**Лекція 5 Життєвий цикл інформаційної системи Концептуальне проектування баз даних. Логічне проектування баз даних**

**Файл-серверні та клієнт-серверні СКБД**

**Файловий сервер.** БД розташовується на файловому сервері (або декількох файлових серверах), у якості якого може використовуватися найбільш потужний комп’ютер із об’єднаних у мережу. Функції файлового сервера полягають у зберіганні баз даних і забезпеченні доступу до них користувачів, що працюють на різних (клієнтських) комп’ютерах. Файли бази даних відповідно до користувальницьких запитів передаються на робочі станції, де, в основному, й відбувається обробка. Передані дані обробляються СКБД, що також перебувають на комп’ютерах користувачів. Після того як користувачі виконають необхідні операції з даними, вони копіюють файли назад на файловий сервер, де інші користувачі, у свою чергу, зможуть з ними працювати. Крім того, кожен користувач може створювати на локальному комп’ютері свої власні бази даних, використовувані ним монопольно.

Ця схема підходить для невеликих обсягів даних. При збільшенні числа комп’ютерів у мережі або росту обсягів БД, продуктивність інформаційної системи різко падає. Це пов’язано зі збільшенням обсягу даних, переданих мережею, оскільки вся обробка відбувається на комп’ютері користувача. Недоліком подібного підходу є висока ймовірність втрати змін, виконаних користувачами, під час збереження змінених файлів на центральному сервері. Справа у тому, що користувачі можуть і не підозрювати, що окрім них ще хтось змінював дані, і тільки через деякий час помітять, що дані перебувають не в тому стані, що очікувалося. Прикладами СКБД, призначених безпосередньо для розробки локальних користувальницьких додатків БД, тобто додатків, що працюють на одному локальному комп’ютері, або в комп’ютерній мережі є Microsoft Visual FoxPro, Microsoft Access, Paradox for Windows, dBase for Windows та ін.

**Клієнт-сервер.** Технологія клієнт-сервер передбачає, що окрім зберігання бази даних центральний комп’ютер (сервер бази даних) має ще й забезпечувати виконання основного обсягу обробки даних. При технології клієнт-сервер запит клієнта (робочої станції) на виконання якої-небудь операції з даними (наприклад, звичайної вибірки) провокує на сервері пошук і добування даних. Витягнуті дані (але не файли) транспортуються мережею від сервера до клієнта.

Системи, що використовують технологію клієнт-сервер, складаються з двох частин:

 клієнтська частина (Front-End) забезпечує інтерфейс користувача (у тому числі графічний) і перебуває на його комп’ютері;

 серверна частина (Back-End) забезпечує керування даними, поділ інформації, адміністрування й безпеку. Серверна частина перебуває на спеціально виділених для цього комп’ютерах-серверах. Прикладами СКБД, що працюють за технологією клієнт-сервер, є Microsoft SQL Server, Oracle, MySQL, Firebird та ін.

**Життєвий цикл бази даних**

Життєвий цикл БД нерозривно пов'язаний із життєвим циклом інформаційної системи. Він складається з таких етапів (рис. 1):

