

ЛЕКЦІЯ. СЕРЕДОВИЩЕ БАЗ ДАНИХ

План:

1. Архітектура баз даних.
2. Моделі даних.
3. Програмні і мовні засоби баз даних.
4. Архітектура інформаційної системи.

Література

1. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. К.: КНУБА, 2005. – 204 с.
2. Гайна Г.А. Організація баз даних і знань. Мови баз даних: Конспект лекцій.–К.:КНУБА, 2002. – 64 с.
3. Гайна Г.А., Попович Н.Л. Організація баз даних і знань. Організація реляційних баз даних: Конспект лекцій.–К.:КНУБА, 2000. – 76 с.

1. Архітектура баз даних

Для організації роботи з БД необхідно забезпечити незалежність прикладних програм від даних. Це обумовлено тим, що при зміні системи, а також з метою забезпечення ефективного обслуговування користувачів необхідно виконувати роботи щодо зміни методів зберігання даних в БД, шляхів доступу до даних, змінювати структури і формати даних та зв'язки між ними. Якщо не застосовувати спеціальні підходи і при написанні застосувань вводити програмний опис методів доступу, засобів зберігання даних, формати даних, то при будь-якій зміні в БД для перелічених випадків буде необхідно корегувати текст програми користувача, що потребує значних витрат.

Незалежність застосувань від даних забезпечується засобами СУБД. Цей підхід базується на тому, що користувачі застосовуючи БД, не знають внутрішнє представлення даних.

На рис. 1 показана трирівнева модель архітектури СУБД, що була запропонована Комітетом планування стандартів і норм SPARC (Standarts Planning and Requirements Committee) Американського національного інституту стандартів ANSI (American National Standarts Institute).

Опис структури даних на будь-якому рівні називається схемою. Існує три різних типи схем БД, які визначаються згідно з рівнями абстракції архітектури

СУБД. На самому верхньому рівні є декілька зовнішніх схем, які відповідають різним представленням даних. Цей рівень визначає точку зору на БД окремих застосувань. Кожне застосування бачить і обробляє тільки ті дані, які необхідні цьому застосуванню.

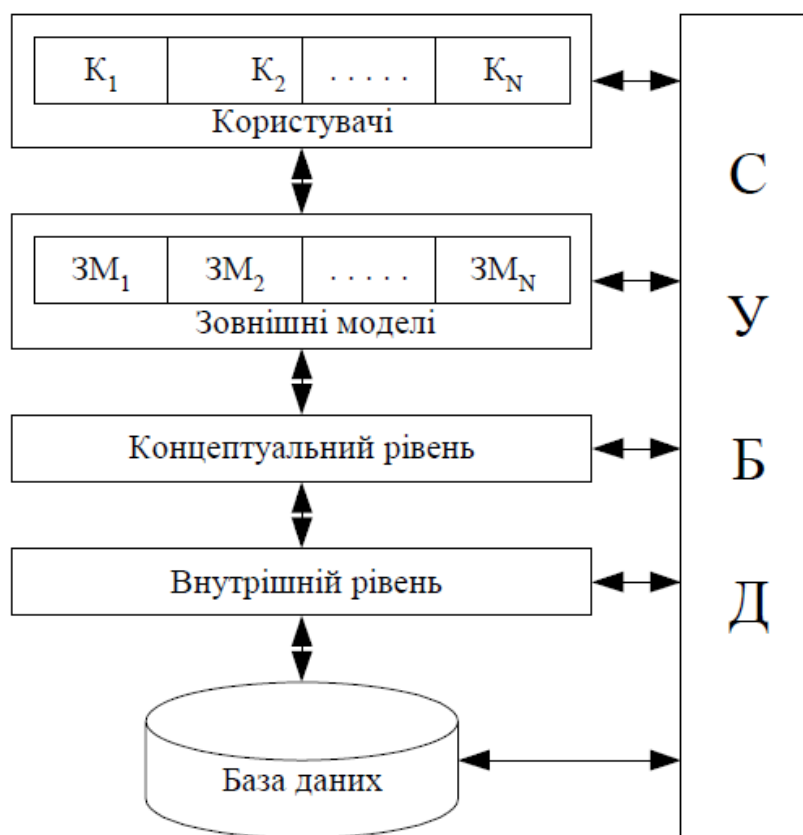


Рис. 1. Трирівнева архітектура СУБД

На концептуальному рівні опис БД називається концептуальною схемою. Тут БД представлена в найбільш загальному вигляді, який об'єднує дані, що використовуються всіма застосуваннями, які працюють з БД. Фактично концептуальний рівень відображає модель предметної області, для якої створювалася БД.

