

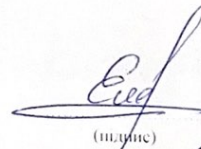
**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
Факультет транспорту і будівництва
Кафедра логістичного управління та безпеки руху на транспорті**

**ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломної кваліфікаційної роботи
освітньо-кваліфікаційного рівня магістр**

галузі знань 27 – «Транспорт»
спеціальності 273 Залізничний транспорт/ Інтероперабельність і безпека на залізничному транспорті

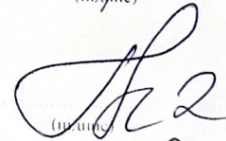
на тему: «Удосконалення методів організації бази даних в автоматизованій системі управління безпекою руху поїздів»

Виконав: здобувач вищої освіти
групи ІБЗТ-223м
Гавриленко О.В.



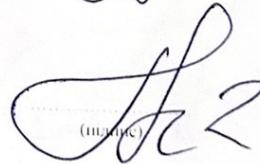
(підпис)

Керівник: проф. Чернецька-Білецька Н.Б.



(підпис)

Завідувач кафедри: проф. Чернецька-Білецька Н.Б.



(підпис)

ЗМІСТ

	ВСТУП.....	3
1.	АНАЛІЗ АВТОМАТИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ БЕЗПЕКОЮ РУХУ ПОЇЗДІВ.....	7
1.1.	Аналіз функцій системи безпеки руху поїздів.....	7
1.2.	Інформаційні функції управління безпекою руху та нагляду автоматизованої АСУ БД.....	8
1.3.	Керуючі функції управління безпекою руху.....	11
2.	АНАЛІЗ ПРОБЛЕМ ТЕХНОЛОГІЇ ПОБУДОВИ БАЗ ДАНИХ.....	12
2.1.	Аналіз проблем проектування баз даних	12
2.2.	Аналіз методик проектування баз даних на концептуальному рівні...	15
2.3.	Аналіз моделей даних концептуального рівня.....	24
3.	МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ БАЗ ДАНИХ.....	35
3.1.	Методика концептуального проектування систем баз даних.....	35
3.2.	Побудова організаційної моделі бази даних.....	38
3.3.	Побудова моделі інформаційного проходження бази даних.....	38
3.4.	Методи нормалізації інформаційних структур користувачів.....	42
3.5.	Метод побудови концептуальної структури бази даних.....	49
3.6.	Методи проектування логічних структур баз даних.....	51
3.7.	Методи розрахунку основних характеристик структур БД.....	59
3.8.	Створення графічних моделей баз даних концептуального і логічного рівнів.....	63
4.	РОЗРОБКА СТРУКТУРИ БАЗИ ДАНИХ АНАЛІТИЧНОГО ЦЕНТРУ АСУ БД.....	72
4.1.	Проектування загальної структури бази даних АСУ БД.....	72
4.2.	Поточне планування поїздоутворення та відправлення поїздів.....	77
4.3.	Можливості оперативного прогнозування виконання простою місцевого вагона на станції	84
4.4.	Розрахунок економічної ефективності впровадження АСК.....	90
	ВИСНОВКИ	93
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	95

ВСТУП

Зростання обсягів і складності оброблюваної інформації, збільшення вимог до достовірності і повноти інформації в автоматизованих системах (АС) обумовлює необхідність вдосконалення їх програмного та інформаційного забезпечення.

Інформаційні системи (ІС), що розробляються на основі автоматизованих баз даних, дозволяють утворювати гнучкі структури даних, що об'єднуються спеціальним чином - організовані бази даних (БД). Для організації високоефективної АС необхідно коректне і продумане проектування відносин в БД, що становить її основу. Кожна БД повинна володіти строго певною сукупністю значень параметрів, відповідати необхідним вимогам і мати структуру, яка містить надлишкові інформаційні елементи і взаємозв'язки.

В даний час не існує типової структури для АС і формалізованих методів проектування структур БД, тому їх розробка і реалізація є актуальним завданням.

Починаючи з 70-х років широке поширення набула технологія розробки та експлуатації АС, заснована на концепції автоматизованих баз даних [16,19].

База даних являє систему інформаційних, програмних, мовних і технологічних засобів, призначену для централізованого зберігання та колективного використання даних [21]. Основні складові автоматизованої БД: власне база даних і система управління базами даних.

База даних - структурована сукупність даних, що відноситься до конкретної предметної області (ПО) [9]. Для АС, де дані мають складну структуру (сотні і тисячі типів елементів) і велику розмірність (до 10 - 100 Гб і більше), вони організовуються в систему баз даних (СБД), що складається з безлічі щодо незалежних предметних баз даних (ПБД). Кожна ГТБД використовується кількома програмами.

Актуальність теми дослідження. Система управління базами даних (СКБД) - це спеціалізований комплекс програмних засобів, який забезпечує

введення, накопичення, зберігання, пошук і видачу даних [23].

Створення, впровадження та експлуатація БД є досить складною проблемою. У проектуванні БД виділяють два основні завдання [11, 22]: розробка БД і розробка СУБД. Наявність безлічі універсальних і спеціалізованих СУБД дозволило звести задачу розробки СУБД до вибору типової СУБД на відповідному етапі проектування, що задовольняє вимогам, що пред'являються.

Проектування БД для нового ПЗ є самостійною задачею. Відповідно до багаторівневої архітектури в розробці БД виділяють три етапи: концептуальне, логічне і фізичне проектування. Їх структура і основні завдання розглянуті в [9,29].

Концептуальне проектування полягає в специфікації інформаційних вимог користувачів і синтезі їх в проект системи баз даних. Логічне проектування пов'язано з побудовою схеми БД, що задовольняє пропонованим до неї вимогам. На етапі фізичного проектування визначаються характеристики методів доступу і структур зберігання щодо обраної логічної схеми.

Реляційні методи проектування [27] і методи відображення реляційних структур даних [27] в ієрархічні і мережні, а також кошти сучасних СУБД дозволили формалізувати й автоматизувати процеси логічного і фізичного проектування. Етап концептуального проектування є найменш розробленим і формалізованим [28]. У той же час це перший і найбільш важливий етап, результати виконання якого впливають на всі наступні фази розробки та якість готової системи ПБД [31,45].

З усього вищесказаного випливає, що розробка методів і засобів проектування БД на концептуальному етапі є актуальним завданням.

Вирішення цього завдання пов'язане з визначенням послідовності етапів розробки, а також сукупності застосовуваних методів і засобів. Склад завдань концептуального проектування БД розглядається в [16, 22]. Для систем БД це: опис інформаційних вимог, синтез концептуальної схеми (КС) БД, визначення структури СБД, планування її розробки. Однак включення того чи іншого кроку

в загальний процес проектування і вибір послідовності їх виконання є в даний час дискусійною проблемою.

Ефективність процесу і результатів концептуального проектування багато в чому залежать від рівня формалізації опису даних, який визначається концептуальною моделлю даних (КМД). Незважаючи на наявність ряду КМД різних класів: логічних, об'єктних, бінарних і т.д. [35, 38] проблема створення нових моделей БД зберігає свою актуальність [2, 13].

Мета і завдання дослідження - розробка методології проектування баз даних.

Поставлена мета досягається вирішенням наступних завдань:

- створення методу концептуального проектування систем баз даних;
- створення графічних моделей баз даних концептуального і логічного рівнів;
- створення методики логічного проектування баз даних.

Об'єкт дослідження - система однорідних БД, що представляють інформацію про програмне забезпечення складної структури і великої розмірності.

Предмет дослідження – методи організації баз даних в автоматизованій системі управління безпекою руху поїздів.

Методи виконання роботи. При виконанні роботи застосований комплекс методів дискретної математики, що включає методи теорії множин, і теоретико - графові моделі.

Наукова новизна отриманих результатів.

- удосконалена ефективна дворівнева модель БД, покладена в основу створення системи проектування баз даних, яка відрізняється від існуючих тим, що відображає інформаційні потреби користувача як на концептуальному, так і на логічному рівнях;
- запропонований метод багатокритеріальної оптимізації структур БД, що дозволяє проводити оптимізацію вже на першому етапі проектування, що значно звужує дерево пошуку оптимального варіанта структури;

- здійснено вибір і обґрунтування формалізованих методів і процедур, які є єдиною методологією проектування баз даних.

Практичне значення отриманих результатів. Впровадження запропонованої системи проектування дозволяє заощадити час розробки бази даних. Запропонована система проектування використана при проектуванні баз даних для АСУ БД.

Публікації. Відповідно до теми дипломної роботи опубліковані наукові публікації у фахових виданнях України, результати роботи представлені на науковій студентській конференції.

Структура і об'єм роботи. Кваліфікаційна робота магістра складається зі вступу, 4 розділів, заключення, списку використаних джерел з 64 найменувань на 6 сторінках. Загальний об'єм кваліфікаційної роботи магістра складає 100 стор. Робота включає 23 рисунка та 7 таблиць по тексту.

1. АНАЛІЗ АВТОМАТИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ БЕЗПЕКОЮ РУХУ ПОЇЗДІВ

1.1. Аналіз функцій системи безпеки руху поїздів

З грудня 1997 року введено в постійну експлуатацію на мережі залізниць України інформаційно-довідкова підсистема багаторівневої автоматизованої системи управління безпеки руху поїздів (ІСП масу БД), призначена для оперативного інформування працівників контрольного апарату залізниць та ЦРЛ МІС про випадки і причини порушень безпеки руху для розробки заходів щодо їх запобігання. В даний час за допомогою цієї системи щодня в управліннях залізниць, ЦРЛ МІС інших Департаментах складаються довідки про порушення безпеки руху «Оперативні донесення про випадки шлюбу в поїзної і маневрової роботи», що містять відомості про абсолютне числі розслідуваних шлюбів в поїзної і маневрової роботи за кожні минулу добу з розподілом по господарствах. У цих довідках наводяться також порівняльні дані про кількість порушень безпеки руху (наростаючим підсумком) з початку поточного місяця і року по поточну дату і аналогічно - за минулий рік. Ці відомості використовуються керівництвом МІС для оцінки поточного стану безпеки руху та оперативного відстеження (моніторингу) тенденцій його зміни для своєчасного вжиття запобіжних заходів щодо недопущення погіршення стану безпеки руху. Недоліками такої системи управління безпекою руху є: Ці відомості використовуються керівництвом МІС для оцінки поточного стану безпеки руху та оперативного відстеження (моніторингу) тенденцій його зміни для своєчасного вжиття запобіжних заходів щодо недопущення погіршення стану безпеки руху. Недоліками такої системи управління безпекою руху є: Ці відомості використовуються керівництвом МІС для оцінки поточного стану безпеки руху та оперативного відстеження (моніторингу) тенденцій його зміни для своєчасного вжиття запобіжних заходів щодо недопущення погіршення

стану безпеки руху. Недоліками такої системи управління безпекою руху є:

- оцінка стану безпеки руху проводиться за абсолютними значеннями числа порушень (без урахування обсягів виконаної експлуатаційної роботи, значною мірою визначає число порушень безпеки руху);
- при оцінці стану безпеки руху за абсолютними значеннями числа шлюбів не враховується статистичне розсіювання показників безпеки руху (абсолютних і відносних), властиве їм як випадкових величин, тобто прийняття рішення про зміну (і, зокрема, погіршення) стану безпеки руху, в порівнянні з аналогічним періодом минулого року не пов'язане з побудовою довірчих інтервалів для показників аварійності;
- неможливо своєчасне виявлення тенденцій до погіршення безпеки руху та прийняття відповідних коригуючих заходів при відносно малому щодобовому числі окремих видів шлюбу (з урахуванням розподілу їх загальної кількості у видах, залізницям та іншими показниками).

У запропонованій автоматизованій паспортній системи безпеки руху нового покоління АСУ БД виключаються всі зазначені недоліки за рахунок використання для аналізу безпеки руху імовірнісних значень показників безпеки руху, які допускають можливість їх статистичної обробки методами теорії ймовірності та математичної статистики.

Передача інформації в масу БД здійснюється за допомогою повідомлень, які накопичуються на великій машині сервері, тому база даних являє собою сукупність масивів даних, що складаються з набору послідовно розташованих повідомлень. Така організація даних має нереляційних структуру, тому при такій організації неможливо використання сучасних методів проектування.

1.2. Інформаційні функції управління безпекою руху та нагляду автоматизованої АСУ БД

До інформаційних функцій управління безпекою руху, що автоматизуються АСУ БД відносять [65]:

- 1) передача і зберігання первинної інформації про кількість, причини та наслідки переходів руху в небезпечні стану;
- 2) передача і зберігання первинної інформації про кількість небезпечних відмов технічних засобів, причини їх виникнення, впливів (не вплине) їх на рух поїздів, про наслідки впливів;
- 3) передача і зберігання первинної інформації про кількість небезпечних помилок персоналу залізниць, їх причини, про впливи їх на рух поїздів, про наслідки впливів;
- 4) збір, передача і зберігання первинної інформації про неприпустимі за умовами безпеки руху відхилення параметрів технічних засобів від їх номінальних значень;
- 5) автоматизуються частково - збір, передача і зберігання первинної інформації про неприпустимі за умовами безпеки руху відхилення психофізіологічних параметрів машиністів і чергових по станціях від їх номінальних значень;
- 6) автоматизуються частково - збір, передача і зберігання первинної інформації про обсяг поїзної роботи на окремих ділянках і обсязі маневрової роботи окремих станцій;
- 7) автоматизуються частково - збір, передача і зберігання первинної інформації про кількість технічних засобів певного виду на окремих ділянках і про напрацювання кожного з них до небезпечної відмови;
- 8) автоматизуються частково - збір, передача і зберігання первинної інформації про кількість фахівців окремих професій на окремих ділянках і про напрацювання кожного з них до небезпечної помилки;
- 9) попередня обробка первинної інформації перед її передачею до пристроїв, що здійснюють розрахунок відносних показників безпеки руху поїздів;
- 10) передача первинної інформації від місць її зародження до пристроїв, що здійснюють розрахунок відносних показників безпеки руху поїздів;
- 11) розрахунок відносних показників фактичної безпеки руху поїздів різних

видів за окремими маршрутами і ділянкам залізниць;

12) розрахунок відносних показників фактичної безпеки функціонування технічних засобів різного типу і призначення;

13) розрахунок відносних показників фактичної безпеки функціонування фахівців різних професій; прогнозування значень показників безпеки руху поїздів;

14) прогнозування показників безпеки функціонування технічних засобів.