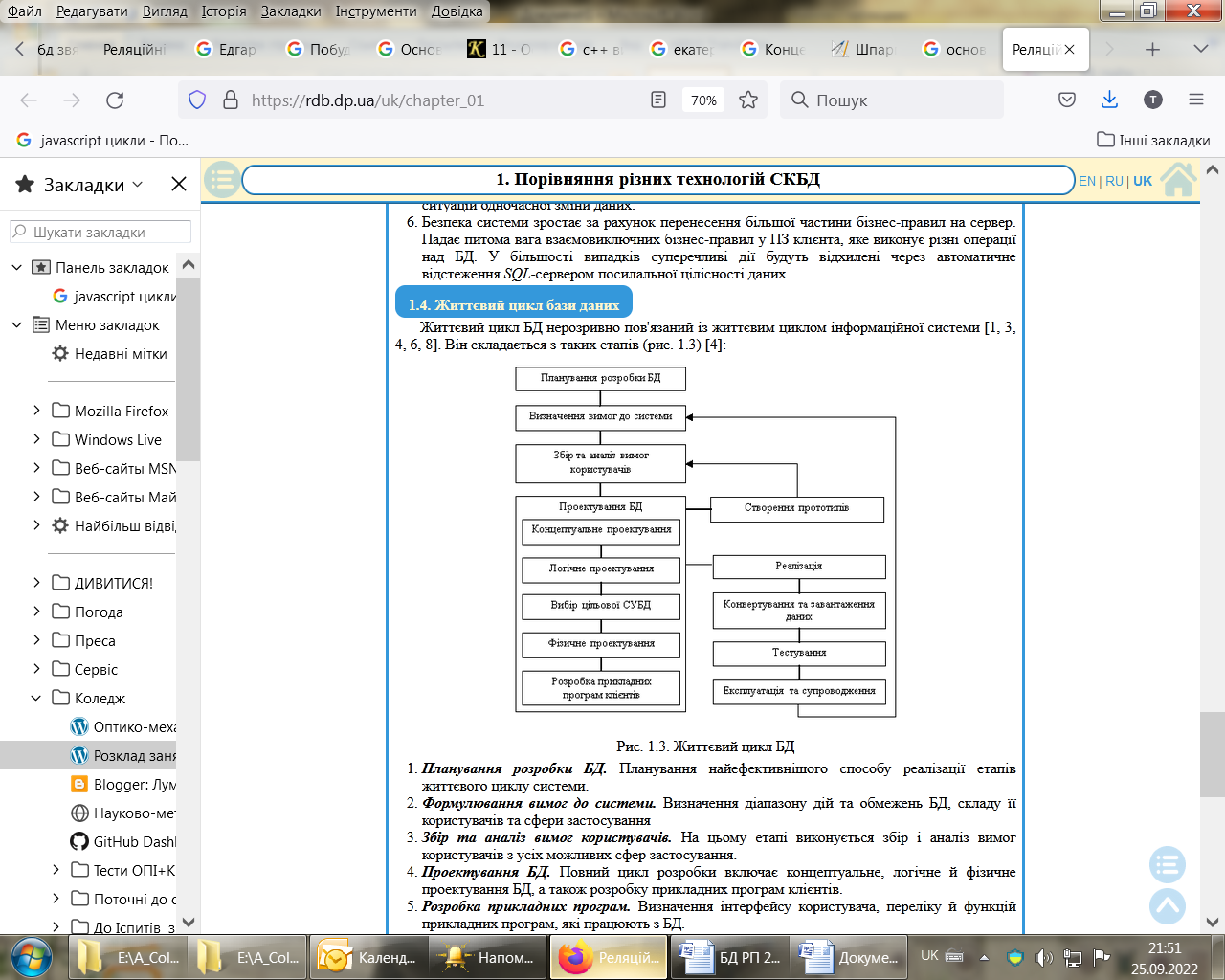


Рис. 1. Життєвий цикл БД

1. ***Планування розробки БД.*** Планування найефективнішого способу реалізації етапів життєвого циклу системи.
2. ***Формулювання вимог до системи.*** Визначення діапазону дій та обмежень БД, складу її користувачів та сфери застосування
3. ***Збір та аналіз вимог користувачів.*** На цьому етапі виконується збір і аналіз вимог користувачів з усіх можливих сфер застосування.
4. ***Проектування БД.*** Повний цикл розробки включає концептуальне, логічне й фізичне проектування БД, а також розробку прикладних програм клієнтів.
5. ***Розробка прикладних програм.*** Визначення інтерфейсу користувача, переліку й функцій прикладних програм, які працюють з БД.
6. ***Створення прототипів (необов'язково).*** Тут створюється робоча модель БД, яка дозволяє розробникам або користувачам уявити й оцінити остаточний вигляд і способи функціонування системи.
7. ***Реалізація.*** На цьому етапі в програмному коді реалізується концептуальна зовнішня й внутрішня функціональність БД і прикладного ПЗ, які визначені на попередніх етапах.
8. ***Конвертування й завантаження даних.*** Процес, який полягає в перетворенні та завантаженні даних (і прикладних програм) зі старої системи в нову.
9. ***Тестування.*** БД тестується з метою виявлення помилок, а також її перевірки на відповідність усім вимогам, які висунуті користувачами.
10. ***Експлуатація та супровід.*** На цьому етапі БД вважається повністю розробленою і реалізованою. Надалі уся система буде перебувати під постійним спостереженням і відповідним чином підтримуватися. За потреби у прикладні програми, що функціонують, можуть вноситися зміни, які відповідають новим вимогам. Вони реалізуються за допомогою повторного виконання деяких етапів життєвого циклу, які наведені раніше.

Для малих БД з невеликою кількістю користувачів життєвий цикл може виявитися не дуже складним. При проектуванні великих БД, з десятками й навіть тисячами користувачів, сотнями запитів та прикладних програм, воно може стати занадто складним.

Слід відзначити, що розглянуті етапи не є надто послідовними, а містять деяку кількість повторів попередніх кроків у вигляді циклів зворотного зв'язку. Наприклад, при проектуванні БД можуть виникнути проблеми, для вирішення яких потрібно буде повернутися до етапу збору й аналізу вимог. Цикли зворотного зв'язку можуть існувати майже між усіма етапами.

П**роектування бази даних**  починається по завершенні процедури аналізу усіх вимог до проекту, висунутих з боку підприємства-замовника. Розглянемо методологію, пропоновану для процедури проектування бази даних, що входить у загальний життєвий цикл прикладної програми, що використовує бази даних реляційного типу. Пропонована методологія являє собою поетапний посібник, що охоплює всі три основні фази процесу проектування баз даних, а саме: концептуальне, логічне і фізичне проектування.

-       **Концептуальне проектування**– створення концептуального представлення бази даних, що включає визначення типів найважливіших сутностей і існуючих між ними зв'язків.

-       **Логічне проектування**– перетворення концептуального представлення в логічну структуру бази даних, включаючи проектування відношень.

-       **Фізичне проектування**– ухвалення рішення про те, як логічна модель буде фізично реалізована у базі даних, створюваної за допомогою обраної СКБД.