На внутрішньому рівні опис БД називається внутрішньою схемою. Тут БД представлена у вигляді безпосередньо даних, що розташовані в файлах, які відповідають фізичній організації БД.

Трирівнева архітектура СУБД дозволяє забезпечити незалежність від даних. Це означає, що зміни на нижніх рівнях не впливають на верхні рівні. Розрізняють логічну і фізичну незалежність при роботі з даними. Логічна незалежність від даних означає захищеність зовнішніх схем від змін, що вносяться в концептуальну схему. Зміни концептуальної схеми БД не

викликають необхідності в корегуванні існуючих зовнішніх схем для користувачів, і відповідно не викликають змін в застосуваннях, що працюють з цими схемами. Фізична незалежність від даних означає захищеність концептуальної і зовнішніх схем від змін, що вносяться у внутрішню схему. До змін внутрішньої схеми належать використання різних файлових систем або структур даних, різних пристроїв зберігання, модифікація пошукових структур тощо.

Крім трьох названих рівнів абстрагування в БД існує ще один рівень, що передує їм. Цей рівень відображає інформацію про предметну область, а модель цього рівня називається інфологічною моделлю предметної області. Таким чином головними рівнями абстрагування в БД є рівні:

- інфологічний;
- зовнішній;
- концептуальний;
- внутрішній.

Перехід від одного рівня абстрагування до наступного і складає в загальному вигляді процес проектування БД.

2. Моделі даних

Модель даних – це деяка абстракція, в якій знаходять своє відображення найбільш важливі аспекти функціонування визначеної предметної області, а другорядні – ігноруються. Модель даних являє собою деяку цільову модель предметної області. У моделі даних розрізняють три головні складові:

- структурна частина, яка визначає правила породження допустимих для даної СУБД видів структур даних;
- управляюча частина, яка визначає можливі операції над такими структурами ;
- класи обмежень цілісності даних, які можуть бути реалізовані засобами цієї системи.

Моделювання даних – це процес створення логічного представлення структури бази даних.

На рис. 2 показана класифікація моделей даних.