15) прогнозування показників безпеки функціонування фахівців .

До інформаційних функцій нагляду, автоматизуються АСУ БД відносять [65]:

1) збір первинної інформації про порушення нормативних документів, що регламентують оперативне управління рухом поїздів в реальному масштабі часу (машиністами поїздів і черговими по станціях);

2) збір первинної інформації про порушення нормативних документів, що регламентують роботи з поточного утримання технічних засобів;

3) збір первинної інформації о порушеннях нормативних документів, що регламентують роботи по ремонту технічних засобів.

4) збір первинної інформації про порушення нормативних документів, що регламентують підтримку професійного рівня персоналу залізниць, чия діяльність впливає на безпеку руху поездов.1.3. Керуючі функції управління безпекою руху та нагляду, автоматизує АСУ БД.

До керуючим функцій управління безпекою руху, що автоматизуються АСУ БД відносять [65]:

1) вироблення рекомендацій щодо забезпечення безпеки руху при виявленні відхилень параметрів технічних засобів від нормативних значень на неприпустимі величини за умовами безпеки руху;

2) вироблення рекомендацій щодо забезпечення безпеки руху при виявленні відхилень психофізіологічних параметрів машиністів поїздів і чергових по станціях від нормативних значень на неприпустимі величини за умовами безпеки руху.

1.3. Керуючі функції управління безпекою руху

До керуючих функцій управління і нагляду, автоматизованих АСУ БД відносять [65]:

- 1) вироблення сигналу, що інформує осіб, відповідальних за безпеку на ділянці шляху або на станції, якщо в межах цієї ділянки або станції виявлено порушення нормативних документів, що регламентують експлуатацію, поточного утримання та ремонту технічного засобу;
- 2) вибір з безлічі відомих заходів запобігання переходу руху в небезпечний стан внаслідок цього порушення найбільш раціонального і вироблення відповідних рекомендацій.

Для реалізації перерахованих функцій необхідна база даних, яка містила б всю необхідну інформацію. Внаслідок цього завдання роботи зводиться до аналізу інформаційних і керуючих функцій системи, виділення предметної області даних і розробки концептуальної і логічної структур банку даних використовуючи обрану методологію побудови баз даних, а також створення фізичної моделі даних з подальшим впровадженням. Для реалізації завдання в роботі необхідно проаналізувати проблеми технології побудови баз даних, що виникають при проектуванні. У другому розділі потрібно провести повний аналіз моделей, методів і методик проектування баз даних, враховуючи що етап концептуального проектування є найменш розробленим і формалізованим. У той же час це перший і найбільш важливий етап, результати виконання якого впливають на всі наступні фази розробки та якість готової системи БД.

З висновків першої та другої глав робиться висновок про необхідність розробки власного методу проектування, який включає в себе побудову організаційної моделі, моделі інформаційного проходження і подальшої розробки моделі концептуального і логічного проектування. Для відображення отриманих моделей необхідне створення графічних моделей баз даних концептуального і логічного рівнів, що наведено в третьому розділі.

2.АНАЛІЗ ПРОБЛЕМ ТЕХНОЛОГІЇ ПОБУДОВИ БАЗ ДАНИХ

2.1.Аналіз проблем проектування баз даних

Застосування ЕОМ обробки даних зазвичай призводить до поділу даних і їх смислового змісту. ЕОМ має справу, головним чином, з даними як такими. Велика частина змісту взагалі не фіксується в явній формі (ЕОМ не "знає", чи є «21.50» вартістю квитка або часом відправлення).

Існує принаймні дві історичні причини, за якими застосування ЕОМ призвело до відділення даних від їх семантики.

По-перше, ЕОМ не мали достатніми можливостями для обробки текстів.

По-друге, вартість пам'яті ЕОМ була спочатку досить велика. Пам'ять використовувалася для зберігання самих даних, а семантика традиційно покладалася на користувача. Користувач закладав семантику даних в свою програму, яка «знала», наприклад, що шість вводиться значення в ЕОМ пов'язано з часом відправлення поїзда, а четверте - з часом його прибуття. Це істотно підвищувало роль програми, так як поза семантики дані представляють собою не більш ніж сукупність бітів в пристрої, що запам'ятовує.

Жорстка залежність між даними і використовують їх програмами робить використання їх менш гнучкими.

Розробники прикладних програм (написаних, наприклад, на Бейсике, Паскалі або Сі) розміщують потрібні їм дані в файлах, організовуючи їх найбільш зручним для себе чином. При цьому одні й ті ж дані можуть мати в різних додатках абсолютно різну організацію (різну послідовність розміщення в запису, різні формати одних і тих же полів і т.п.). Усуспільнити такі дані надзвичайно важко: наприклад, будь-яка зміна структури записи файлу, вироблене одним з розробників, призводить до необхідності зміни іншими розробниками тих програм, які використовують записи цього файлу.

Активна діяльність з відшукування способів усуспільнення безупинно зростаючого обсягу інформації привела до створення на початку 60-х років

спеціальних програмних комплексів, званих Системи управління базами даних (СКБД).

СУБД являє собою сукупність процедур для введення і збереження не тільки самих даних, але і описів їх структури. Файли, забезпечені описом збережених у них даних і знаходяться під управлінням СУБД, стали називати банками даних, а потім базами даних (БД).

При виконанні запиту на читання даних, виданого прикладної програмою, СУБД виконує ряд дій:

- інтерпретацію запиту;
- пошук всіх описів даних, на які видано запит;
- формування команд, за якими операційна система пересилає з запам'ятовуючих пристроїв в буфера СУБД вміст всіх фізичних записів з необхідними даними;

виділення з цих записів потрібних даних, їх форматування (якщо це необхідно), створення заданого виду і послідовності виведення і пересилання на термінал або в робочу область прикладної програми

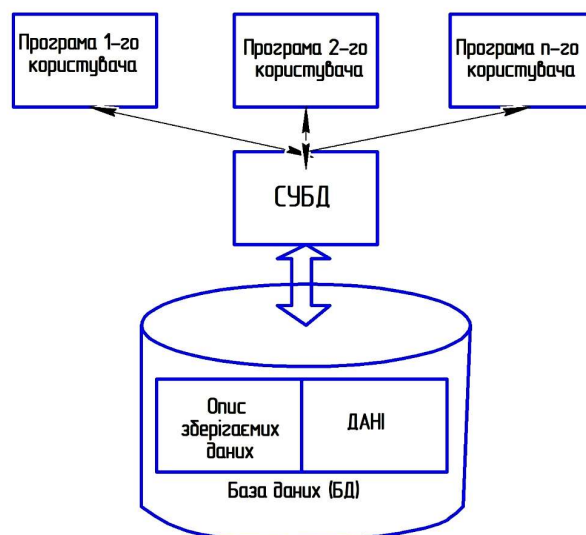


Рис.2.1. Зв'язок програм і даних при використанні системи управління базою даних (СУБД)

Аналогічні дії виконуються при оновленні або виведення даних. На обмін даними через СУБД потрібно більше часу, ніж на обмін аналогічними даними прямо з файлів, спеціально створених для того чи іншого додатка СУБД повинна надавати доступ до даних будь-яким користувачам, включаючи і тих, які практично не мають і (або) не хочуть мати уявлення про:

- фізичному розміщенні в пам'яті даних і їх описів;
- механізмах пошуку запитуваних даних;
- проблемах, що виникають при одночасному запиті одних і тих же даних багатьма користувачами (прикладними програмами);
- способах забезпечення захисту даних від некоректних оновлень і (або) несанкціонованого доступу;
- підтримці баз даних в актуальному стані;
- і безлічі інших функцій СУБД.

При виконанні основних з цих функцій СУБД повинна використовувати різні описи даних. Природно, що проектування бази даних треба починати з аналізу предметної області та виявлення вимог до неї окремих користувачів. Проектування зазвичай доручається людині (групі осіб) - адміністратора бази даних (АБД).

В результаті аналізу технологічного процесу, об'єднуючи приватні уявлення про вміст бази даних, отримані в результаті опитування користувачів, і свої уявлення про дані, які можуть знадобитися в майбутніх додатках, АБД створює узагальнене неформальний опис створюваної бази даних. Це опис, виконане з використанням природної мови, математичних формул, таблиць, графіків і інших засобів, зрозумілих всім людям, які працюють над проектуванням бази даних, називають інфологической моделлю даних (рис.2.2).

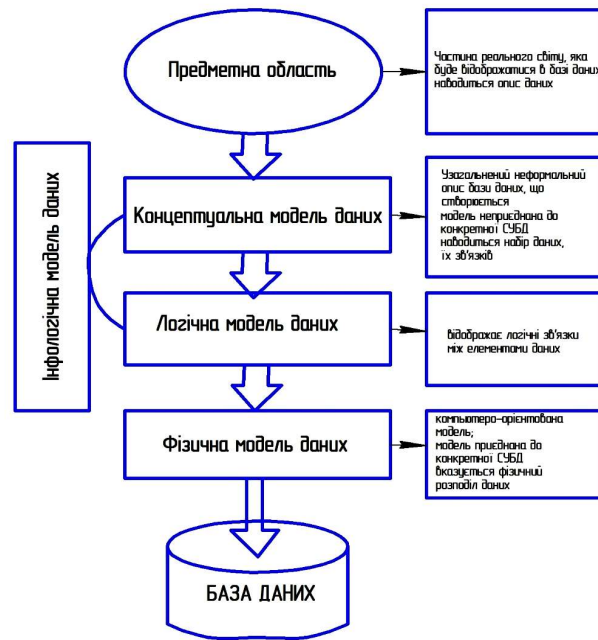


Рис.2.2. Етапи проектування бази даних

У сучасній технології баз даних передбачається, що створення БД, її підтримка і забезпечення доступу користувачів до неї здійснюється централізовано за допомогою СУБД. Ядром БД є модель даних - сукупність структур даних та операцій їх обробки. Одну і ту ж інформацію можна представити за допомогою різних засобів. Моделі в інформатиці призначені для спільного використання різними людьми, і, крім того, їх повинна «розуміти» ЕОМ.

В даний час загальноприйнятою є трирівнева архітектура БД, запропонована дослідницькою групою інституту стандартів по системах управління базами даних АН81 / ХЗ / 8РАЯС і робочою групою по базах даних Кодас, згідно з якою виділяються три рівня моделювання БД: концептуальний, логічний і фізичний [35]

2.2. Аналіз методик проектування баз даних на концептуальному рівні

Створення і ведення БД, особливо для великих АС, - складний процес, що вимагає праці колективу фахівців відповідної кваліфікації протягом значної

кількості часу. Процес проектування БД являє собою послідовність взаємопов'язаних кроків. Відповідно до традиційного підходу [9, 16], його структура визначається багаторівневою організацією даних і включає етапи концептуального, логічного і фізичного проектування [58].

Концептуальне проектування полягає у вивченні ПО, визначенні, моделюванні та поданні інформаційних потреб різних груп користувачів і їх аналізі. Результатом етапу концептуального проектування є концептуальна схема ПО, що представляє високорівневе визначення вимог до БД.

Логічне та фізичне проектування пов'язані з відображенням отриманої концептуальної схеми в структури даних обраної СУБД. На етапі логічного проектування виконується опис схеми на мові опису даних. Фізичне проектування включає визначення ознак упорядкування записів, вибір методу доступу, розміщення даних на зовнішніх носіях, а також ряд інших процедур.

Загальний аналіз етапів проектування БД дан в [28, 29]. Там же зазначається, що найменш дослідженим і в той же час найбільш важливим є етап концептуального проектування. Його формалізація та автоматизація пов'язані з вирішенням таких проблем, як визначення складу і послідовності вирішення завдань концептуального проектування, методів і засобів їх вирішення, переліку необхідних для проектування даних [18, 57, 59].

Комплексну автоматизацію процесу концептуального проектування можна здійснити, розглядаючи створення БД як складову частину розробки АС. При створенні АС можливе застосування одного з трьох підходів [26].

Перший підхід є традиційним і визначається життєвим циклом системи, що включає фази аналізу або передпроектного обстеження, проектування, введення в експлуатацію, експлуатації та розвитку системи [28,31]. В цьому випадку етап концептуального проектування БД збігається з фазою аналізу, а логічного і фізичного - зі стадією проектування.

Недолік підходу полягає в тому, що користувач бачить кінцевий результат тільки на етапі введення в експлуатацію і не має можливості виявити помилки, допущені на стадіях аналізу і проектування, зокрема, пов'язані з нерозумінням

або неправильним відображенням вимог до системи. Це призводить до створення системи, непридатною до використання.

Другий підхід до проектування набуває все більшого поширення [43, 47]. Він заснований на тому, що користувач може краще визначити і сформулювати свої вимоги, маючи готову систему. АС визначається через безліч ітерацій, кожна з яких включає всі фази життєвого циклу системи. Результатом ітерації є робоча модель системи, досліджуючи яку користувач виявляє недоліки і знову формулює свої вимоги.

Відсутність в даний час автоматизованих засобів, що дозволяють проектувальнику в короткі терміни побудувати модель робочої системи, обмежує застосування цього підходу при створенні великих АС [52] і призводить до поширення різних комбінованих методик [39,40].

Суть комбінованого підходу полягає в тому, що початкові фази життєвого циклу системи розглядаються як ітераційні процедури [39]. Найчастіше такі процедури використовуються на фазі аналізу і її окремих кроках. Застосування цього підходу до фази проектування полягає в тому, що при першому аналізі з низьким рівнем деталізації система розбивається на безліч підсистем, для кожної з яких послідовно проводиться детальне проектування [26] і усуваються виявлені недоліки.

Як показало дослідження існуючих методик проектування АС, перелік яких дано в таблиці 2.1, обидва комбінованих підходу застосовуються і при розробці їх інформаційного забезпечення.

Розглянемо докладніше етап аналізу системи. У його складі можна виділити наступні кроки, які стосуються концептуального проектування систем баз даних (СБД):

- визначення та специфікація потреб користувачів;
- аналіз вимог;
- проектування структури СБД;
- планування розробки системи.

Треба відзначити, що розглянуті методики не включають всі перелічені вище завдання і відрізняються послідовністю їх виконання та рівнем їх автоматизації. Склад кроків кожної методики і рівень їх автоматизації дані на рис.2.3.