**Введення в методологію проектування баз даних**

**Методологія проектування** – структурований підхід, що передбачає використання спеціалізованих процедур, технічних прийомів, інструментів, документації і націлений на підтримку і спрощення процесу проектування.

Методологія проектування передбачає розбивку всього процесу на кілька фаз, кожна з яких, у свою чергу, складається з декількох етапів. На кожнім етапі розробнику пропонується набір технічних прийомів, що дозволяють вирішувати задачі, що стоять перед ним на даній стадії розробки. Крім того, методологія пропонує методи планування, координації, керування, оцінки ходу розробки проекту, а також структурований підхід до аналізу і моделювання всього набору пропонованих до бази даних вимог і дозволяє виконати ці дії стандартизованим і організованим чином.

**Концептуальне, логічне і фізичне проектування бази даних**

 У методології проектування баз даних весь процес розробки розділяється на три основні фази: концептуальне, логічне і фізичне проектування.

**Концептуальне проектування** **бази даних** – процедура конструювання інформаційної моделі, наприклад, підприємства, що не залежить від яких-небудь фізичних умов реалізації. Фаза концептуального проектування бази даних починається зі створення концептуальної моделі даного підприємства, цілком незалежної від будь-яких деталей реалізації. До останнього відносяться: обраний тип СКБД, склад програм прикладної програми, використовувана мова програмування, конкретна обчислювальна платформа і будь-які інші фізичні особливості реалізації.

**Логічне проектування** **бази даних** – Процес конструювання інформаційної моделі підприємства на основі існуючих конкретних моделей даних, не залежних від використовуваної СКБД і інших фізичних умов реалізації.

Фаза логічного проектування бази даних полягає в перетворенні концептуальної моделі даних у логічну модель даного підприємства з урахуванням обраного типу СКБД (наприклад, передбачається використання деякої реляційної СКБД). Логічна модель даних є джерелом інформації для фази фізичного проектування. Вона надає розробнику фізичної моделі засоби проведення всебічного аналізу різних аспектів роботи з даними, що має винятково важливе значення для вибору дійсно ефективного проектного рішення.

**Фізичне проектування бази даних** – Процес створення опису конкретної реалізації бази даних, розташовуваної у вторинній пам'яті. Передбачає опис структури збереження даних і методів доступу, призначених для здійснення найбільш ефективного доступу до інформації.

Фаза фізичного проектування бази даних передбачає прийняття розроблювачем остаточного рішення про способи реалізації створюваної бази. Тому фізичне проектування обов'язкове виробляється з урахуванням всіх особливостей використовуваної СКБД. Між фазами фізичного і логічного проектування завжди мається визначений зворотний зв'язок, оскільки рішення, прийняті на етапі фізичного проектування з метою підвищення продуктивності розроблювальної системи, можуть зажадати деякого перегляду логічної моделі даних.

У цілому, процедура розробки включає наступні етапи.

**Концептуальне проектування бази даних**

Етап 1. Створення локальної концептуальної моделі даних виходячи з представлень про предметну область кожного з типів користувачів.

Етап 1.1.   Визначення типів сутностей.

Етап 1.2.   Визначення типів зв'язків.

Етап 1.3.   Визначення атрибутів і зв'язування їх з типами сутностей і зв'язків.

Етап 1.4.   Визначення доменів атрибутів.

Етап 1.5.   Визначення атрибутів, що є потенційними і первинними ключами.

Етап 1.6.   Спеціалізація або генералізація типів сутностей (необов'язковий етап).

Етап 1.7.   Створення діаграми "сутність-зв'язок".

Етап 1.8.   Обговорення локальних концептуальних моделей даних з кінцевими користувачами.

**Логічне проектування бази даних (для реляційної моделі)**

Етап 2. Побудова і перевірка локальної логічної моделі даних на основі уявлення про предметну область кожного з типів користувачів.

Етап 2.1.   Перетворення локальної концептуальної моделі даних у локальну логічну модель.

Етап 2.2.   Визначення набору відношень виходячи зі структури локальної логічної моделі даних.

Етап 2.3.   Перевірка моделі за допомогою правил нормалізації.

Етап 2.4.   Перевірка моделі у відношенні транзакцій користувачів.

Етап 2.5.   Створення діаграм "сутність-відношення".

Етап 2.6.   Визначення вимог підтримки цілісності даних.

Етап 2.7.   Обговорення розроблених локальних логічних моделей даних з кінцевими користувачами.

Етап 3. Створення і перевірка глобальної логічної моделі даних.

Етап 3.1.   Злиття локальних логічних моделей даних у єдину глобальну модель даних.

Етап 3.2.   Перевірка глобальної логічної моделі даних.

Етап 3.3.   Перевірка можливостей розширення моделі в майбутньому.

Етап 3.4.   Створення остаточного варіанта діаграми "сутність-відношення".

Етап 3.5.   Обговорення глобальної логічної моделі даних з користувачами.

**Фізичне проектування бази даних (з використанням реляційної СКБД)**

Етап 4. Перенос глобальної логічної моделі даних у середовище цільової СКБД.