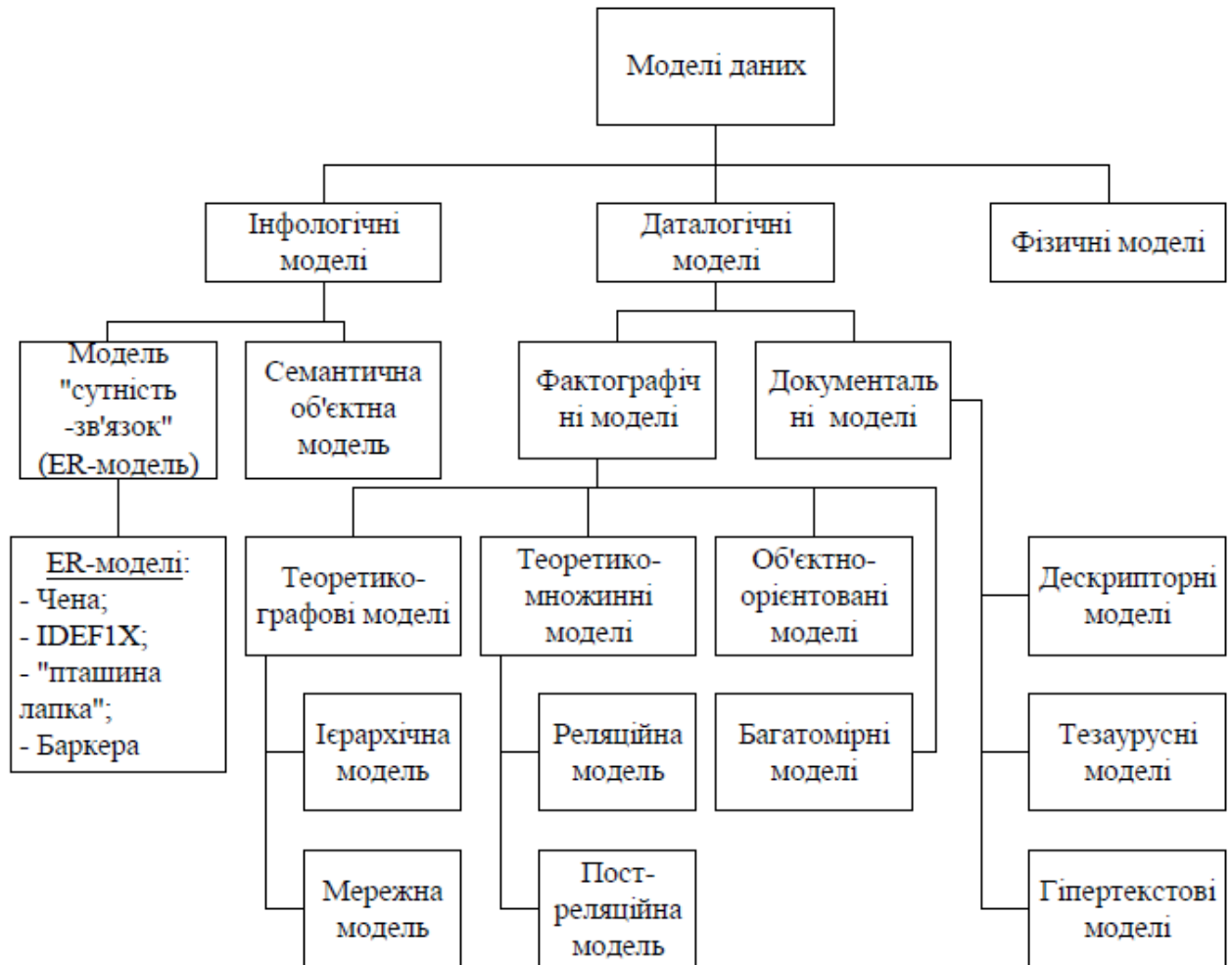


Рис. 2. Класифікація моделей даних

Кожному рівню представлення інформації відповідає певна модель.

Інфологічна модель – відображає інформацію про предметну область у вигляді незалежного від СУБД, що використовується. Ця модель відображає інформаційно-логічний рівень абстрагування, який пов'язаний з описом об'єктів предметної області, їх властивостей і взаємозв'язків.

Часто ці моделі ототожнюють з концептуальними моделями предметної області і називають концептуальними інфологічними моделями (внутрішня і зовнішня концептуальні інфологічні моделі).

Даталогічна модель – модель логічного рівня, яка відображає логічні зв'язки між елементами даних безвідносно до їх змісту і середовища збереження. Часто ці моделі ототожнюють з логічними моделями.

Фізична модель – описує те, як дані зберігаються в комп'ютері, представляючи інформацію про структуру записів, їх впорядкованість і про існуючі шляхи доступу до даних.

Модель "*сутність-зв'язок*" (ER-модель) – описує модель предметної області і складається з множини сутностей, множини зв'язків між сутностями, а також з атрибутів сутностей і зв'язків. В модель входить обмеження цілісності даних, що пов'язано з двома множинами сутностей і називається залежністю по існуванню. ER-моделі дозволяють графічно представляти моделі предметних областей. Вони є складовою частиною багатьох CASE-продуктів.

Семантична об'єктна модель – описує модель предметної області і являє собою модель даних. Ця модель складається з семантичних об'єктів, що містять сукупність атрибутів. Атрибути групуються у класи. Модель даних володіє більш розвиненими засобами відображення семантики у порівнянні з теоретико-множинними і теоретико-графовими моделями.

Теоретико-графова модель – модель даних, в якій дозволені структури даних можуть бути представлені у вигляді графа загального або спеціального виду, наприклад дерева. Необхідну групу операцій на мові маніпулювання даними, що засновані на цій моделі, представляють навігаційні операції. Операції над даними мають позаописовий характер.

Теоретико-множинна модель – модель даних, в якій використовується математичний апарат реляційної алгебри, реляційного обчислення, а операції над даними маніпулюють таблицями.

Фактографічні моделі – містять відомості, які представлені у вигляді спеціальним чином організованих сукупностей формалізованих записів даних.