Процес визначення і опису вимог є першим кроком в методиці проектування. В [39] виділяють дві складові цього процесу: вивчення ПО і специфікація вимог, встановлюється безліч можливих користувачів системи. Згідно з існуючими методиками ПО може бути розглянута як:

- безліч співробітників, кожен з яких виконує певні функції [55];
- життєвий цикл продукту, що випускається, на основі якого визначається безліч областей діяльності і їх функції (BSP);
 - безліч замовлень підприємства (BICS);
 - безліч функцій і цілей (OAM);
- структура організації і безліч впливають на неї факторів, таких, як оточення, технологія та інші (OARSM).

Специфікація вимог пов'язана з їх формальним визначенням в термінах однієї моделі або комплексу моделей, зрозумілих системним аналітиком і кінцевим користувачам. Сама специфікація включає дві складові: опис процесів обробки даних і опис самих даних. Відповідно до цього розрізняють методики, орієнтовані на процеси (ТПВ), орієнтовані на дані (МAM, БОБ) і змішані (Е2Б2,1САМ).

У більшості методик (1ББОБ, ОAM, 1Сам, БВМБ) процес специфікації нерозривно пов'язаний з процесом аналізу вимог, мета якого - встановити неповноту і неточність опису системи, виявити і вирішити конфлікти в поглядах різних користувачів, і представляє безліч ітерацій.

Діаграми проходження даних є графічним представленням проходження інформації в системі і показують: які дані є вхідними, як і якими функціями вони користуються, де і коли змінюються. Ці діаграми є важливим інструментом системного аналізу, так як дозволяють визначити джерело даних,

первинні і похідні дані, встановити безліч загальних для декількох функцій даних.

Діаграми процесів, транзакцій і подій показують як дані змінюються в системі з плином часу. Для кожного типу даних (елемента або безлічі елементів) повинні бути вказані події і дії, що визначають їх створення, використання і зміна. В іншому випадку робиться висновок про неповноту специфікації системи. Конкретний вид діаграм залежить від виду використовуваної моделі даних.

Модель даних (МД) займає центральне місце в проектуванні БД серед всіх перерахованих коштів. Застосовувані в методиках МД дуже різні: від простих діаграм даних, що містять елементи даних і зв'язку між ними, які використовуються в орієнтованих на процеси методиках (1БАС, бабіт), до високорівневих засобів, що дозволяють описувати складні структури даних, їх відносини і процеси їх обробки (Батогом , Б2Б2, ИРИС), які прийнято називати концептуальними моделями даних [44, 47, 48].

Існуючі в даний час концептуальні моделі даних (КМД) [56] надають різні можливості опису семантики даних, тому вибір моделі - одна з проблем процесу концептуального проектування, так як застосовується МД впливає на вибір інших засобів і на методи вирішення завдань проектування, зокрема , на методику інтеграції схем даних.

Інтеграція схем - це процес проектування концептуальної структури (КС) даних БД як інтегрованої сукупності даних багатьох користувачів, що є складовою частиною аналізу вимог. Більшість методик проектування включає цей крок. Однак пов'язана з ним ідентифікація синонімів, омонімів і тавтологію елементів даних і їх зв'язків важко піддається автоматизації і формалізації, тому інтеграція проводиться проектувальниками з використанням, в кращому випадку, спеціальних методик [51, 57] і програмного забезпечення як допоміжний засіб [21].

Огляд існуючих методик інтеграції схем користувачів і пов'язаних з цим процесом проблем дан в [37]. У загальному випадку, будь-яка методика

зводиться до встановлення відносин між об'єднуються схемами (ідентичності, еквівалентності, сумісності, несумісності). Залежно від встановленого відносини здійснюється або об'єднання схем і визначення додаткових межсхемних відносин, або виявлення і вирішення конфліктів різних уявлень, таких, як конфлікти іменування (синонімія, омонімія) і структурні конфлікти (типів, ключів, залежностей).

Чим простіше застосовується МД, тим менше моделюють конструкцій, властивостей, обмежень вона використовує і, отже, тим менше можливість конфліктів типів даних, простіше операції трансформації схем і сама методика [38]. З іншого боку, в порівнянні з моделями, що мають безліч конструкцій і механізм абстракцій [20], прості моделі є слабким інструментом в руках проектувальників для встановлення подібності, відмінності та сумісності схем.

Відповідно до застосовуваної МД розрізняють два підходи до інтеграції схем [39]. Перший використовує реляційну або функціональну моделі даних і зводить інтеграцію до побудови універсального відносини для всієї ПО [38]. При цьому передбачається унікальність імен елементів даних у всій ПО, що не дає можливості відобразити представлення різних користувачів. Другий підхід призначений для семантичних МД, а саме моделей типу «сутність- зв'язок», зводиться до аналізу функціональних залежностей даних і використовує семантичну виразність засобів застосовуваного класу моделей [37, 11].

Для документування та аналізу вимог, описаних в термінах розглянутих моделей: організаційної моделі, діаграм процесів, КМД використовуються формальні мови. У великих системах, де повний опис ПО становить, як правило, сотні і тисячі операторів мови, без застосування цього засобу провести аналіз вимог і визначити помилки специфікацій не представляється можливим. Використання формальних мов створює передумови для автоматизації процесу аналізу.

В даний час основним інструментом підтримки процесу проектування АС стали словники даних або словники-довідники даних (ССД). Система ССД (СССД) являє собою програмний засіб, який призначений для логічної

централізації даних про інформаційні ресурси: опис властивостей, взаємозв'язків і характеристик використання даних, - і є інформаційним ядром будь-якої автоматизованої системи проектування (REVS, ISDOS, BSP і т.д.) .

На етапі специфікації і аналізу вимог користувачів СССД служить для документування описів в міру їх надходження і забезпечує контроль цілісності даних. Програмні засоби СССД дозволяють організовувати ефективну взаємодію між проектувальником і кінцевим користувачем, отримувати різного роду стандартні і довільні звіти за зібраними даними, контролювати хід розробки. Використання СССД орієнтує розробників на застосування прогресивних структурних методів роботи і автоматизованих процедур вирішення завдань проектування.

В даний час існують промислові СССД, оформлені у вигляді пакета прикладних програм (111111), в складі конкретних СУБД (IDMS, Oracle) або незалежні (TIS, VCC - 10) [2]. Основні типи даних, підтримуваних такими СССД: запис, файл, елемент, модуль, програма, фізичний пристрій і т.д. Засоби розширення структури даних, наявні в деяких системах, не дають можливості вводити абсолютно новий набір типів об'єктів і атрибутів. Отже, існуючі системи розраховані на супровід функціонуючих БД і можуть при невеликих змінах використовуватися на етапах логічного і фізичного проектування. Для автоматизації розробки на концептуальному рівні виникає необхідність створення спеціалізованих СССД [38], орієнтованих на конкретну методику моделі.

На відміну від двох розглянутих кроків аналізу системи, які включені практично в усі методики проектування, наступні два кроки: формування структури системи бази даних (СБД) та планування її розробки, розглядаються лише в ISAC, BSP і орієнтовані на створення великих АС і застосування комбінованого підходу на етапі проектування БД.

Для великих АС, де дані мають великий обсяг і складну структуру, створення єдиної інтегрованої БД в даний час є фактично недосяжною завданням [23]. Тому інформаційне забезпечення таких АС складається з

безлічі щодо незалежних ПБД. Кількість ПБД залежить від розміру ПО. Такий підхід дозволяє:

- прискорити розробку системи, так як проєктовані БД менш складні і обслуговують кілька додатків;
- на етапі проєктування вирішувати завдання вибору СУБД для кожної ПБД, виходячи зі структури даних і необхідних характеристик, розподілу СБД по вузлах мережі в разі розподілу обробки;
- підвищити ефективність системи в цілому, отримуючи віддачу від уже впроваджених ПБД;
- підвищити ефективність супроводу системи, так як зменшення обсягу ПБД і їх незалежність полегшують контроль цілісності і доступу до даних, знижують витрати часу на зняття копій, процедури відкоту і відновлення;
- знизити витрати на процес інтеграції схем, якщо він проводиться після кроку формування структури СБД.

Часто вміст ПБД вибирається без застосування будь-якої формальної процедури [9, 11], виходячи зі здорового глузду і досвіду проєктувальників. Формалізовані процедури вибору структури СБД в якості вихідних даних використовують або тільки елементи даних і їх зв'язку [9], або КСД і зв'язку «процес-дані» [26].

Перший підхід спирається на властивості застосовуваної КМД. В [50] вихідної є матриця зв'язків між елементами даних, а декомпозиція здійснюється із застосуванням теорії нормалізації, що дозволяє отримати мінімальне з'єднання між елементами різних ПБД. Метод, викладений в [56], розроблений для моделей типу «сутність- зв'язок». Всі типи сутностей класифікуються спочатку як головні і похідні, а розбиття на ПБД здійснюється як декомпозиція відносин між головними сутностями.

Використання зв'язків «процес-дані» дозволяє ькласифікувати зв'язку між об'єктами [26], отримувати заходи подібності інформаційних елементів [17] або заходи близькості структур двох додатків. На основі цих заходів, застосовуючи методи кластерного аналізу, розбивають елементи даних або програми на

непересічні групи, при цьому можуть враховуватися обмеження на розміри одержуваних ПБД, на частоту використання даних.

Згідно розглянутому на початку параграфа комбінованого підходу до проектування розробка і впровадження ПБД проводяться послідовно. Це обумовлено ще і тим, що на проектування системи, зазвичай, виділяються обмежені ресурси. Тому виникає необхідність планування розробки ПБД і пов'язаних з ним програм. Перевага в плані повинна віддаватися ПБД з швидко досягаються наочними результатами.

Аналіз факторів, що впливають на пріоритет розробки ПБД, дан в [9]. До них відносяться витрати на проектування і терміни проектування, при виділених ресурсах на розробку, взаємозв'язку ПБД по процесам обробки, наявні в наявності системи, очікуваний прибуток від впровадження, різні непрямі доходи, які важко оцінити в грошовому вираженні. Планування розробки в методиках, що включають цей крок (BSP, D2S2, ISAC), здійснюється неформальними методами.

Проведений аналіз існуючих методик і проблем концептуального проектування БД складної структури і великої розмірності дозволяє зробити наступні висновки:

1. Існуючі методики проектування орієнтовані, в основному, на процеси специфікації і аналізу вимог. Для створення БД складної структури і великої розмірності необхідно розробити методику концептуального проектування, що передбачає комплексне вирішення всіх розглянутих завдань і можливість зниження з розмірності.
2. Так як центральне місце серед засобів концептуального проектування БД займає МД, і від її виду залежить виконання багатьох кроків методики, то доцільно провести аналіз і обґрунтувати вибір КМД.

2.3. Аналіз моделей даних концептуального рівня

Модель даних відіграє ключову роль на всіх етапах проектування БД. Під МД розуміється безліч базових концепцій, правил породження і операцій маніпулювання даними, тобто сукупність методів і засобів, що дозволяють абстрагуватися від істоти фізичних об'єктів і їх взаємозв'язків і описати інформаційний зміст ПО [18, 19,22].

Кожна модель пропонує свій підхід до опису схеми даних (СД) ПО. Однак, в загальному випадку, вона повинна визначати [29, 32]:

- статичні властивості (структуру) даних;
- динамічні властивості даних;
- обмеження цілісності.

Структурна частина МД описується мовою визначення даних (МОД) і використовує такі концепції [29]:

- структурні елементи для визначення елементарних об'єктів, їх властивостей і класифікації об'єктів;
- структурні відносини для визначення зв'язків між об'єктами;
- структурні абстракції для опису складних об'єктів, створених над структурними елементами і відносинами.

Динамічна частина МД описується мовою маніпулювання даними (ЯМД) і визначає:

- елементарні оператори доступу, включення, виключення даних;
- правілакомпозиціїелементарнихоператоров: послідовність, вибір та інші;
- складні оператори (транзакції) контролю цілісності, обчислення значень, опису запитів та інші.

Обмеження цілісності представляють логічні обмеження, які вводяться в МД для підвищення її семантичності. Найбільш поширеними є обмеження на значення даних і на зв'язку між даними. За способом завдання вони поділяються на внутрішні (базові властивості концепцій МД), явні, що задаються користувачем в явному вигляді, і неявні, які є наслідком перших двох

[29]. Ці обмеження можуть бути представлені декларативно при описі структури та / або процедурно в динамічній частині моделі.

Вибір МД визначається відповідністю наданих нею можливостей і засобів цілям і області її застосування [30]. КМД повинна задовольняти деяким загальним вимогам:

- описувати тільки концептуально важливі аспекти, не торкаючись представлення даних і їх фізичної реалізації;
- забезпечувати багаті і виразні описи, за допомогою яких можна було б визначити всі пов'язані з ПО правила, поняття та обмеження;
- враховувати особливості наступних етапів проектування БД.

Для цілей аналізу існуючих КМД в роботі використовуються наступні характеристики:

1. Можливості МД за визначенням складних структур даних ПО і можливості ЯМД за описом динамічних властивостей.
2. Формальне математичне визначення основних концепцій моделі, що забезпечує їх однозначне тлумачення.
3. Формальний опис семантики ЯОД і ЯМД, що базуються на наборі первинних концепцій МД.
4. Простота і наочність коштів, наданих МД, тобто невелике число основних концепцій, наявність графічної інтерпретації структурної та динамічної частини СД.
5. Зв'язок з наступними етапами проектування. МД повинна підтримуватися хоча б однієї з існуючих СУБД, або повинні існувати правила і процедури відображення в МД логічного рівня: ієрархічну, мережеву, реляційну.

В даний час існує понад сорок МД, використовуваних на етапі концептуального проектування і суміжних з ним етапах [53].