Етап 4.1.   Проектування основних таблиць у середовищі цільової СКБД.

Етап 4.2.   Реалізація бізнес-правил підприємства в середовищі цільової СКБД.

Етап 5. Проектування фізичного представлення бази даних.

Етап 5.1.   Аналіз транзакцій.

Етап 5.2.   Вибір файлової структури.

Етап 5.3.   Визначення вторинних індексів.

Етап 5.4.   Аналіз необхідності введення контрольованої надмірності даних.

Етап 5.5.   Визначення вимог до дискової пам'яті.

Етап 6. Розробка механізмів захисту.

Етап 6.1.   Розробка представлень користувача (переглядів).

Етап 6.2.   Визначення прав доступу.

Етап 7. Організація моніторингу і настроювання функціонування системи.

Концептуальне і логічне проектування бази даних включає три основних етапи. Задачею першого етапу є розбивка проекту на групу щодо невеликих (і більш простих) задач, виходячи з представлень про предметну область прикладної програми, властивих кожному з типів кінцевих користувачів. Результатом виконання цього етапу є створення локальних концептуальних моделей даних, що представляють собою повне і точне відображення представлень про предметну область прикладної програми окремих типів користувачів.

На другому етапі локальні концептуальні моделі даних перетворяться в локальні логічні моделі даних (для реляційної моделі даних). На цьому етапі з моделей даних видаляються небажані елементи, що утрудняють реалізацію моделей даних у середовищі реляційних СКБД. Потім коректність логічних моделей даних перевіряється за допомогою правил нормалізації. Нормалізація являє собою ефективний засіб, що дозволяє переконатися в структурній узгодженості, логічній цілісності і мінімальній надлишковості прийнятої моделі даних. Додатково модель даних перевіряється з метою виявлення можливості здійснення транзакцій, що будуть виконуватися користувачами створюваної прикладної програми. Усі ці перевірки дозволяють одержати необхідну впевненість у тому, що прийнята модель даних є цілком коректною. Локальні логічні моделі даних при необхідності можна використовувати для генерації прототипів реалізацій бази даних, призначених для окремих типів користувачів прикладної програми.

На третьому етапі виконується інтеграція локальних логічних моделей даних (котрі відображають уявлення про предметну область окремих типів користувачів) у єдину глобальну логічну модель даних усього підприємства (узагальнюючі уявлення про предметну область усіх типів користувачів).

У методології проектування істотна роль приділяється кінцевим користувачам, що постійно залучаються розроблювачами для ознайомлення і перевірки створюваних моделей даних і супровідної документації. Проектування баз даних звичайно являє собою ітеративний процес, що має конкретну точку початку і включає необмежене число циклів покращень і доробок. Хоча цей процес може здатися чітко визначеною послідовністю процедур, варто особливо підкреслити, що це ні в якій мірі не означає, що розробка бази даних неодмінно повинна виконуватися саме в такій манері. Дуже імовірно, що відомості, отримані при виконанні деякого етапу, можуть зажадати змінити рішення, прийняті на одному з попередніх етапів. Тому практика попереднього аналізу можливих результатів виконання наступних етапів може виявитися корисною при виконанні початкових етапів розробки. Таку методологію варто розглядати як загальну схему, що дозволить підвищити загальну ефективність роботи з проектування баз даних.

## Життєвий цикл та методологія проектування

Процес створення такої структури бази даних, яка б відповідала вимогам користувачів, називається проектуванням бази даних. Його можна порівняти зі зведенням нової будівлі: визначення вимог, проектування, конструювання і, нарешті, реалізація.

Життєвий цикл системи баз даних є концепцією, в межах якої корисно й зручно розглядати розвиток такої системи. Він, як і життєвий цикл будь-якої програмної системи, складається з двох основних фаз: проектування та реалізації (рис.  1)

Фаза проектування поділяється на такі етапи:

* визначення стратегії;
* аналіз предметної області;
* концептуальне моделювання;
* логічне й фізичне проектування.

Фаза реалізації складається з таких пунктів:

* власне програмна реалізація;
* документування;
* дослідне впровадження.



Рисунок 1 – Етапи життєвого циклу БД

Методологія проектування баз даних — це сукупність принципів, методів, інструментів і засобів, що застосовуються для послідовного розроблення структури бази даних. Оскільки система баз даних складається з програм і даних, методологія проектування баз даних розглядається як невід’ємна частина загальної методології проектування програмних систем.

До методології проектування баз даних висуваються певні вимоги. Прийнятною вважається база даних, яка відповідає вимогам користувачів (ефективність, адаптивність, незалежність, захищеність, цілісність тощо) і вимогам до апаратного забезпечення. Методологія має бути достатньо гнучкою, доступною розробникам із різним досвідом проектування, що використовують різні моделі даних і різне програмне забезпечення СКБД.

Методологія проектування баз даних визначає:

* процес проектування;
* методику виконання розрахунків і критеріїв оцінювання альтернативних рішень на кожному етапі проектування;
* інформаційні вимоги як вихідні дані для процесу проектування;
* засоби опису вихідних даних і відображення результатів кожного етапу проектування.