Документальні моделі – передбачають, що в якості одиничного елемента інформації виступає неподільний на менші складові частини документ, а інформація про документ, як правило, не структурується, або структурується в обмеженому вигляді. В цих моделях в основному розглядаються тексти на природній мові, формати документів є вільними.

Ієрархічна модель – модель даних в основі якої використовується ієрархічна, деревоподібна структура даних. Вершинами цієї структури є записи,

які складаються з простих елементів даних різних типів. Батьківському запису відповідає довільне число екземплярів підлеглих записів кожного типу.

Мережна модель – модель даних, в якій дозволені структури даних можуть бути представлені у вигляді графа загального вигляду. Вершинами такого графа можуть бути дані різних типів – від атомарних елементів даних до записів складної структури. На відміну від ієрархічної моделі наступник в цій моделі може мати довільне число батьків.

Реляційна модель – модель даних, яка заснована на математичному понятті відношення і представленні відношень у формі таблиць.

Постреляційна модель – розширена реляційна модель, яка знімає обмеження неподільності даних, що зберігаються в записах таблиць. Ця модель допускає багатозначні поля – поля, значення яких складається з підзначень. Набір значень багатозначних полів вважається самостійною таблицею, яка вбудована в основну таблицю. Часто ці моделі ототожнюють з об'єктно-реляційними моделями.

Об'єктно-орієнтована модель – модель даних, яка базується на понятті об'єкта, тобто сутності, що володіє станом і поведінкою. Стан об'єкта визначається його атрибутами, а поведінка визначається сукупністю операцій, що визначені для цього об'єкта. Також передбачається можливість підтримки зв'язків між типами об'єктів.

Багатомірна модель – модель даних, яка оперує багатомірним представленням даних (у вигляді гіперкубу) і орієнтована на підтримку аналізу даних. Передбачається конструювання різноманітних агрегацій даних у межах гіперкубу, побудова різних його проєкцій – підмножин гіперкубу, деталізація і обертання даних, а також цілий ряд інших операцій.

Дескрипторна модель – описує кожен документ за допомогою дескриптора. Дескриптор має жорстку структуру і являє собою набори деяких лексичних одиниць (слов, словосполучень, термінів), які потрібні для роботи з документами. Дескриптори між собою не зв'язані.

Тезаурусна модель – описує кожен документ за допомогою дескрипторів, а також змістовних відношень між лексичними одиницями (ціле-частина, род-

вид, клас-підклас і т.ін.). Ці моделі дозволяють підвищити ефективність дескрипторних моделей за рахунок більш ефективного відображення предметної області.

Гіпертекстова модель – модель, що заснована на розмітці документа за допомогою спеціальних навігаційних конструкцій, які відповідають змістовим зв'язкам між різними документами, або окремими фрагментами одного документа. Такі конструкції утворюють деяку семантичну мережу в базі документів.

2. Програмні даних

Оснoву програмних засобів банку даних складає СУБД. В СУБД можна виділити ядро СУБД, яке підтримує сукупність базових механізмів роботи з БД, а також інші компоненти, які забезпечують засоби тестування, налагодження системи, утіліти, які забезпечують виконання таких додаткових функцій, як відновлення БД, збір статистики і т.ін. Важливою компонентою СУБД є транслятори і компілятори для мов, що використовуються. Для роботи з БД розробляються застосування.

Застосування – програма, яка призначена для рішення деякої сукупності задач в даній предметній області, або яка являє собою типовий інструментарій, що застосовується в різних предметних областях. Застосування може використовувати різні джерела даних (фактографічні, документальні, WEB і т.ін.), мати різну архітектуру (дволанкову, триланкову, розподілену).

Застосування бази даних – застосування, яке використовує ресурси деякої системи баз даних. Для доступу до БД використовується інтерфейс прикладного програмування СУБД, в середовищі якої він підтримується. Застосування можуть бути написані на стандартній алгоритмічній мові програмування (Pascal, C, Basic тощо) з вбудованими операторами на мові SQL.

Мова даних – мова, яка призначена для визначення даних, маніпулювання даними, а також інших функцій в термінах понять і рамках можливостей, які передбачені в моделі даних, що підтримується розглядуваною СУБД.

Мова запитів – мова доступу до БД, яка орієнтована на користувача. Мова запитів належить до декларативних мов, описує властивості і взаємозв'язки сутностей, але не описує алгоритм рішення задачі. Як правило мова запитів використовується в інтерактивному режимі, а також може вбудовуватися в програмний код застосувань.