Для аналізу КМД в роботі використовується класифікація МД, наведена на рис.2.3 [41]. Відповідно до класифікації МД розбиваються за двома ознаками:

- за рівнем вимог до категоризації даних на структуровані і неструктуровані моделі [35];
- по використанню формальної системи для опису даних на математичні та евристичні [27, 56]

<u>Моделі</u>	<u>Неструктуровані МД</u>		<u>Структуровані моделі</u>				
<u>Математичні моделі</u>	<u>Логічні моделі</u>		<u>Теоретико-множинні моделі, засновані на абстрактних типах даних</u>				
<u>Евристичні моделі</u>	<u>Теоретико-графові моделі; семантичні мережі; фреймові моделі</u>		<u>Семантичні моделі</u>				
	<u>Структуровані вершини</u>	<u>Неструктуровані вершини</u>	<u>Моделі, що не використовують «зв'язок»</u> <u>Об'єктні моделі</u>	<u>Моделі типу «об'єкт – зв'язок»</u>			
				<u>МД з парними зв'язками</u>		<u>Бінарні МД</u>	
			<u>МД з атрибутами для об'єктів зв'язків</u>	<u>МД з атрибутами тільки для об'єктів</u>	<u>МД з атрибутами для об'єктів «1:n»</u> <u>«n:m»</u>	<u>МД без атрибутів</u>	

Рис.2.3. Класифікація концептуальних моделей даних

До математичних неструктурованих МД відносяться логічні моделі. Найбільш поширені моделі, засновані на обчисленні предикатів першого порядку. У них об'єкти представляються у вигляді термів, а властивості, зв'язку, обмеження - у вигляді логічних висловлювань. Існує БД фактів, висловлювання про які обчислюються для відповіді на запит. Зміни в додатку здійснюються шляхом додавання логічних висловлювань. Такі МД мають наступні переваги:

- спільність уявлення, так як запити, обмеження цілісності і факти подаються одним і тим же мовою предикатів першого порядку;
- єдиний механізм для обробки запитів і задоволення обмежень цілісності;
- можливість виведення даних, які не містяться явно в БД.

Ефективна реалізація логічного підходу залишається невирішеною проблемою, особливо для дуже великих БД з частим оновленням.

До евристичних неструктурованих МД відносяться теоретикографові моделі, семантичні мережі і фреймових моделі.

Завдання концептуального проектування БД

Завдання	Що досягається вирішенням даного завдання
<p>Виявлення і повний опис інформаційних об'єктів і їх взаємозв'язків, що представляють інтерес в аналізованій предметній області ПЗ.</p>	<p>Адекватність відображення моделлю реальних властивостей даних і їх взаємозв'язків.</p> <p>Стабільність моделі.</p>

У теоретико-графових моделях зовнішній світ моделюється графом. Вершини такого графа можуть бути як структуровані, так і не структурована. Структуровані вершини - це вершини, які відповідають типам записів або кортежів, які можуть бути розділені на елементи даних такі як, наприклад, записи мережевий моделі даних (БВТО). Неструктуріруемые вершини - це вершини, відповідні елементам записи, полів запису. Ребра в теоретико-графових моделях є кошти переходу від однієї вершини до іншої і можуть розглядатися як бінарні відносини на множині вершин.

У деяких моделях даних на ці відносини накладаються вимоги функціональності, тобто має бути визначено, що в відношенні є аргументом, а що значенням функції, що відповідає цьому аргументу. Таке обмеження характерно для мережевий, ієрархічної та інших моделей даних.

У деяких моделях даних вводяться обмеження на вид графа. У графі моделі даних Об'єкт-ставлення є три типи вершин, і ребра можуть з'єднувати тільки вершини різних типів, тобто граф повинен бути тріхдольним. Граф ієрархічної моделі даних повинен бути об'єднанням дерев.

У моделях, заснованих на абстрактних типах даних, база даних представляється як накопичувач типів даних [20]. Кожен тип даних визначає безліч об'єктів за допомогою безлічі операцій над ними і безлічі правил: аксіом і обмежень, що вказують допустимі поєднання або прийнятність операцій до того чи іншого об'єкту-станом типу даних.

Евристичні Структуровані МД отримали назву семантичних [57, 63]. Вони перевершують математичні по можливості адекватного опису ПО через різноманітного набору наданих ними коштів і набули найбільшого поширення при проектуванні БД. Семантичні МД з'явилися як розвиток класичних моделей: ієрархічної, мережевої, реляційної [2, 27, 33]. З метою посилення можливостей моделювання в них використовуються структурні абстракції: узагальнення [36], агрегації [38, 39], асоціації (групування) [56]. Абстракції дозволяють будувати більш загальні категорії і безпосередньо, в декларативній частині СД, описувати складні об'єкти, їх властивості та обмеження цілісності. За своїми структурними характеристиками, тобто від прийнятого способу опису об'єктів, зв'язків і властивостей, семантичні моделі розбиваються на підкласи (див. Рис.2.4).

МД, в яких зв'язки не вказуються явно, відносяться до об'єктним моделям: RM / T [48], SDM [57], TAXIS. Вони з'явилися як розширення класичної реляційної моделі. Основними концепціями моделей цього класу є поняття властивостей (атрибутів) і відносин. У RM / T, SDM відносини використовуються для опису об'єктів і зв'язків між ними, а зв'язки між відносинами задаються через однакові значення атрибутів. У TAXIS зв'язку між типами об'єктів і обмеження на зв'язку задаються процедурно за допомогою транзакцій. SDM і TAXIS надають великі можливості по специфікації прикладних процесів. Основна перевага об'єктних МД - мале число основних концепцій, які використовуються для опису ПО, і властивий їм семантичний релятивізм, який дозволяє легко узгоджувати погляди окремих користувачів.

МД, в яких зв'язку задаються явно, відносяться до моделей «об'єкт-зв'язок», але розрізняються за способом визначення зв'язків і властивостей

об'єктів. За типами зв'язків виділяють моделі з n-арними зв'язками, тобто зв'язками між більш ніж двома об'єктами, і бінарними зв'язками. За властивостями (атрибутам) - моделі з атрибутами для об'єктів і зв'язків, тільки для об'єктів і без атрибутів.

МД з n-арними зв'язками діляться на два підкласу: з атрибутами для об'єктів і зв'язків і з атрибутами тільки для об'єктів. До першого підкласу відноситься одна з найбільш відомих семантичних моделей «сутність- зв'язок» або ПМ-модель П.Чена [44]. Основні концепції моделі:

- суті, які за однаковими ознаками класифікуються в типи сутностей;
- зв'язку між сутностями;
- атрибути як асоціації між сутностями і наборами значень або між зв'язками і наборами значень;
- безлічі значень (домени).

Підтримувані обмеження цілісності:

- на діапазон значень в домені;
- на діапазон значень властивостей;
- на існуючі значення в БД;
- на зв'язку.

СД має графічне представлення у вигляді ПМ -діаграмм [44, 45] - графа з вершинами трьох типів, відповідних типів сутностей, типами зв'язків і атрибутів.

Існують і інші моделі типу «сутність-зв'язок». EERM (розширені сутність-зв'язок) [47], ECRM (сутність-категорія зв'язок) [46] відрізняються введенням додаткових обмежень і абстракцій в базову - ER-модель. Моделі цього класу застосовуються при проектуванні ІС [42], баз знань [46], в якості концептуальної моделі в СУБД.

Другий підклас моделей з n-арними зв'язками мало відрізняється від ER-моделей за своїми основними застосовуваними концепціями, принципам структурування даних, підтримуваними обмеженнями цілісності. Такий підхід отримав назву EAR (сутність-атрибут-зв'язок) [44]. Єдина його відмінність

полягає в тому, що зв'язки не можуть мати властивостей і, отже, аналогічно атрибутам самі описують деякі властивості сутностей.

Розглянутий клас МД з n-арними зв'язками є одним з найбільш поширених для опису ПО. До його переваг належить невелика кількість основних концепцій, природність уявлення; структурність даних у вигляді сутностей, зв'язків і властивостей; наочність графічної інтерпретації схеми у вигляді ER-діаграм. За своєю семантичною потужності вони не поступаються об'єктним і бінарним МД. До недоліків даного підходу відносять відсутність семантичного релятивізму, властивого об'єктним моделям, тобто можливість представлення одного і того ж факту ПО різними користувачами у вигляді сутності або зв'язку. Однак цей недолік властивий і ряду моделей інших класів: PPM [24], TAXIS, а застосування високорівневих сутностей, побудованих агрегацією над типом зв'язку дозволяє усунути його.

- Серед МД з бінарними зв'язками розрізняються моделі з атрибутами для об'єктів і моделі без атрибутів. Моделі з атрибутами для об'єктів в свою чергу за видами зв'язків діляться на: допускають зв'язку «багато-до-багатьох»: Б НМ (семантична ієрархічна модель), АКМ (модель асоціації на мультіспісках об'єктів), ОСЮМ (об'єктно-орієнтована модель даних) [64];

- допускають лише зв'язку «один-до-одного» і «один-ко-многим» PPM (реляційно-решітчаста модель) [24], 8НМ + [41].

У МД з бінарними зв'язками і атрибутами для об'єктів основних поняттям, аналогічно об'єктним моделям, є клас. Клас - n-арное відношення, відповідне типу об'єкта або зв'язку.

Зв'язки між класами задаються явно і представляють відносини «частина-ціле», «рід-вид», «елемент-група». Моделі мають графічну інтерпретацію у вигляді одного (PPM, 8НМ +, ОСЮМ) або декількох графів (АКМ). У деяких класифікаціях ці моделі відносяться до об'єктним.

Бінарні МД без атрибутів традиційно відносяться до бінарним підходу [35]. Вони з'явилися трохи пізніше реляційних і спираються на поняття бінарного факту, що встановлює зв'язок між двома об'єктами. Властивості об'єктів ПО

розглядаються як самостійні об'єкти. СД в термінах бінарної моделі можна представити у вигляді графа, вершини якого відповідають типам об'єктів ПО, а дуги - фактам. Такий граф є змістовне уявлення структури і називається графом типів. Залежно від конкретної моделі дуги графа можуть бути спрямованими або ненаправленою, поміченими або непоміченими. До моделей даного типу відносяться: 8ВМ (семантична бінарна модель), ЕБМ (модель безлічі сутностей), Мопра (модель опису предметної області), СВ-модель (канонічна бінарна модель).

Моделі з бінарними зв'язками дуже прості. До їх основних недоліків відноситься те, що багато взаємозв'язку в реальному світі ставляться більш ніж до двох об'єктах і для опису таких взаємозв'язків в СД доводиться вводити додаткові конструкції. Результати аналізу семантичних МД зведені в таблиці. У таблиці дано конкретні представники різних класів моделей.

За результатами аналізу видно, що існуючі моделі різних класів мають приблизно однаковими моделируючими можливостями і жодна з них не задовольняє всім поставленим вимогам.

Найбільш розробленою моделлю цього класу за можливостями ЯМД, графічної інтерпретації структурної та динамічної частини СД, наявності правил відображення в реляційні та мережеві МД є ЕСКМ [51]). Для цілей концептуального проектування представляється доцільним розширити можливості ПМ-моделі за описом складних властивостей ПО і розробити формалізоване опис її компонентів.

З точки зору користувача модель даних повинна задовольняти вимогам:

- простоти: модель повинна мати невелике число типів структур з безпосередніми правилами побудови і мінімумом атрибутів, які повинен знати користувач;
- наочності: модель даних повинна бути візуально представимо;
- легкості інформаційного моделювання: модель даних повинна дозволяти безпосереднє моделювання систем реального світу;

- розбиваємості: необхідно, щоб модель даних надавала можливість простий декомпозиції бази даних;
- сумісності: необхідно передбачити можливість використання моделі даних з іншими моделями;
- незалежності: модель даних не повинна містити деталей, пов'язаних з конкретною реалізацією.

Основним недоліком цих критеріїв є їх неформальність. На жаль, наявні спроби дати формальні критерії моделі даних часто носять спрощений характер і не охоплюють всіх сторін поняття.

Таким чином, проведена класифікація існуючих МД, що застосовуються на етапі концептуального проектування і суміжних з ним етапах, і аналіз моделей різних класів на відповідність вимогам, що пред'являються процесом концептуального проектування. На підставі виконаного аналізу обґрунтований вибір класу моделей «об'єкт-зв'язок» з п-арними зв'язками і атрибутами для об'єктів і зв'язків для цілей концептуального проектування. Проектуванню БД присвячена велика кількість теоретичних робіт. Серед них слід виділити перш за все [8, 10, 15]. Зважаючи на специфіку методів розробки і великого обсягу робіт передбачається виділення БД в самостійний об'єкт проектування [10, 12, 13]. В результаті проведеного аналізу різних методів проектування був зроблений висновок про те, що, незважаючи на їх різноманіття,

На першому етапі проектування (концептуальний рівень) проводиться виявлення об'єктів і їх взаємозв'язків, що представляють інтерес в аналізованій предметній області. Проводиться структурування предметної області (ПО) з урахуванням інформаційних інтересів користувачів системи і проводиться побудова концептуальної структури. Вона дає можливість систематизувати інформаційний зміст ПО, дозволяє як би «піднятися вгору» над ПО і побачити її окремі елементи. Концептуальна структура є поданням точки зору користувача на предметну область і не залежить ні від програмного забезпечення системи управління базою даних, ні від технічних рішень.

На відміну від концептуального, логічне моделювання несе в собі порівняно малу семантичне навантаження. У найбільш поширеному випадку (реляційний підхід) логічне проектування зводиться до того, щоб правильно сформулювати об'єкти, їх атрибути і взаємозв'язку з урахуванням методологічних вимог ліквідації надмірності, нормалізації і цілісності.

Під оптимізацією на концептуальному рівні розуміється побудова декількох варіантів концептуальної моделі і вибір проектувальником одного з них. При проектуванні концептуальної моделі всіх зусиль розробника повинні бути спрямовані в основному на структурування даних і виявлення взаємозв'язків між ними без розгляду особливостей реалізації і питань ефективності обробки.

При розробці логічних структур використовуються в основному наближені методи з формальною оцінкою якості прийнятих рішень. На етапі логічного проектування можна виділити наступні критерії ефективності проектування [4]:

- зменшення обсягу пам'яті, необхідного для зберігання всієї інформації;
- зменшення часу пошуку необхідної інформації;
- зменшення сумарного часу первинного завантаження;
- зменшення ймовірності перестроювання набору відносин при введенні нових типів даних;
- звільнення набору відносин від аномалій додавання, зміни і видалення;

На етапі фізичного проектування основними критеріями є:

- зменшення середнього часу доступу до даних бази;
- зменшення витрат на зберігання даних в базі даних;
- зменшення числа оброблюваних сторінок;
- зменшення загального часу роботи з базою даних;
- зменшення часу роботи системи введення-виведення або центрального процесора.

Згідно з проведеним аналізом, можна зробити висновок про великому різноманітті критеріїв якості проекту бази даних.