**Процес проектування.**

Для баз даних можна застосувати ітераційне низхідне проектування. Процес проектування добре структурований, оскільки кожний його етап завершується певним результатом, а також тому, що допускається ітераційне повторення попередніх етапів, якщо отриманий результат не відповідає вимогам замовника або системним вимогам. Це дає можливість переглядати й змінювати проектні рішення на будь - якому етапі.

З проектуванням тісно пов’язане експертне оцінювання проекту. Мета експертизи - знайти помилки й виправити їх на ранніх етапах проектування. Зазвичай експертиза виконується після завершення кожного з етапів.

Етап проектування БД вважається одним із самих складних етапів створення БД, який не має явно вираженого початку й закінчення. У порівнянні з аналізом вимог до БД або розробкою додатків, проектування БД, на думку багатьох провідних фахівців, є невдало структурованим завданням. Якщо всі етапи створення БД перекриваються один з одним у своїй послідовності, то етап проектування перекривається з усіма іншими етапами. Проектування починається з моменту прийняття стратегічних рішень і триває на етапах реалізації й тестування.

Процес проектування БД охоплює кілька основних сфер:

* проектування об'єктів БД (таблиці, подання, індекси, тригери, збережені процедури, функції, пакети) для подання даних ПЗ в БД;
* проектування інтерфейсу взаємодії з БД (форми, звіти й т.д.), тобто проектування додатків, які будуть супроводжувати дані в БД і реалізовувати питально-відповідні відношення на цих даних;
* проектування БД під конкретне обчислювальне середовище або інформаційну технологію (архітектура "клієнт-сервер", паралельні архітектури, розподілене обчислювальне середовище);
* проектування БД під призначення системи (інтелектуальний аналіз даних, OLAP, OLTP і т.д.).

**Критерії оцінювання**

Оцінювання необхідне для ухвалення рішень за наявності альтернатив. Труднощі у визначенні критеріїв і виборі альтернатив пов’язані з тим, що часто розробляється кілька проектів структури бази даних і потрібно оцінити, який з них є кращим. Зробити це буває досить складно.

Критерії є кількісні (час обробки запитів, вартість операцій маніпулювання даними, витрати пам’яті тощо) та якісні (гнучкість, адаптивність, сприйнятливість та сумісність).

**Інформаційні вимоги**

Визначаючи вимоги до інформації, врахуйте, що є інформація, яка стосується структури даних (опис даних та зв’язків безвідносно до конкретних способів їхнього використання й обробки), та інформація про спосіб використання даних (опис вимог до обробки даних).

**Засоби опису**

Це мовні засоби, призначені для опису результатів виконання кожного етану проектування. А саме, йдеться про такі засоби.

* Природна мова, якою строго означуються всі необхідні для опису результатів проектування поняття. Використовується, як правило, на етапі визначення стратегії.
* Стандартні форми, анкети та бланки. Використовуються переважно на етапі аналізу.
* Спеціальні формалізовані мови концептуального моделювання (семантичні мережі, числення предикатів та ЕR-мови). Використовуються переважно на етапі концептуального моделювання.
* Формалізовані мова означення даних (МОД) і мова маніпулювання даними (ММД). Використовуються на етапі логічного проектування. Зазвичай з цією метою застосовують мову SQL.

## Етапи проектування БД - ****визначення стратегії****

Метою етапу визначення стратегії є формування спільно з замовником прикладних моделей, вироблення переліку рекомендацій і ухвалення узгодженого плану, складеного з урахуванням наявних організаційних, фінансових і технічних обмежень, що відображує як поточні, так і майбутні потреби організації.

**Опис.** Детальний аналіз структури організації може бути початковою базою для розроблення перспективного плану створення системи, але витрати на його проведення навряд чи будуть економічно виправданими. Як правило, стратегія розроблення інформаційної системи визначається в результаті узагальненого аналізу, на підставі якого потім будується великомасштабна модель прикладної області. Стратегія має визначатися в достатньо стислі терміни з тим, щоб результати проектування не втрачали актуальності.

Результати цього етапу мають узгоджуватися одне з одним і бути достатньо чітко сформульовані, щоб замовник міг легко співвіднести запропоновану стратегію зі своїми завданнями і зрозуміти, які саме чинники обумовили ухвалення тих чи інших рішень. Окрім того, йому має бути викладена перспектива подальшого аналізу, уточнення й перегляду стратегічних рішень.

**Результати.** Основними результатами цього етапу мають бути

* опис напрямів прикладної діяльності, зокрема формулювання її цілей і завдань, визначення пріоритетів, обмежень, критичних чинників успіху та ключових показників ефективності;
* опис цілей і завдань автоматизації, витрат і можливого виграшу;
* узагальнена діаграма сутностей і зв’язків;
* узагальнена ієрархічна схема завдань (виробничих та управлінських);
* рекомендації щодо майбутньої реалізації та подолання можливих труднощів;
* визначення меж і окреслення сфери застосування системи баз даних;
* можлива архітектура системи;
* поетапний план проектування бази даних

Такий підхід до моделювання предметної області передбачає її відображення з трьох різних точок зору:

* загальний напрям прикладної діяльності;
* прикладні завдання;
* інформаційні потреби.