Мова маніпулювання даними (Data Manipulation Language – DML) – мова, яка реалізує операційні можливості моделі даних, що використовується. Ця мова визначає операції, які допустимі над даними, що знаходяться в БД.

Мова визначення даних (Data Definition Language – DDL) – мова, яка служить для опису структури БД, обмежень цілісності, а також, можливо, для специфікації процедур, що зберігаються, тригерів, обмежень управління доступом і т.ін. Функціональні можливості мов визначення і маніпулювання можуть інтегруватися в єдину мову даних.

Мова програмування баз даних – мова, яка забезпечує концептуально єдине інтегроване середовище, яке засновано на єдиній моделі даних, для програмування застосувань і управління даними в БД. Такі мови об'єднують функції традиційних мов програмування із засобами опису і маніпулювання даними в БД.

Мова програмування базова – традиційна мова програмування, для якої дана СУБД забезпечує інтерфейс прикладного програмування (API). Прикладна програма, яка написана на цій мові, має доступ до деяких функціональних можливостей СУБД і може виконувати з її допомогою доступ до БД.

Мови, які належать до мов четвертого покоління (Fourth-Generation Language – 4GL), мають такі функціональні можливості:

- генератори екранних форм для створення шаблонів вводу і відображення даних;
- генератори звітів на основі інформації, що зберігається в БД;
- генератори застосувань для створення програм обробки даних;
- генератори запитів;
- генератори для представлення даних у вигляді різного роду діаграм.

Для формування запиту за допомогою різних СУБД найчастіше використовуються дві основні мови опису запитів:

- *SQL (Structured Query Language)* – структурована мова запитів;
- *QBE (Query By Example)* – мова запитів за зразком.

Головна різниця між цими мовами полягає в тому, що мова QBE передбачає ручне або візуальне формування запиту, а мова SQL – програмування запиту.

Мова SQL є найбільш поширеною мовою для роботи з БД.

На даний час існують такі міжнародні стандарти на мову SQL: SQL1, SQL2, SQL3.

Мова SQL не володіє функціями повноцінної мови розробки і орієнтована на доступ до БД. Використання мови SQL може бути самостійним і вона може включатися в склад засобів розробки програм. В цьому випадку її називають вбудованим SQL. Розрізняють два головних методи використання вбудованого SQL: статичний і динамічний.

Статичне використання передбачає застосування в програмі функцій викликів мови SQL, які включаються в програмний модуль і виконуються після компіляції програми.

Динамічне використання передбачає динамічну побудову викликів функцій мови SQL та інтерпретацію цих викликів у ході виконання програми. Динамічний метод застосовується тоді, коли вид SQL запиту заздалегідь невідомий і будується у діалозі з користувачем.

Будь-яке SQL-застосування реляційної БД складається з трьох частин: інтерфейса користувача, набору таблиць в БД і SQL-машини.

4. Архітектура інформаційної системи

Ефективність функціонування інформаційної системи багато в чому залежить від її архітектури. Функціональні частини інформаційної системи можуть розміщуватися на одному або на декількох комп'ютерах. У разі, якщо інформаційна система розміщується на одному комп'ютері, можливі такі варіанти використання програмних засобів:

застосування і СУБД, застосування і ядро СУБД, незалежне застосування.

У першому випадку взаємодія користувача і СУБД виконується або напряму через користувацький інтерфейс СУБД, або за допомогою застосування (рис. 3).



Рис. 3. Використання застосування і СУБД

У другому випадку взаємодія користувача і СУБД виконується за допомогою застосування (рис. 4). Такий підхід дозволяє підвищити швидкість роботи застосування, зменшити об'єм необхідної пам'яті.



Рис. 4. Використання застосування і ядра СУБД

Створення незалежних застосувань дозволяє звертатися до БД без СУБД (рис. 5). Такий підхід дозволяє ще більше підвищити швидкість роботи застосування, зменшити об'єм необхідної пам'яті. Недоліки такого підходу пов'язані з трудомісткістю доробки застосувань, відсутністю стандартних засобів СУБД по обслуговуванню БД.