1. Критерії, що відображають вимога підтримки в структурі бази даних семантичних властивостей і особливостей даних і зв'язків, зафіксованих при аналізі предметної області. Сюди відносяться:

1.1 цілісність даних;

1.2 несуперечливість даних;

1.3 надмірність даних;

1.4 виключення проблеми вставки і видалення даних;

2. Критерії, що характеризують часові параметри доступу і економічні показники ефективним розробкам бази даних:

2.1 число і склад логічних масивів;

2.2 число зв'язків, встановлених між логічними масивами;

2.3 сумарна довжина шляхів доступу до шуканим даними (шлях доступу - послідовність примірників зв'язку, прохідних з метою пошуку необхідних в запитах елементів даних); сумарний час первинного завантаження;

2.4 обсяг пересилаються в оперативну пам'ять даних (число і склад структурних одиниць обміну між системою управління бази даних і робочої областю оперативної пам'яті); число звернень до зовнішньої пам'яті ЕОМ.

3. МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ БАЗ ДАНИХ

3.1. Методика концептуального проектування систем баз даних

На основі проведеного аналізу процесу проектування БД може бути запропонована методика концептуального проектування, орієнтована на створення структури бази даних [6]. Методика передбачає використання наступних моделей: організаційної моделі, моделі інформаційного проходження і КМД. Дано визначення цих моделей.

Організаційна модель являє собою безліч функцій ПО та відносин безпосереднього підпорядкування заданих на множині функцій.

Під функцією розуміється деяка існуюча або передбачувана область діяльності ПО, що має певну мету. Якщо в якості ПО розглядати підприємство, то розподіл функцій буде, швидше за все, відповідати ситуації, що організаційну структуру, наприклад: збут, кадри, фінанси і т.д. Для інших ПО безліч функцій можна визначити методом структурно-функціональної декомпозиції.

В процесі специфікації вимог для кожної функції дається її опис. Для неї визначається схема даних в термінах використовуваної КМД, безліч вхідних / вихідних документів, частота виконання, час виконання, кількість виконавців і інші характеристики.

Модель інформаційного прямування представляє орієнтований граф, де вершин графа відповідають функції, а ребра графа показують інформаційну взаємозв'язок функцій. На ребрі вказуються імена елементів даних або документів, які передаються від однієї функції до іншої, а напрямок ребра визначає напрямок їх слідування.

Аналогічні організаційні моделі і графи інформаційного проходження застосовуються в методиках BSP, Про АМ, OARSM. Вибір КМД, як вже було зазначено, вимагає спеціального аналізу і обґрунтування і залежить від вимог методики проектування.

Формалізована методика концептуального проектування системи БД включає наступні кроки.

Крок 1. Визначити організаційну схему ПО і занести в словник-довідник дані про функції ПО.

Крок 2. Перевірити організаційну схему: якщо визначена концептуальна структура даних (КСД), то перейти до кроку 3, інакше повторити крок 1.

Крок 3. Побудувати граф інформаційного проходження.

Крок 4. Якщо в побудованому графі інформаційного проходження немає протиріч і він повністю описує всю предметну область, використовуючи представлені джерела і споживачі інформації то перейти до кроку 5, інакше - до кроку 1 (уточнити опис даних).

Крок 5. Сформувати безліч відносин еквівалентності КСД.

Крок 6. Побудувати КСД.

Крок 7. Визначити структуру системи БД (СБД), що складається, з безлічі предметних баз даних (ПБД).

Крок 8. Виконати інтеграцію КСД додатків за допомогою застосовної КМД.

Крок 9. Дати оцінки очікуваної ефективності і часу розробки ПБД.

Крок 10. Визначити послідовність розробки кожної ПБД.

Пояснимо основні кроки методики.

Перший крок відноситься до етапу специфікації вимог до системи. В результаті побудови організаційної схеми визначається коло можливих користувачів, безліч вирішуваних завдань, документується опис даних.

На кроках 2-4 здійснюється аналіз вимог на повноту і несуперечність:

- перевіряється наявність опису всіх функцій, завдань і їх даних (крок 2);
- встановлюється проходження даних в системі (крок 3);
- визначаються джерела і споживачі інформації (крок 4).

Перераховані кроки представляють безліч ітерацій, на кожній з яких уточнюється опис вимог. Після того, як виконано і уточнено опис всіх завдань

ПО і в організаційній схемі і схемі інформаційного прямування не виявлено протиріч, формується структура СБД (кроки 5 - 7).

Кроки 5-7 відповідають методиці концептуального проектування, запропонованої в [54]. На кроці 5 формуються безлічі відносин еквівалентності КСД, що дозволяє проаналізувати елементи різних додатків. На основі результатів кроку 5 на кроці 6 підраховуються оцінки КСД додатків і, використовуючи їх, з безлічі додатків формується СБД, що складається з непересічних ПБД (крок 7).

На кроці 8 здійснюється інтеграція КСД додатків однієї ПБД.

Кроки 9-10 відповідають планування розробки СБД. На кроці 9 експерти оцінюють складність проектування і ефективність ПБД, на основі яких на кроці 10 виконується призначення пріоритетів ПБД на розробку і впровадження.

Запропонована методика, заснована на існуючих методиках [49, 54], визначає послідовність дій при концептуальному проектуванні БД складної структури і великої розмірності і, на відміну від існуючих методик, в явному вигляді включає виконання завдань даного етапу проектування.

Методика використовує комбінований підхід до проектування БД, який передбачає ітеративне виконання процедур специфікації і аналізу вимог (кроки 1 - 4) та послідовну розробку ПБД, що входять в систему. Застосування описаних організаційної моделі та моделі інформаційного проходження дозволяє вже на перших кроках проектування виявити повноту і неточність описів, що значно скорочує кількість ітерацій при специфікації і аналізі вимог. Формування структури СБД, відповідно до методики, дозволяє знизити розмірність завдань інтеграції КСД завдань і додатків в КСД ПБД і подальшої розробки СБД.

На підставі обраного підходаразроботана методика концептуального проектування систем баз даних, що передбачає взаємозв'язок і рішення сформульованих завдань концептуального проектування СБД.

3.2. Побудова організаційної моделі бази даних

Організаційна модель БД включає в себе формалізований опис предметних областей і формування моделей специфікацій інформаційних вимог користувачів.

Під предметною областю користувачів будемо розуміти інформацію про сукупність об'єктів автоматизації та їх характеристиках, яка подається у вигляді спеціальних структур даних і використовується користувачами для вирішення різних функціональних завдань.

Опис предметної області користувачів включає наступні основні компоненти: автоматизує функції і завдання (процедури) обробки даних і їх характеристики, користувачі, інформаційні елементи і відносини між ними, характеристики інформаційних елементів і процедур обробки даних, відносини між інформаційними елементами і процедурами.

Дана модель предметної області використовується при формуванні моделей специфікацій інформаційних вимог користувачів.

Побудова моделі специфікації здійснюється для кожного користувача безлічі P шляхом послідовно аналізу матриць суміжності, що описують відносини (взаємозв'язку) між компонентами предметної області. Алгоритм рішення задач формування специфікацій інформаційних вимог користувачів складається з наступних кроків: на підставі аналізу матриць PP і HP визначається перелік функцій і завдань по кожному користувачеві; з використанням матриць PB та PZ формуються пари структурних елементів $\langle \text{об'єктів} \rangle$ де - відносини між об'єктами автоматизації.

3.3. Побудова моделі інформаційного проходження бази даних

Побудова моделі інформаційного проходження включає в себе аналіз інформаційних вимог користувачів і формування графів інформаційних структур.

Нехай безліч інформаційних вимог користувачів, $D = \{d_1, d_2, \dots, d_{10}\}$ - безліч структурних елементів входять до складу k -го вимоги, I - безліч структурних елементів предметної області.

Під матрицею семантичної суміжності $B_k = \|b_{ij}\|$ інформаційної вимоги S_k будемо розуміти квадратну бінарну матрицю, проіндексовану по обох осях безліччю структурних елементів I_k і містить запис $b_{ij} = 1$, якщо на підставі інформації користувачів про семантичної зв'язності елементів k -го вимоги, між структурними елементами d_i і d_j існує відношення R_k таке, що елемент d_i становить (розширює, доповнює і т.д.) смислове зміст елемента d_j і $b_{ij} = 0$ в іншому випадку

i/j	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
d_1	0	1	0	0	1	0	1	0	1	0
d_2	0	0	0	0	0	1	0	0	0	1
d_3	0	0	0	1	0	0	0	1	1	0
d_4	0	0	0	0	0	0	0	0	0	0
d_5	0	0	0	0	0	0	0	0	0	0
d_6	0	0	1	0	0	0	0	0	1	0
d_7	0	0	0	0	0	0	0	0	0	0
d_8	0	0	0	0	0	0	0	0	0	0
d_9	0	0	0	0	0	0	0	0	0	0
d_{10}	0	0	0	0	0	0	0	0	0	0

Рис.3.1. Матриця суміжності B_k

Матриці B_k ставиться у відповідність орграф інформаційної структури до-го вимоги S_k (O_k, I_k) (Рис.3.2), безліччю вершин якого є структурні елементи безлічі I_k , а ребро (d_i, d_j) відповідає записи $b_{ij} = 1$, в матриці B_k . Таким чином, ребра орграфа S_k відображають наявність або відсутність семантичної зв'язності між структурними елементами інформаційного вимоги B_k

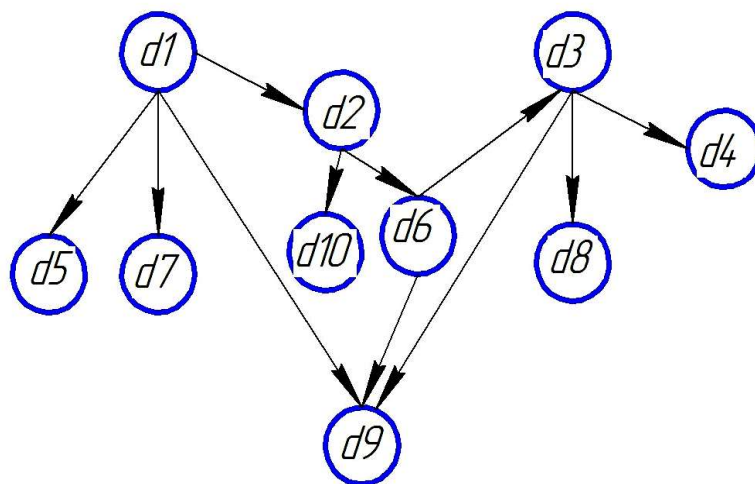


Рис.3.2. Орграф інформаційної структури K-ї вимоги

Для виявлення взаємозв'язків між структурними елементами, виділення груп інформаційних вимог і визначення їх складу з використанням матриці В к формується матриця семантичної досяжності $A_k = NI$ [25].

Під матрицею досяжності A_k (рис.3.3) будемо розуміти квадратну бінарну матрицю, проіндексовану однакоовим чином по обох осях безліччю структурних елементів $D_k = \{d, / l \in L_k \text{ з } L\}$. Запис $a_{ij} = 1$ матриці A_k відповідає наявності смислового ставлення досяжності R_0 елемента d_j з елемента d_i $d_i R_0 d_j$. При цьому вважається, що елемент d_j семантично досяжний з елемента d_i , якщо на графі G існує шлях від вершини d_i до вершини d_j , який має певне смислове зміст.

Передбачається, що відношення досяжності задовольняє умові транзитивності якщо $d_i R_0 d_n$ и $d_n R_0 d_j$, то $d_i R_0 d_j$, $i, j, n = 1, 2, \dots, L_k$. Матриця A_k дає можливість визначення множин передування $S(e;)$ і досяжності $P(e;)$ $U_s // e \in I_k$. Безліч $S(e;)$ формується з елементів, відповідних одиничних записів в i -му рядку матриці A_k . Аналіз множин $S(e;)$ дозволяє виділити базові типіструктурних елементів, з яких конструюється k -я інформаційна структура - інформаційні елементи і групи. Інформаційним елементам відповідає ті елементи структури, для яких $S(e;) = 0$. На графі S_k (рис.3.4) їм відповідають

висячі вершини. Безліч інформаційних елементів позначимо через I_k . Безліч груп I_k визначається з виразу: $\mathcal{I} > * = I_k \setminus I_k$.

Для розглянутого прикладу маємо:

$$C(d_1)=[d_2,d_3,d_5,d_6,d_7,d_9,d_{10}], \quad C(d_2)=[d_3,d_4,d_6,d_9,d_{10}], \quad C(d_3)=[d_4,d_8,d_9], \\ C(d_4)=0, \quad C(d_5)=0, \quad C(d_6)=[d_3,d_9], \quad C(d_7)=0, \quad C(d_8)=0, \quad C(d_9)=0, \quad C(d_{10})=0.$$

$$F(d_1)=0, \quad F(d_2)=[d_1], \quad F(d_3)=[d_1,d_2,d_6], \quad F(d_4)=[d_2,d_3], \quad F(d_5)=[d_1], \quad F(d_6)=[d_1,d_2], \\ F(d_7)=[d_1], \quad F(d_8)=[d_3], \quad F(d_9)=[d_1,d_2,d_3,d_6], \quad F(d_{10})=[d_1,d_2].$$

$$D_k^e = [d_1, d_2, d_3, d_6].$$

$$D_k^o = [d_4, d_5, d_7, d_8, d_9, d_{10}].$$

<i>i/j</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d6</i>
<i>d1</i>	1	1	1	1
<i>d2</i>	0	1	1	1
<i>d3</i>	0	0	1	0
<i>d6</i>	0	0	1	1

Рис.3.3. Підматриця досяжності A_k

На підставі цього визначення будь-які дві групи одного і того ж рівня або не пов'язані один з одним, або є двосторонні зв'язки (цикли) між цими елементами. Крім того, умова $\Gamma (\wedge) =$ забезпечує те, що всі зв'язки з групи d_i до інших груп знаходяться на тому ж рівні, що і d_i , в той час як всі зв'язки від інших груп до d_i знаходяться або на тому ж самому рівні, або на більш низькому рівні. Дане визначення дозволяє з використанням матриці досяжності A_k розділити безліч груп D_k на підмножина відповідно до рівнів $r_{t>} t = 1, M_0$ їх розташування.

