Семантичні мережі.
* Фрейми.
* Логічні моделі знань.
* Системи продукцій.

Семантичні мережі часто представляються графом, де вершини – це поняття (об’єкти, події, процеси), а дуги – це відношення на множині понять. Відношення можуть бути дуже різноманітними ( що підтримує ознаку – наявність зв’язку).

Фрейм(frame) – це структура виду , де – ім’я фрейма, а - slot; - – ім’я слота, - його значення.

Логічна модель знань

M = <T, P, A, F>, де T – множина базових елементів; P – множина синтаксичних правил; А – аксіоми(синтаксично правильні формули); F – правила виводу. (теореми).

Продукція – трійка: (ім’я продукції, умова, оператор);

Є термінальні стани, досягнувши яких система зупиняється.

**15. Основні властивості експертних систем.**

ЕС – використовує знання з ефективним представленням, процедури логічного виводу для проведення експертизи, пояснює шляхи логічного виводу, працює з нечіткими знаннями.

Три головні якості характерні для ЕС:

* Корисність,
* Продуктивність,
* Прозорість (можливістьотримати пояснення)

Класифікація ЕС по інтерфейсу.

* Системи типу запит-відповідь здіалоговим спілкуванням на професійній мові користувача;
* Системи-консультанти – аналіз, узагальнення та зберігання досвіду провідних фахівців;
* Системи-розв’язувачі – велика кількість розрахунків.

Умови для предметної області:

* Наявність достатньої множини експертів, які готові передати свої знання ЕС;
* Експерти можуть вербалізувати свої знання;
* Схожість оцінок різних експертів;
* Складність і практична значимість задачі;
* Не потрібна велика кількість знань ПрО на рівні здорового глузду;
* Необхідність евристичних методів.

Класифікація ЕС за призначенням:

* Ідентифікуючі (хім. структури, корисні копалини, спілкування іноземною мовою) – данні часто неточні чи неповні, потрібна оцінка, припущення;
* САПР (розрахунок та проектування конструкцій, автомат. програм) – ПРИЗ, УТОПИСТ;
* Планові та прогнозні – (погода, транспортні потоки, оцінка урожаю, воєнні дії) – неповна інформація, велика кількість сценаріїв;
* Діагностичні системи (мед. діагностика, мех. діагностика, енергетичні мережі, помилки прог. забезпечення) – здатність знаходити причини нестандартної поведінки, порівняння з еталонною поведінкою
* Системи спостереження, контролю та керування (сигнал про втручання оператора)
* Навчальні системи

**16. Аксіоми Армстронга. Система Р.**

Аксіоми Армстронга — множина аксіом (або, точніше, правил висновування), що використовуються для висновування всіх функціональних залежностей у реляційній базі даних

Є принаймні дві причини для чого потрібні аксіоми Армстронга:

1. множина функціональних залежностей реляційної БД є структурою, точніше верхньою

напіврешіткою;

1. існують мінімальні структури функціональних залежностей, які є суттєво зручнішими

при декомпозиції реляцій до 3НФ чи 3НФп.

Система аксіом Армстронга (система Р):

Якщо якась властивість (залежність) задана на [], то вона задана і на []

**17. Залежності по з’єднанню без втрат. 5НФ.**

Для реляції R(А,В,С) А ~» В ⬄ можлива декомпозиція на 2 реляції R(А,В) і R(А,С) без втрат даних. Такі залежності будемо називати залежностями по з’єднанню без втрат.

Залежність по з’єднанню без втрат називається тривіальною, якщо вона дублюється багатозначною, інакше вона є нетривіальною. Реляція знаходиться в 5НФ, якщо вона знаходиться в 4НФ і не має нетривіальних залежностей по з’єднанню без втрат.