Побудовані на цьому етапі моделі мають бути зрозумілими для замовника, а для того щоб досягти повного узгодження різних точок зору на прикладну область і можливі напрями діяльності, проводяться групові координаційні наради Стратегії еволюціонують і розвиваються, обставини й завдання з часом можуть змінюватися, відтак неможливо запропонувати директивний метод моделювання стратегій. Тому важливо поєднувати неупередженість до нових рішень зі здатністю швидко оцінювати альтернативні напрямки діяльності з урахуванням заданих обмежень і пріоритетів.

**Ключові чинники успіху:**

* використання всіх можливих засобів, що дають змогу підвищити рівень знань про предметну область;
* активна участь у розробленні стратегії осіб, які добре розуміють справжні потреби організації;
* проведення плідних нарад із ретельним розглядом усіх питань

### ****Аналіз предметної області****

Підсумки етапу визначення стратегії є вихідними даними для етапу аналізу, де вони ретельно перевіряються, уточнюються і деталізуються, для того щоб забезпечити предметній області адекватність моделі, гарантувати можливість реалізації рішень і сформувати тверде підґрунтя для етапів концептуального моделювання, логічного й фізичного проектування. Цей етап є найменше вивченим, найважчим і найтривалішим. Проте він найважливіший, оскільки саме на ньому формується більшість проектних рішень

**Опис.** Аналіз предметної області складається з аналізу даних та аналізу завдань. Аналіз даних передбачає документування всіх атрибутів. Аналіз завдань може потребувати застосування різноманітних методів побудови діаграм для дослідження зв'язків і способів використання даних, подій, станів даних, а також детального опису алгоритмів.

Вивчається потреба в заходах із контролю та захисту даних, їхньому резервному копіюванні та відновленні. Має бути проведений детальний аналіз наявних систем та інших чинників, що впливають на процес впровадження системи. Потрібно виявити всі обмеження і припущення, що можуть вплинути на подальше проектування, використання ресурсів і терміни проведення робіт.

**Підхід.** На цьому етапі аналітики й користувачі працюють пліч-о-пліч, встановлюючи й перевіряючи вимоги. Аналіз предметної області передбачає:

* проведення бесід з користувачами;
* перегляд усіх документів та бланків, які обробляються і формуються організацією;
* аналіз потоків документів;
* аналіз способів вирішення завдань організації;
* фіксація правил, обмежень та законів, що діють у предметній області.

Результати:

* узгоджена діаграма сутностей і зв’язків;
* відомості про обсяги даних, частоту виконання завдань, очікуваний користувачем рівень продуктивності;
* деталізовані й узгоджені описи завдань:
* первинний варіант стратегії впровадження;
* опис заходів з ревізії і контролю даних, резервного копіювання й відновлення;
* загальний опис процедур, що не автоматизуються;
* критерії прийнятності, якості, гнучкості та продуктивності;
* попереднє оцінювання обсягів системи;
* узгоджений підхід до здійснення етапу проектування й фази реалізації;
* уточнений план розроблення системи.

**Ключові чинники успіху:**

* активна участь користувачів;
* ретельна перевірка достовірності, повноти й несуперечності даних;
* виявлення всіх питань та припущень, що мають ключове значення для проектування і впровадження;
* встановлення точних характеристик ключових завдань і даних; жорсткий контроль за ходом робіт, концентрація зусиль на виконанні календарних планів і дотриманні запланованих термінів

### ****Концептуальне моделювання предметної області****

Етап концептуального моделювання полягає в побудові опису предметної області в термінах формальної мови, наприклад у термінах моделі сутностей і зв’язків. Ідеї побудови концептуальної моделі предметної області беруть свій початок із публікації робочої групи ANSI/SPARC, присвяченій архітектурі СКБД.

**Опис.** На базі змістовного опису предметної області, отриманого в результаті її аналізу, розроблюється строгий формальний опис її інформаційного забезпечення.

Результати:

* формальний опис інформаційного забезпечення предметної області:
* докладний і строгий опис сховищ даних;
* детальний опис потоків даних;
* детальний опис ієрархії й специфікація завдань, що вирішуються; детальний опис чинних у предметній області правил і обмежень.

Ключові чинники успіху :

* глибоке знання і практичний досвід використання мов опису концептуальної моделі,
* знання методів проектування реляційної моделі та/або інших моделей даних.

### ****Логічне та фізичне моделювання даних****

Етап проектування полягає в пошуку і визначенні якнайкращого способу реалізації та виконання вимог, сформульованих на етапі аналізу. При цьому має забезпечуватись належний рівень сервісу в умовах певного технологічного середовища, що відповідає ухваленим рішенням щодо рівня автоматизації.

Логічне проектування — це розроблення структур зберігання, методів доступу й логічної структури системи баз даних без прив’язки до конкретної СКБД.

Фізичне проектування — це проектування бази даних у конкретній СКБД.