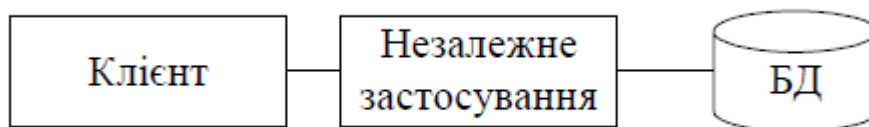


Рис. 5. Використання незалежного застосування

При інтеграції комп'ютерів в мережі виникає можливість розподілу застосувань, що працюють з єдиною БД, а також самої БД по мережі. Найбільш поширеною є схема, при якій кожен користувач має свою персональну БД (КБД) і звертається до серверної БД (СБД) за інформацією, що спільно використовується багатьма користувачами (рис. 6).

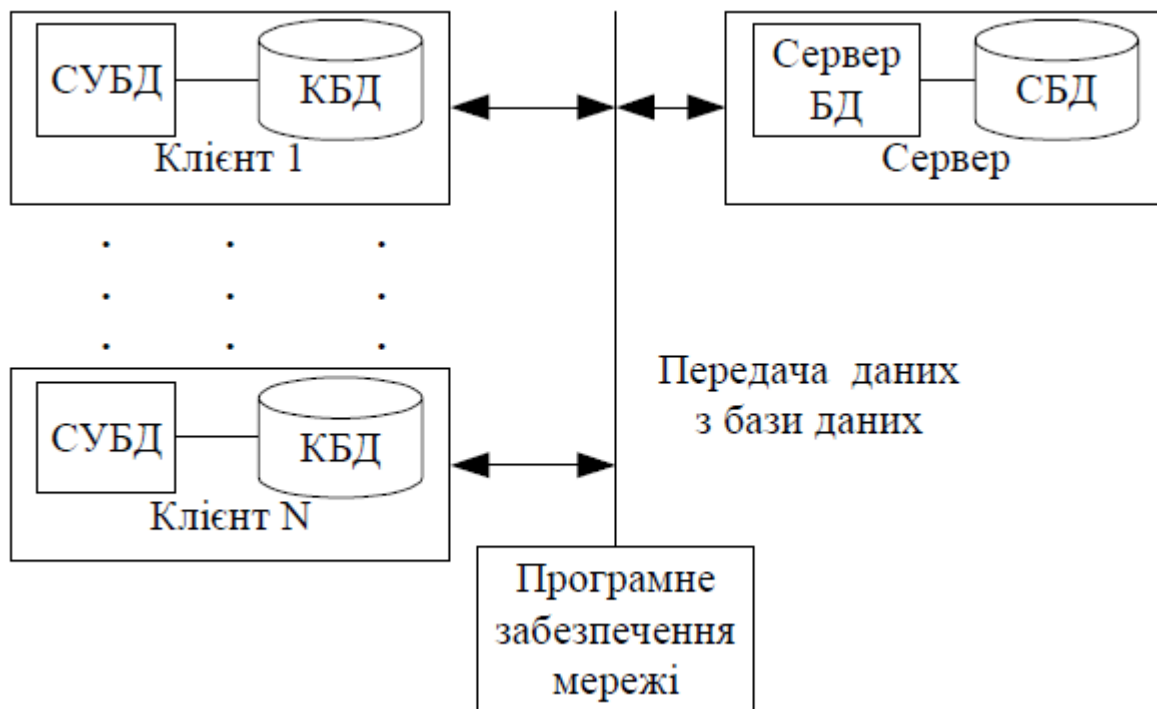


Рис. 6. Використання сервера БД

Під сервером розуміється комп'ютер або програма, які керують певними ресурсами. Клієнт – це теж комп'ютер або програма, які використовують цей ресурс.

Такий підхід дозволяє поєднувати переваги централізованого зберігання з індивідуальною роботою користувачів.

Контрольні запитання

1. Представити та пояснити архітектуру бази даних.
2. Які головні рівні абстрагування в базах даних?
3. Назвати основні моделі даних і дати їх характеристику.
4. В чому різниця між реляційною і об'єктно-реляційною моделями?
5. Які переваги і недоліки об'єктно-орієнтованої моделі?
6. Дати класифікацію програмних і мовних засобів по роботі з БД.
7. Вказати основні архітектурні рішення інформаційної системи з базами даних.
8. Що таке схема БД? Перелічити її компоненти.
9. Що розуміється під незалежністю даних?

Розробив: к.т.н., доц. Полотай О.І.