Група I_k належить множині груп верхнього рівня

$$F(d_i^e) \cap C(d_i^e) = F(d_i^e).$$

Належність інших груп рівнями ієрархії $p_t, t > 2$ визначається ітеративним чином із співвідношення:

$$p_m = \{d_i^z \in D_k^z \setminus p_1 \setminus \dots \setminus p_{m-1} / F_{m-1}(d_i^z) \cap C_{m-1}(d_i^z) = F_{m-1}(d_i^z)\},$$

де $F_{m-1}(d_i^z)$ и $C_{m-1}(d_i^z)$ - відповідно, множини досяжності і передування групи $d_i^z \in D_k^z$ на підмножині $\supseteq D_k^z \setminus p_1 \setminus \dots \setminus p_{m-1}$.

Впорядкування груп дозволяє виділити групи, які є кореневими групами структури і групи, що займають проміжне положення. Кореневі групи визначають можливі точки входу в інформаційну структуру, а проміжні розширюють відомості про інформацію, вміщеній в корневих і верхніх групах.

Подальший аналіз доцільно проводити для кожної групи інформаційної структури користувачів. Для визначення інформаційного складу груп $H \{d^*\}$ необхідно видалити з вихідної матриці суміжності B_k записи $b_{ij}^k = 1$ для індексів i і j елементів d_i^z, d_j^z , які увійшли в безліч D_k , що забезпечує розрив зв'язку між групами інформаційних елементів і входження в групу інформаційних елементів матриці B_k , для яких записи D_k для елементів матриці B_k , для яких записи $b_{ij}^k = 1, \forall d_i \in D_k^z$ для відповідного стовбця $d_j^z \in D_k^z$, т.е. $H(d_j^z) = \{d_i / b_{ij}^k, \forall d_i \in D_k^z\}$

3.4. Методи нормалізації інформаційних структур користувачів

Під нормалізацією інформаційних структур користувачів будемо розуміти процес приведення їх до виду, що забезпечує мінімальну надмірність і дубльованих даних і зв'язків, а також специфікацію типів інформаційних елементів груп даних (ключів і атрибутів).

Виділені на попередньому етапі групи $\langle Z^* \rangle$ е ОГК зазвичай містять досить велику кількість повторюваних (дубльованих) інформаційних елементів. Допустимість виключення дубльованого елемента визначається в результаті

аналізу шляхів доступу між групами, в яких він з'являється, на підставі матриці семантичної досяжності груп $L^* = \| a^{(i,j)}$.

Дубльовані елементи виключаються з усіх груп, крім однієї, в тому випадку, якщо розглянуті групи пов'язані одним з можливих шляхів доступу.

Виняток дубльованих елементів може бути здійснено в будь-якій з виділених пов'язаних груп. Якщо немає обмежень на вибір групи, то дубльовані елементи виключаються з груп нижчих рівнів, а єдиний елемент залишається в групі, що має вищий рівень. Така стратегія забезпечує зниження часу пошуку даного елемента з кореневої групи при відповідях на запити користувачів.

Дубльованими елементами є елементи й4, А9 і С110. Інформаційний елемент А4 використовується в пов'язаних групах і інформаційний елемент А9 в пов'язаних групах * /,% з12, і </ б% а інформаційний елемент с / 10 використовується в пов'язаних групах і с12.

При виключення дубльованих елементів в групах на графі Ок (Ок, ик) виключаються взаємозв'язку, що ведуть у групи, з яких вийшов розглянутий елемент.

Крім існування дубльованих елементів у вихідній інформаційній структурі користувача в ній можлива наявність надлишкових взаємозв'язків між групами. Надлишкова взаємозв'язок між парою груп існує в тому і тільки тому випадку, якщо є ребро (/, у) з'єднує групи (1 і </ *, і шлях, що проходить через деякий безліч інших груп. Дуга (/, у) є надмірною і може бути видалена з нашої інформаційної структури.

Виділені на попередньому етапі групи <3 * е ОГК зазвичай містять досить велику кількість повторюваних (дубльованих) інформаційних елементів.

З метою забезпечення мінімальної надмірності збережених даних потрібно виявити безліч дубльованих елементів в аналізованих структурах.

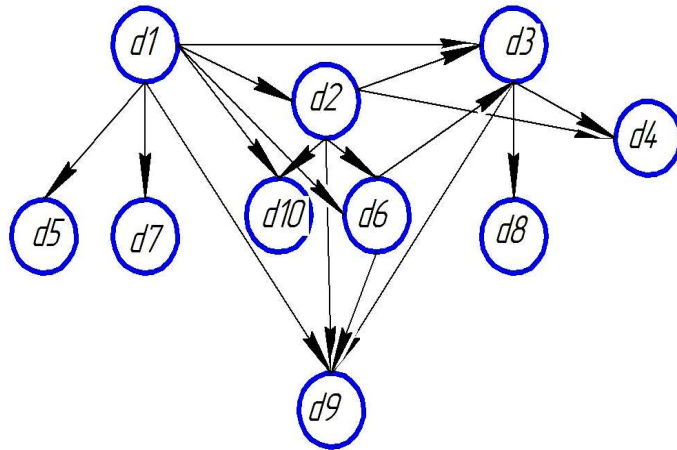


Рис.3.4. Перетворений оргграф інформаційної структури

<i>i/j</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>	<i>d5</i>	<i>d6</i>	<i>d7</i>	<i>d8</i>	<i>d9</i>	<i>d10</i>
<i>d1</i>	0	1	1	0	1	1	1	0	1	1
<i>d2</i>	0	0	1	1	0	1	0	0	1	1
<i>d3</i>	0	0	0	1	0	0	0	1	1	0
<i>d4</i>	0	0	0	0	0	0	0	0	0	0
<i>d5</i>	0	0	0	0	0	0	0	0	0	0
<i>d6</i>	0	0	1	0	0	0	0	0	1	0
<i>d7</i>	0	0	0	0	0	0	0	0	0	0
<i>d8</i>	0	0	0	0	0	0	0	0	0	0
<i>d9</i>	0	0	0	0	0	0	0	0	0	0
<i>d10</i>	0	0	0	0	0	0	0	0	0	0

Рис.3.5. Матриця досяжності A_k

Допустимість виключення дубльованого елемента визначається в результаті аналізу шляхів доступу між групами, в яких він з'являється, на підставі матриці семантичної досяжності груп $L^* = || a^{ij} ||$. Дубльовані елементи виключаються з усіх груп, крім однієї, в тому випадку, якщо розглянуті групи пов'язані одним з можливих шляхів доступу. Цей випадок

відповідає наявності в матриці Ак елементів $a^* = \dots = a^0 = \dots = a^*$, для $\langle / ;, A \rangle$,
 .. Agr, C / ;, Agп, що лежать на одній колії доступу.

Належність інших груп рівням ієрархії, $p_m, m \geq 2$
 визначається ітеративним чином із співвідношення:

$$p_m = \{d_i^z \in D_k^z \setminus p_1 \setminus \dots \setminus p_{m-1} / F_{m-1}(d_i^z) \cap C_{m-1}(d_i^z) = F_{m-1}(d_i^z)\},$$

де $F_{m-1}(d_i^z)$ $C_{m-1}(d_i^z)$ - відповідно, безліч досяжності і передування груп на підмножині
 $D_k^z \setminus p_1 \setminus \dots \setminus p_{m-1}$.

Виняток дубльованих елементів може бути здійснено в будь-який з виділених пов'язаних груп. Якщо немає обмежень на вибір групи, то дубльовані елементи виключаються з груп нижчих рівнів, а єдиний елемент залишається в групі, що має вищий рівень. Така стратегія забезпечує зниження часу пошуку даного елемента з кореневої групи при відповідях на запити користувачів. При виключення дубльованих елементів в групах на графі Ок (Ок, ик) виключаються взаємозв'язку, що ведуть у групи, з яких вийшов розглянутий елемент.

Кожна з отриманих матриць ВГК (X) порівнюється з вихідною матрицею ВГК. У тому випадку, коли в матриці Вк є елемент $b^{\wedge} = 1$, а в матриці ВГК {X} відповідний елемент є надмірним.

В результаті використання розглянутих вище процедур упорядкування та виключення дубльованих елементів і надлишкових взаємозв'язків формується структурована матриця суміжності В к і відповідної їй оргграф к-й інформаційної структури Сск (Пк, ик). Матриця Вск (рис.3.9) і граф Ск не містять дубльованих елементів в групах даних і надлишкових взаємозв'язків між групами.

В роботі передбачаються два режим виділення ключів і атрибутів груп: автоматичний, який використовує формальне поняття ключа групи і автоматизований, що враховує інформацію про типах взаємозв'язків між інформаційними елементами груп, яка задається розробником БД в режимі діалогу з ЕОМ. Вихідними даними для автоматичного виділення можливих

ключів і залежних від них атрибутів є безлічі областей визначення інформаційних елементів груп і послідовність появи конкретних значень інформаційних елементів у відповідних кортежі.

При аналізі інформаційних структур необхідно розрізняти два типи ключів: основний ключ групи і допоміжний.

Основним ключем групи d_j є інформаційний елемент $d_j \in H(d_j)$, повністю і однозначно ідентифікує групу і її елементи, тобто елемент, для якого виконується така умова:

$$d_i^1 \neq d_i^2 \neq \dots \neq d_j^\mu \neq d_j^{\mu_0},$$

де μ_0 – максимальна потужність множин областей визначення інформаційних елементів групи d_j^2 .

Допоміжним ключем є інформаційний елемент, що належить множині атрибутів первинного ключа і однозначно визначає деяку підмножину безлічі атрибутів. Для допоміжного ключа виконується наведене вище умова на підмножині атрибутів групи.

Для визначення основних і допоміжних ключів груп в автоматичному режимі пропонується наступна послідовність операцій:

- 1) Визначення по сволока V_k інформаційного складу груп $H(d_j)$, $d_j \in D_k$.
- 2) Завдання областей визначення інформаційних елементів ПРО, і визначення порядку їх слідування в межах груп.
- 3) Формування множин кортежів R_{JM} ($j \leq L * k$) для кожної групи окремо.
- 4) Визначення можливих основних ключів груп і атрибутів (шляхом послідовного аналізу інформаційних елементів групи на виконання умови проіндексованих різними значеннями).
- 5) Формування матриць суміжності значень атрибутів кожної групи - $M_{JK} =$ атрибутів груп d_j , $d_r \in H(d_j)$. У матриці M (фіксується запис $m_{kjr} d_j = 1$, якщо значення d_f і d_{vr} спільно присутні в одному з кортежів Π і $m_{jr} \gamma \sim 0$) в іншому випадку.

Визначимо інформаційний склад груп для даного прикладу:

$$H(d_1^2) = \{d_5, d_7, d_9\}, H(d_2^2) = \{d_6, d_{10}\}, H(d_3^2) = \{d_4, d_8\}.$$

Задамо область визначення інформаційних елементів (залізничний транспорт) і сформуємо безліч кортежів для кожної групи:

$$H(d_1^2) = \{d_5, d_7, d_9\}$$

d_1^2 - залізниця;

d_5 - код залізниці (1,17,29);

d_7 - найменування залізниці;

d_9 - експлуатаційна довжина (100000, 120000, 140000); $H(d_2^2) = \{d_6\}$

d_2^2 - відділення залізниць;

d_6 - код відділення залізниць;

d_{10} - найменування відділення залізниці;

d_3^2 - станції;

d_4 - найменування станції;

d_8 - кількість стрілочних переводів;

Визначимо можливі ключі груп:

- для групи d1 ключ d5;
- для групи d2 ключ d6;
- для групи d3 ключ d4;

Перевірка для кожної пари елементів матриці μ умови

$$\sum_{v=1}^{v_r} m_{d_i^{\mu}, d_j^{\nu}}^{kj} = 1, \mu = 1, \mu_l,$$

Виконання даної умови за всіма різним значенням деякого інформаційного елемента d , на підмножині атрибутів $\{dr\}$ означає, що елемент d , але відіграє допоміжну ключем групи для підмножини її атрибутів $\{dr\}$. Аналіз матриці МЖк дозволяє також ідентифікувати типи взаємозв'язків між інформаційними

елементами кожної групи. Простому типу взаємозв'язків відповідає виконання умови, складної – умови

$$\exists d_i^u : \sum_{v=1}^{v_r} m_{d_i^u, d_r^v}^{kj} > 1, \forall d_r \in H(d_j^z)$$

Аналіз матриці TkJ дозволяє зробити наступні висновки: - основним ключем групи d 'є інформаційний елемент d

$$d_i \in H(d_j^z), \text{ для якого } t_{ir} = 1, \forall d_r \in H(d_j^z);$$

Якщо в результаті аналізу груп немає можливості виділення основного ключа групи, то інформація про це видається користувачеві, який формує зчеплений або унікальний ключ.

Після прийняття розробником остаточного рішення щодо складу ключів (основних і допоміжних) і атрибутів здійснюється перетворення структурованої матриці Вк і графа ССК (йк, ик) до виду, зручного для подальшої побудови концептуальної структури БД. Перетворення полягає в наступному:

1. З графа ЄК видаляються вершини, що відповідають групам елементів даних, і зв'язку, провідні в них з відповідних елементів даних.
2. На графі ЄК встановлюються зв'язки, що ведуть з основного і допоміжних ключів групи в атрибути, безпосередньо залежні від них.
3. Фіксуються зв'язку між допоміжними і основними ключами групи.
4. Все зв'язку між групами на початковому графі Ск (Ек, ик) замінюються на зв'язку (з тим же напрямком) між ключами груп.

Матриця Вк перетворюється в наведену матрицю суміжності В *, проіндексовану по рядках і стовпцях елементами безлічі ключів і атрибутів.

Описані вище процедури виділення ключів і атрибутів виконуються для кожної групи інформаційних структур користувачів, в результаті чого формується безліч наведених матриць суміжності {5 *} і відповідних їм оргграфів.

3.5.Метод побудови концептуальної структури бази дани

Вихідною інформацією, необхідної для побудови концептуальної структури БД, є наведені матриці суміжності і відповідні їм орграфів $\langle \Gamma, \ast \rangle$, отримані на попередньому етапі.

Формалізовані процедури, розроблені стосовно до етапу побудови концептуальної Структура БД, орієнтовані на вирішення наступних завдань: формування беззбиточного безлічі ключів і атрибутів \mathcal{K}_2 і повного беззбиточного безлічі інформаційних елементів \mathcal{I} ; побудова узагальненої матриці суміжності $B_0 = \| b_{ij} \rangle$ і узагальненого графа концептуальної структури БД $O_0 (O_0, i_0)$; виявлення та аналіз пересічних атрибутів і надлишкових взаємозв'язків, що забезпечує побудову результуючої матриці суміжності концептуальної структури БД B і відповідного орграфа \mathcal{C} .