**Опис.** Під час виконання цього етапу модель сутностей і зв’язків перетворюється на схему бази даних і специфікації зберігання даних на зовнішніх носіях. Прикладні задачі перетворюються на модулі й процедури з необхідними засобами ревізії/контролю та резервного копіювання й відновлення. Проектуються формати звітів, визначаються міжмодульні зв’язки. Виходячи із завдань, сформульованих на попередніх етапах, створюються проектні рішення з архітектури комунікаційної мережі. Для полегшення процесу пошуку проектних рішень можуть застосовуватися прототипи. Нарешті, на етапі проектування розроблюються програмні специфікації і план тестування системи, а отримана інформація й нові погляди на майбутню систему застосовуються для доопрацювання та уточнення стратегії її реалізації.

**Підхід**. Аналітики, розробники і проектувальники баз даних спілкуються з користувачами менше, ніж на етапі аналізу, проте вони повинні надати їм для перевірки результати своєї роботи або запропонувати на вибір різні варіанти проектних рішень. Корисним є створення прототипів, проте воно має розглядатися лише як засіб, а не самоціль.

Результати:

* архітектура системи:
* схеми системних модулів:
* логічна і фізична схеми;
* схема бази даних і файлів;
* детальні часові та ємнісні характеристики;
* програмні специфікації;
* специфікації неавтоматизованих процедур;
* чорновий варіант посібника для користувача;
* узгоджена стратегія впровадження, що складається з планів приймання і здачі системи, організаційної підготовки, заходів зі збирання даних, переходу на нову систему та встановлення обладнання;
* план випробувань системи;
* чорновий варіант експлуатаційної документації;
* уточнений план розроблення системи.

**Ключові чинники успіху.** В результаті виконання даного етапу має бути створений проект, що забезпечує задоволення прикладних вимог з урахуванням наявних технічних обмежень. Ключовими чинниками успіху в досягненні цієї мети є:

* знання можливостей апаратного й програмного забезпечення;
* розуміння прикладних потреб;
* ухвалення обґрунтованих компромісних рішень;
* виявлення і вирішення потенційних проблем.

## Бізнес-модель процесу проектування бази даних

Процес проектування БД може бути представлений у вигляді моделі бізнес-процесів. Бізнес-модель процесу проектування дозволяє:

* відобразити суб'єктивну думку розробника БД на процес проектування конкретної БД;
* врахувати особливості ІТ-проекту, у рамках якого проектується БД;
* досить швидко скласти план проектування конкретної БД;
* прорахувати тривалість проектних робіт (створити тимчасову модель проектування).

### Типова бізнес-модель процесу проектування бази даних

Значна частина проектів у сфері інформаційних технологій спрямована на розроблення й створення ІС, у рамках яких здійснюється обробка даних різної складності. Практично у всіх таких проектах вирішується завдання проектування БД певного типу.

В експлуатації БД повинна задовольняти набір вимог щодо ряду інтегрованих параметрів, таких, як:

* функціональність й адаптованість;
* продуктивність обробки трансакцій;
* пропускна здатність;
* час реакції;
* безпека.

Такі параметри іноді перебувають у протиріччі один з одним. Так, високі вимоги до функціональності на даному конкретному устаткуванні можуть вступати у конфлікт з високими вимогами до продуктивності. Наприклад, звіти можуть генеруватися протягом декількох годин і знизити в цей час реакції користувачів, що працюють із системою у діалоговому режимі.

Етап проектування БД вважається одним із найскладніших етапів створення БД, який не має явно вираженого початку й закінчення. У порівнянні з аналізом вимог до БД або розробленням додатків, проектування БД, на думку багатьох провідних фахівців, є невдало структурованим завданням. Якщо всі етапи створення БД перекриваються один з одним у своїй послідовності, то етап проектування перекривається з усіма іншими етапами.

Розглянемо типову бізнес-модель процесу проектування БД. На рис..2 наведена контекстна діаграма процесу проектування БД.

Як видно з рисунка.2, на вхід процесу проектування БД подаються:

* інформаційна модель ПЗ БД: діаграми "сутність-зв'язок" (ER-діаграми);
* функціональна модель ПЗ БД: бізнес-модель процесів, діаграми потоку даних (DF-діаграми), діаграми станів, - діаграми життєвих циклів сутностей, специфікації на системи (вимоги), бізнес-правила;
* загальносистемні вимоги й обмеження;
* завдання зворотного впливу.

На виході процесу проектування БД формуються такі результати:

* фізична модель БД, що може бути перетворена у скрипт для створення БД;
* фізична БД;
* специфікація модулів додатків БД;
* план тестування БД.