Формується беззбиточное безліч ключів структури. Спочатку знаходяться підмножини попарних перетинів множин для всіх $k = 1, K_0$. Потім вибирається множеству воно доповнюється елементами, що не ввійшли в нього, утворюють безліч W_1^0

$$W_1^0 = W_1^1 \cup W_1^2 \setminus W_1^{1,2} \cup W_1^3 \setminus (W_1^{1,3} \cup W_1^{2,3}) \cup \dots \cup W_1^{K_0} \setminus \bigcup_{j=1}^{K_0-1} W_1^{j,K_0} .$$

Формується безліч атрибутів структури W . Процедура формування аналогічна процедурі формування беззбиточного безлічі ключів і виконується на множинах

$$W_2^k (k = \overline{1, K_0}) .$$

Визначається підмножина перетину.

$$W_{1,2}^0 = W_1^0 \cap W_2^0 .$$

Якщо воно порожнє, то перехід до наступного етапу. В іншому випадку елементи підмножини W виключаються з безлічі W^2 .

1. Формується результуюча матриця суміжності

$$B_0 = \left\| b_{ij}^0 \right\|$$

2. шляхом індексації її по рядках елементами множини, за стовпцями - елементами множин P_1 і W_2 і перенесення одиничних записів матриць B_k в відповідні позиції матриці B_0 . Матриці B_0 ставиться у відповідність оргграф результуючої структури даних G_0 .
3. З матриці B_0 виділяються дві підматриці B_1 і B_2 шляхом індексації їх по рядках елементами множин W^* і $(W_1 \cup W_2)$.
4. Підматриця B_1 послідовно зводиться в ступеня L , $L = 2, 3, \dots$, аналогічно процедурі, описаній в п.3.3, утворюючи безліч матриць B^L, B_1^L
5. для елементів $b_{ij}^0 = 1$ підматриці B_1 і елементів $b_{ij}^0 = 0$ матриці B_2 , здійснюється процедура, описана в п.3.3. В результаті формується список надлишкових взаємозв'язків результуючої структури даних. Цей список проглядається проектувальником БД, який приймає рішення про виключення тих чи інших зв'язків, які увійшли в список. При видаленні зв'язків $(d_i \wedge d_j)$ в матриці B_0 проводиться заміна записів $b_{ij}^0 = 1$ на $b_{ij}^0 = 0$.
6. Для елементів $d_j \in W_2$ (атрибутів) визначаються значення шляхом фіксації j -го стовпця підматриці B_2 і підсумовування по всіх рядках $i \in I$ сволюка (i - індекси ключів структури).

$$b_{ij}^0 = 1 \text{ на } b_{ij}^0 = 0.$$

7. Елементи d_j , для яких ключі, від яких вони залежні, заносяться в список пересічних атрибутів.

На підставі підматриць B_1 і B_2 будується матриця семантичної досяжності $\Delta = \|\Delta_{ij}\|$, аналогічно процедурі, описаній в п. Позиції уздовж головної діагоналі матриці Δ , заповнюються одиничними записами. Для елементів матриці Δ , формуються безлічі передування $S(\Delta)$ і досяжності $F(\Delta)$.

Проводиться впорядкування ключів предметної області за рівнями ієрархії L , аналогічно процедурі описаної в п.3.3, і здійснюваної на безлічі елементів $1 / Y^*$, починаючи з найвищого ($p = 1$) рівня. Формується упорядкований граф структури і переупорядочення матриця досяжності в блок-діагональній формі D

Для кожного пересічного атрибута зберігається залежність від ключа, розташованого на найвищому рівні ієрархії і усувається залежність від ключів нижчих рівнів. Крім того, встановлюються зв'язки, що ведуть від відповідних ключів нижчих рівнів до вибраного найвищого рівня. У подматриці $B02$ поодинокі записи, відповідні залежності розглянутого атрибута від ключів нижчих рівнів, замінюються на нульові. У подматриці $B01$ на перетині стік, відповідними ключами нижчих рівнів, і шпальти для ключа вищого рівня ієрархії фіксуються поодинокі записи.

Проводиться об'єднання перетворених підматриць $B01$ і $B02$, в результаті якого формується матриця суміжності концептуальної структури БД $B = \| b_{ij} \|$. Матриці Y ставиться у відповідність оргграф концептуальної структури C (j, i), який не містить надлишкових взаємозв'язків і пересічних атрибутів.

3.6. Методи проектування логічних структур баз даних

На даному етапі на основі даних про концептуальні структури БД формується логічна структура БД, яка повинна відповідати таким вимогам:

- забезпечувати збереження семантичних властивостей інформаційних елементів предметної області та зв'язків між ними, зафіксованих в концептуальній структурі БД;
- забезпечити зручність і простоту формування запитів до БД;
- бути оптимальною (раціональною) по заданому критерію ефективності.

Синтез логічних структур БД розглядається як процес пошуку оптимальних варіантів відображення концептуальних структур в такі логічні

структури, які забезпечують оптимальне значення заданого критерію ефективності їх розробки або використання і задовольняють основним вимогам і обмеженням, що накладається на логічну структуру.

При синтезі логічних структур потрібно визначити відображення:

$$G(D,U) \xrightarrow{\theta'} G(N,L),$$

де $G(D,U)$ - граф концептуальної структури БД

$(D = \{D_i / i = \overline{1, I}\}$ - безліч інформаційних елементів предметної області,

$U = \{(d_i, d_{i'}) / i, i' = \overline{1, I}\}$ - безліч взаємозв'язків між елементами);

$G(N,L)$ - граф логічної структури БД

$(N = \{n_j / j = \overline{1, J}\}$ - безліч логічних записів,

$L = \{(n_j, n_{j'}) / j, j' = \overline{1, J}\}$ - безліч взаємозв'язків між записами) . Відображення

передбачає об'єднання елемента безлічі $D = \{D_i / i = \overline{1, I}\}$ в елементи множини $N = \{n_j / j = \overline{1, J}\}$ і безлічі зв'язків $U = \{(d_i, d_{i'}) / i, i' = \overline{1, I}\}$, безліч зв'язків $L = \{(n_j, n_{j'}) / j, j' = \overline{1, J}\}$; при якому досягається оптимальне значення деякої цільової функції Π на безлічі альтернативних варіантів відображення в [25].

Моделі синтезу оптимальних логічних структур БД розроблені з використанням критеріїв мінімуму часу завантаження інформації в БД, мінімуму часу обслуговування заданої множини запитів користувача і мінімуму сумарного числа зв'язків між записами.

Вихідними даними для завдань синтезу оптимальних логічних структур БД є параметри концептуальних структур БД, представлених в агрегованих формах, а також характеристики вимог по обробці, формалізуються шляхом завдання параметрів безлічі запитів до БД і коригувань БД.

Результатом вирішення завдань синтезу оптимальних логічних структур БД є визначення числа і складу логічних записів, вибір структури зв'язків між записами, визначення записів, які обирають в якості точок входу в структуру.

3.6.1.Формалізований опис вихідних даних

Введемо ряд визначень, необхідних для формалізації і опису основних положень. Під типом логічного запису будемо розуміти взаємопов'язану сукупність груп, об'єднаних загальним ім'ям (ключем) і розглядаються як єдине семантично закінчене ціле.

Під примірником записи розуміється конкретна реалізація деякої записи, що є одиницею обміну із зовнішньою пам'яттю. Кожен запис складається з певної кількості примірників.

Структурою записи є певним чином упорядкована сукупність вхідних в неї груп даних.

Примірник зв'язку - конкретна реалізація деякої зв'язку, встановленої між відповідними екземплярами записів.

Зв'язки концептуальної структури БД є відображенням відносин «один до одного» і «один до багатьох».

Структура зв'язків - певним чином організоване безліч зв'язків між записами.

Коефіцієнт зв'язку - максимальна кількість примірників записів, підлеглих у відносинах «один до одного» або «один до багатьох» одному примірнику основної групи, що є «власником» відносин.

Точка входу в логічну структуру БД - тип запису, до якої забезпечується прямий доступ з боку програмних засобів. Шлях доступу - послідовність примірників зв'язків, прохідних з метою пошуку необхідних в запитах примірників записів. Кожен шлях доступу починається в одній з точок входу в логічну структуру БД і закінчується в вершинах шуканих записів БД. На етапі синтезу логічної структури БД граф концептуальної структури \mathcal{L} (\mathcal{L} >, \mathcal{L} /) з метою зменшення його розмірності перетворюється в агрегированную форму \mathcal{L} (C , $\{Y\}$), де кожна вершина (група) C , $\leq C$ отримана в результаті об'єднання ключа 1-ї групи і залежних від нього атрибутів (число груп визначається

безліччю ключів концептуальної структури БД), а ребра між вершинами представляють зв'язку між виділеними групами. Саме такий вид графа ас, іх) буде в подальшому використовуватися в задачах синтезу логічної структури БД. Граф С (С, іх) відображає не тільки різні види взаємозв'язків між групами концептуальної структури БД, але і можливі типи відносин між ними. В цьому випадку група, з якої виходить ребро, є головною в цьому відношенні, тобто його «власником», а група, в яку заходить ребро - підлеглої. Таким чином, граф С (С, іх) (рис.3.6) описується безліччю груп даних $S = \{S, / 1 = 1,1\}$ і матрицею взаємозв'язків між групами $IV = | i '(, . |$ (Рис.3.7), де $m >, \gamma = 1$, якщо між 1-й і γ -й групами встановлено взаємозв'язок.

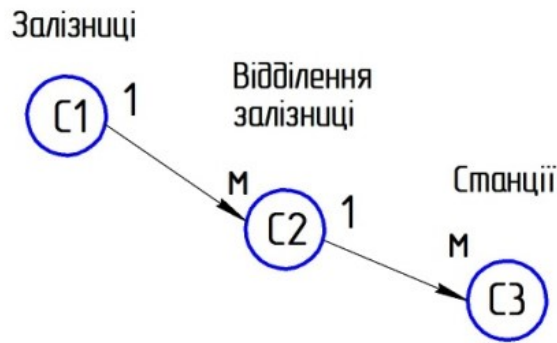


Рис.3.6. Орграф G (C,U1)

	C1	C2	C3
C1	0	1	0
C2	0	0	1
C3	0	0	0

Рис.3.7. Матриця взаємозв'язків між групами W

$R = \{r_1, r_2, \dots, r_i, \dots, r_l\}$ - вектор кількості примірників груп, де γ - кількість примірників i -ої групи

$\rho = \{\rho_1, \rho_2, \dots, \rho_i, \dots, \rho_I\}$ - вектор характеристик довжини кожної групи в бт, де

$\rho_i = \sum_{v \in C_i} \rho_i^v$. Довжина і-ї групи, що дорівнює сумарній довжині елементів $u \in C$, (Ключів і атрибутів), що входять в і-ї групи;

$\Delta = \{\delta_{ii'} / i, i' = \overline{1, I}\}$ - безліч усереднених коефіцієнтів зв'язку, де

$\delta_{ii'} = \left[\frac{r_{i'}}{r_i} \right]$, если $w_{ii'} = 1$; $\delta_{ii'} = 1$, если $w_{i,i'} = 0$, но $w_{i,i} = 1$; $\delta_{ii'} = 0$, если $w_{ii'} = w_{i,i} = 0$.

Усереднений коефіцієнт зв'язку $\delta_{ii'}$ при $i = i' = 1$ визначається наступним чином. Взаємозв'язок (i, i') , встановлена між групами C_i і $C_{i'}$, отражає взаїмоотношенія між екземплярами γ_i і $\gamma_{i'}$ даних груп. Число примірників зв'язку (i, i') визначається числом примірників групи $C_{i'}$. У загальному випадку це число може бути менше ніж так як 1-й примірник $(i = 1)$ 1-ї групи може бути не пов'язаний ні з одним екземпляром i' -групи. Якщо $\delta_{ii'}$ - число примірників i' -групи, пов'язаних з 1-м примірником 1-ї групи і

$\delta_{ii'}^l$ - коефіцієнт зв'язку 1-го екземпляра (ii') -й зв'язку, то $\delta_{ii'} = \left[\frac{\sum_{l=1}^{r_i} \delta_{ii'}^l}{r_i} \right]$, где

$$\sum_{l=1}^{r_i} \delta_{ii'}^l = r_{i'}$$

- Так як кожен екземпляр підпорядкованої групи $C_{i'}$, не може бути пов'язаний з різними екземплярами головної групи, що є «власником» даного відносини. Коефіцієнт зв'язку $\delta_{ii'} = 1$ при $m > 1$, $= 0$, але $m > n = 1$ визначає наявність відносини типу «один до одного» між головною групою C_i і підпорядкованої групою $C_{i'}$.

Кожен екземпляр зв'язку між групами концептуальної структури БД представляється у вигляді зв'язного списку, в якому елементами списку є екземпляр головної групи і підмножина відносяться до нього примірників підпорядкованої групи, а зв'язки між елементами списку реалізуються за допомогою прийнятої системи покажчиків. При цьому число покажчиків, необхідних для реалізації кожного примірника зв'язку за допомогою системи

показчиків на екземпляри підпорядкованої групи, визначається коефіцієнтом зв'язку.

Структура запиту представляється деревом пошуку, що задаються на графі Γ (С, Γ /,) концептуальної структури БД. Як кореневої вершини дерева пошуку окремого запиту повинна бути вказана група, починаючи з якої здійснюється формування вихідних структур. Кореневої вершиною може бути група будь-якого рівня графа Γ (С, Γ /,). Коренева вершина є початком пошуку, і тому до неї повинні бути вказані альтернативні варіанти доступу. Будь-який з варіантів доступу починається в однію з точок входу графа ОС, ід і закінчується в кореневій вершині дерева пошуку. Точками входу в Γ (С, Γ /,) можуть бути підмножини вершин, відповідних групах першого рівня ієрархії (Γ ,) за умови існування шляху на графі Γ (С, С /,) від груп даного підмножини до кореневої вершині дерева пошуку, або вершина, відповідна групі- вершині дерева пошуку. Якщо коренем дерева пошуку є група рівня Ц, для неї існує тільки один варіант доступу. Будь-який інший варіант доступу до кореня пов'язаний з проходженням деякого безлічі зв'язків і груп концептуальної структури БД, які визначають дерево пошуку. Кожен гілка дерева пошуку задає шлях до груп даних, з яких вибираються елементи, необхідні при формуванні вихідних структур. Проходження кожного шляху, що входить до складу дерева пошуку, супроводжується видачею інформації з примірників груп, що задаються умовами пошуку. Завдання умов пошуку повинно визначати також напрямок і стратегію обходу вершин дерева пошуку. Стратегія обходу в значній мірі визначається рекурсивно дерева, яка означає, що кожне дерево (крім вироджених випадків) розпадається на ряд піддерев, які починаються в корені і які, в свою чергу, є деревами, у яких можуть бути свої піддерева і т.д [25].