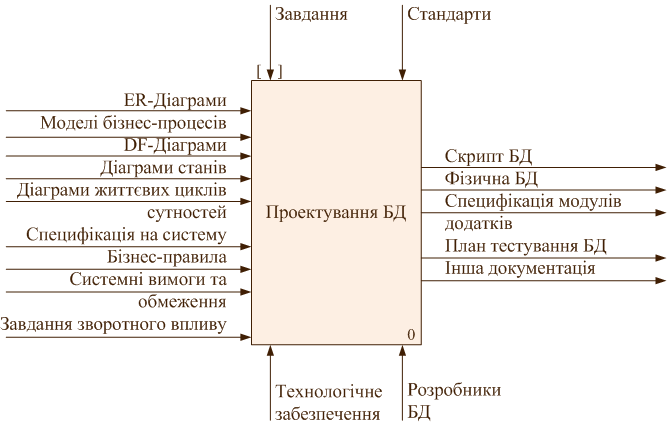


Рисунок 2 – Контекстна діаграма процесу проектування БД

### Діаграма декомпозиції першого рівня

Починаючи функціональну декомпозицію процесу проектування БД, приходимо до діаграми декомпозиції процесу проектування БД першого рівня, яка відбиває основні, найбільш великі професійні завдання (етапи) проектування БД (рис. 3).

Такими завданнями (етапами) є:

* збір і аналіз вхідних даних – це початковий етап проектування, на якому здійснюються збір і контроль якості результатів аналізу ПЗ БД, готується план проектування БД;
* створення логічної моделі БД – це етап, на якому на підставі інформаційної моделі ПЗ БД створюється логічна структура БД, незалежна від її реалізації;
* створення фізичної моделі БД: внутрішня схема – це етап, на якому на підставі логічної моделі БД створюється фізична структура БД, залежна від її реалізації. На цьому етапі виконується перетворення відношення логічної моделі реляційної БД у команди створення об'єктів фізичної БД, у результаті чого створюється так звана внутрішня схема БД. Додатково може бути створена так звана зовнішня схема БД, останнє відбиває точку зору користувачів на дані в БД;
* створення фізичної моделі БД: урахування впливу транзакцій – це етап, на якому аналізуються можливі транзакції системи, за потреби виконується денормалізація відношення для забезпечення більш високої продуктивності БД;
* створення серверного коду – це етап, на якому на підставі функціональної моделі ПЗ БД створюється серверний код БД у вигляді тригерів, збережених процедур і пакетів. Ці модулі створюються розробником БД і виконуються сервером;
* проектування модулів додатків БД – це етап, на якому створюються специфікації модулів додатків, розробляються стратегії тестування БД і додатків, створюється план тестування додатків БД і готуються тестові дані;
* контроль якості проектування БД полягає в перевірці якості результатів проектування на кожному його етапі;
* урахування завдань зворотного впливу полягає у настроюванні деяких трансакцій до БД і локальному перепроектуванні БД відповідно до вимог, що надходять з інших етапів створення БД.

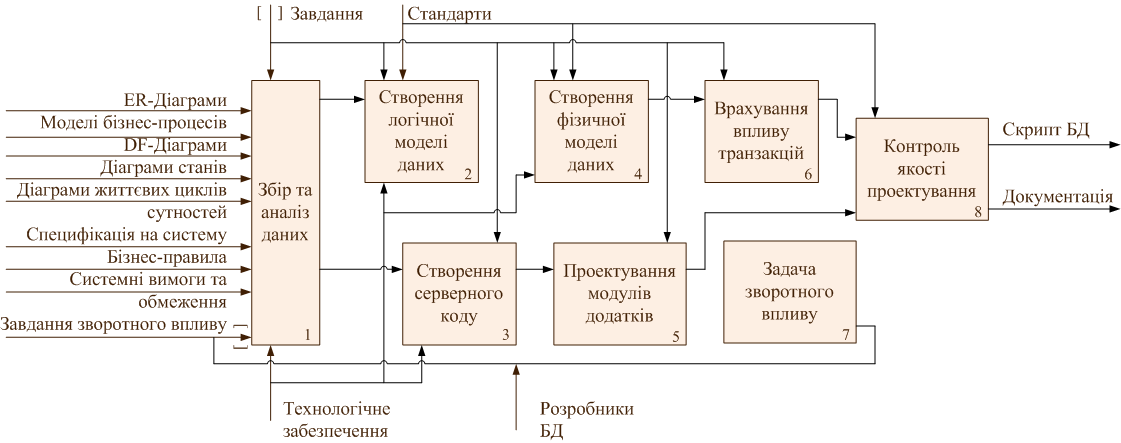


Рисунок 3 – Діаграма декомпозиції процесу проектування БД: перший рівень

**КОНТРОЛЬНІ ПИТАННЯ**

1. Що таке інформаційна система організації та з яких компонентів вона складається?
2. Як ручні картотеки пристосовані для зберігання та обробки даних?
3. Дайте визначення файловій системі з точки зору зберігання даних та керування ними.
4. До чого призводять відомі Вам обмеження файлових систем як сховищ даних?
5. Що таке база даних? Сформулюйте визначення.
6. Які переваги та недоліки має СКБД?
7. Порівняйте відомі Вам архітектури СКБД.
8. Які недоліки СКБД архітектури "файл-сервер" Вам відомі?
9. Які переваги СКБД архітектури "клієнт-сервер" Ви знаєте?
10. Дайте характеристику моделям архітектури "клієнт-сервер".
11. Яка існує функціональна різниця між серверною та клієнтською частинами СКБД архітектури "клієнт-сервер"?
12. Що таке бізнес-правила? Де вони реалізуються?
13. Що Ви знаєте про етапи життєвого циклу БД?