Пошук необхідних елементів даних на дереві пошуку полягає у виконанні ряду обмінів із зовнішньою пам'яттю з метою відбору необхідних примірників груп і безпосередньому пошуку шуканих елементів в зв'язкових списках примірників зв'язків. При цьому пошук в зв'язковому списку зводиться до виконання двох основних операцій: перегляд зображення й вибір показника

зв'язку на наступний елемент списку і порівняння ключів елемента списку з ознаками, які задаються в умовах запитів

Кожен запит $q_k \in Q$, где $Q = \{q_k / k = 1, K\}$ - безліч запитів до БД, повинен однозначно визначати необхідну для його реалізації дерево пошуку, тобто безліч утворюють його груп (вершин дерева) і зв'язків між ними. Запит до БД задається за допомогою матриці використуваних їм груп $A = \|a_{ik}\|$ і матриці зв'язків $F = \|f_{ii'}^k\|$, прохідних при пошуку необхідних інформаційних елементів. Тут: $A_{ik} = 1$, якщо i -я група входить до складу k -го дерева пошуку (запиту), $a_{ik} = 0$, в іншому випадку; $f_{ii'}^k = 1$, якщо (i, i') -я взаємозв'язок використовується k -м запитом при пошуку, $f_{ii'}^k = 0$, в іншому випадку.

Основними параметрами безлічі запитів і відповідних їм дерев пошуку є такі:

$\sigma = \{\sigma_1, \dots, \sigma_k, \dots, \sigma_K\}$ - вектор частот використання запитів, де σ_k - частота запиту до-го типу;

$\beta = \{\beta_{ii'}^k / f_{ii'}^k = 1, i, i' = \overline{1, I}, k = \overline{1, K}\}$ - безліч середніх значень сумарного числа переглядаються при пошуку показників зв'язку і порівнюваних примірників груп, де $\beta_{ii'}^k$ - середнє сумарне число переглядаються показників (i, i') -й зв'язку, що входить в шлях доступу до-го запиту, і порівнюваних примірників групи S .

$P = \{p_{ii'}^k / f_{ii'}^k = 1, i, i' = \overline{1, I}, k = \overline{1, K}\}$ - безліч середніх значень сумарного числа обраних при пошуку примірників груп, де $P_{ii'}^k$ - число обраних примірників S (i, i'), що входить в шлях доступу до-го запиту і групи S .

Значення параметрів β і P залежать від умов сформульованих до БД запитів, які, в свою чергу, задають різні варіанти пошуку даних.

Для опису режиму первинного завантаження БД необхідно визначити загальну кількість завантажуваних примірників для кожної зв'язку концептуальної структури БД u, g . Необхідно відзначити, що для запитів, що вимагають обхід дерев пошуку на графі S ($S, S /$) в прямому і зворотному напрямках і використовують з цієї причини комбіновану систему показників,

кількість листів для завантаження примірників зв'язків в загальному випадку подвоюється в порівнянні з запитами, які вимагають обхід дерева пошуку тільки в прямому напрямку. Таким чином, уп, визначається наступним чином: $u_i = 1$, якщо зв'язок (« ') проходить тільки в прямому напрямку від головної до підлеглої групи, $u_i = 2$, якщо $m > i$, $u_i = 0$, якщо $m > i$, $u_i = 0$.

Параметр u_i , є елементом множини $Y = \{y_{i'} / i, i' = \overline{1, I}\}$ завантажуються примірників зв'язків.

Завдання на коригування даних включають наступну послідовність операцій: доступ до БД; пошук даних, що підлягають коригуванню; внесення безпосередніх змін (зміни, додавання, видалення); запис відкоригованих даних в БД.

Виходячи з вищевикладеного, завдання на коректування характеризуються підмножиною даних, що підлягають коригуванню, деревами їх пошуку і тимчасовими характеристиками пошуку, внесення змін і зчитування / запису відкоригованих даних.

Позначимо через $K = \{K_s / s = \overline{1, S}\}$ -множествозаданійна коригування БД.

Завдання на коригування описуються за допомогою матриці використання груп інформаційних елементів при виконанні завдань: № $W^{kop} = \|w_{si}^{kop}\|$, где $w_{si}^{kop} = 1$, завдання змінює 1-й груповий елемент і $m > 0$ до 0 в іншому випадку; вектора частот використання завдань на коригування адміністратором БД $A_{кор} = \{f_{кор}^T\}$, де $f_{кор}$ - частота використання б-го завдання адміністратором БД; безлічі середніх значень коректованих примірників груп даних $D_{кор} = \{y_{кор}^T\}$, де $y_{кор}$ - середнє значення кількості коректованих примірників б-й групи даних при виконанні б-й коригування.

Таким чином, дані для задачі синтезу оптимальних логічних структур БД включають:

1. Формалізований опис концептуальних структур, що задається системою множин векторів і матриць, що задається системою множин векторів і матриць

$$C = \{c_i / i = \overline{1, I}\}, W = \|w_{ii'}\|, R = \{r_i / i = \overline{1, I}\}, \rho = \{\rho_i / i = \overline{1, I}\}, \Delta = \{\delta_{ii'} / i, i' = \overline{1, I}\};$$

2. Формалізований опис запитів до БД, що задається безліччю

$$Q = \{q_k / k = \overline{1, K}\}, \text{ матрицями } A = \|a_{ik}\| \text{ и } F = \|f_{ii'}^k\|, \text{ вектором } \sigma = \{\sigma_k / k = \overline{1, K}\},$$

множинами $\beta = \{\beta_{ii'}^k / i, i' = \overline{1, I}; k = \overline{1, K}\}, P = \{p_{ii'}^k / i, i' = \overline{1, I}; k = \overline{1, K}\}, y = \{y_{ii'} / i, i' = \overline{1, I}\}.$

3. Формалізований опис завдань на коригування, що задається безліччю

$$K = \{K_s / s = \overline{1, S}\}, \text{ матрицею } W^{kop} = \|w_{si}^{kop}\|, \text{ вектором } \Lambda^{kop} = \{\xi_s^{kop} / s = \overline{1, S}\} \text{ і множиною}$$

Y.

3.7. Методи розрахунку основних характеристик структур БД

Основними характеристиками, використовуваними при постановці і вирішенні задач синтезу логічної структури БД, є значення сумарного числа відвіданих показників зв'язку і порівнюваних примірників груп (Д *) і середнє значення сумарної кількості обраних при пошуку примірників груп (р *). Конкретні значення цих характеристик залежать від структури запитів, що визначає різні варіанти пошуку даних.

Для отримання аналітичних виразів для розрахунку даних показників розглянемо можливі варіанти пошуку примірників і'-ї групи, що лежить на (// ') -м шляху доступу до-го запиту.

Шлях доступу дерева пошуку к-го запиту спрямований від групи С, до групи = 1) і С- головна, а С, - підлегла групи в розглянутому відношенні. Припустимо, що на попередньому етапі пошуку з групи С, було відібрано деяку підмножину її примірників γ^* у $\gamma' = 1, / *$ і за умовою к-го запиту з групи С γ потрібно вибрати запитувана підмножина γ^* , її примірників $\gamma^* = \setminus, pp.$ Тоді в залежності від співвідношення значень γ^* і γ^* можливі наступні варіанти пошуку примірників групи С γ :

Початок доступу

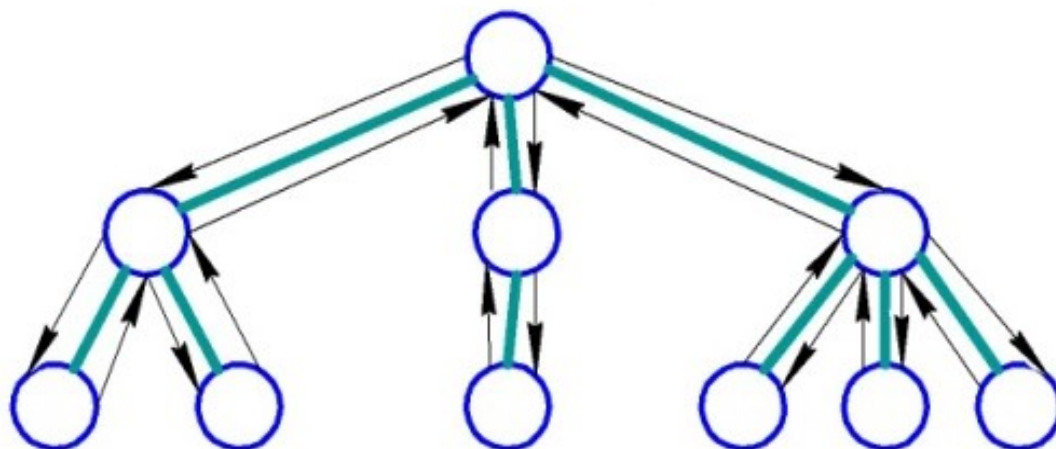


Рис.3.8. Стратегія обходу вершин дерева пошуку

Визначимо для кожного варіанта пошуку середнє сумарне число переглядаються показчиків зв'язку і порівнюваних примірників груп D^* . Очевидно, що $D^* = D^* \cdot \chi D^*$, де p - середнє число переглядаються показчиків зв'язку, $P \wedge p$ - середнє число порівнюваних примірників груп.

Так як операція перегляду показчика зв'язку пов'язана з вибіркою відповідного йому примірника Γ -ї групи і порівнянням його з ознакою, що задається в умові запиту, то $p = D \wedge p$. Дане співвідношення справедливо для всіх варіантів пошуку. Отже $D^* = 2D \cdot P$. Для розрахунку D^* ср доцільно використовувати процедури визначення середнього числа порівнюваних елементів списку, оскільки кожен екземпляр зв'язку за допомогою обраної системи показчиків представляється у вигляді зв'язного списку. Для першого варіанту пошуку маємо: $p = \wedge p$, де p - ймовірність

$$\beta_{ii'}^{kcp} = \sum_{n=1}^{\delta_{ii'}} n \frac{1}{\delta_{ii'}} = \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil.$$

Звідки

$$\beta_{ii'}^k = 2 \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil.$$

Для другого варіанту пошуку

$$\beta_{ii'}^{kcp} = \delta_{ii'},$$

Отже

$$\beta_{ii'}^k = 2\delta_{ii'}.$$

Для третього варіанту пошуку потрібно розглянути два випадки:

1) безліч ознак, що задаються в умовах запитання, і безліч підлеглих примірників групи Сг однаково чинно упорядковано (за зростанням або спаданням ключових елементів); 2) дані безлічі неупорядковані. У першому випадку $\delta_{ii'} = 8$, і $D^* = 28$ „, тобто потрібно перегляд всього списку підлеглих примірників. У другому випадку, для пошуку γ^* примірників Г-ї групи, потрібно γ^* звернень до зв'язного списку. При кожному черговому зверненні до списку знаходиться один з примірників безлічі γ^* , після чого здійснюється повернення до початку списку, тобто до примірника головної групи, що забезпечується використанням комбінованої системи показників. Середнє число порівнюваних примірників групи Сг при прямому перегляді списку одно

$$\beta_{ii'(np)}^{kcp} = r_i^* \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil$$

і отже

$$\beta_{ii'(np)}^k = 2r_i^* \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil.$$

При зворотному перегляді здійснюється повернення в початок списку.

Кількість розглядаємих при цьому показників $\beta_{ii'(об)}^k$ дорівнює $\beta_{ii'(об)}^k = \beta_{ii'(об)}^{ky} = r_i^* - 1$. Для даного випадку маємо:

$$\beta_{ii'}^k = \beta_{ii'(np)}^k + \beta_{ii'(об)}^k = 2\beta_{ii'(np)}^{kcp} + \beta_{ii'(об)}^{ky} = 2r_i^* \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil + r_i^* - 1 = r_i^* \left(1 + 2 \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil \right) - 1.$$

Варіанти 4,5,6 і 7 характеризуються тим, що пошук здійснюється в r^* зв'язкових списках, $r^* > 2$. Для реалізації даних варіантів пошуку необхідно використовувати операцію повернення в початок аналізованого списку, тобто комбіновану систему показчиків зв'язку між групами C_i і C_{i+1} . Отже, для варіантів 4,5,6 і 7 $r^* = 2 + p\%$ (об), где № «*» - середнє сумарне число показчиків зв'язку, необхідних для зворотних переглядів зв'язків з метою вибору наступного списку безлічі. Значення параметра r_k також залежать від варіанту пошуку, що задається умовами запиту до групи C_i . Для розрахунку параметра r_k доцільно використовувати вирази, отримані для $r^* \leq r$, так як при одноразовому перегляді списку примірника зв'язку $r_k = P\% r$, а при багаторазовому значення параметра r_k не більш значення $P\% r$. Таким чином, для першого варіанту пошуку маємо

$$\beta_{ii'}^{kcp} = r_i^* \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil;$$

$$\beta_{ii'}^k = 2r_i^* \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil + r_i^* - 1 = r_i^* \left(1 + 2 \left\lceil \frac{\delta_{ii'} + 1}{2} \right\rceil \right) - 1;$$

Для п'ятого варіанту:

$$\beta_{ii'}^{kcp} = r_i^* \delta_{ii'};$$

$$\beta_{ii'}^k = 2r_i^* \delta_{ii'} + r_i^* - 1 = r_i^* (2\delta_{ii'} + 1) - 1;$$

Середня кількість коректованих примірників груп даних r^* при завданні безлічі коригувань БД визначається адміністратором БД на основі вимог по додаванню, зміні і видалення даних, а також специфікації задач обробки даних конкретної предметної області.

В основі реляційної структури БД лежить математичне поняття теоретико-множинного відносини, під яким розуміється підмножина декартова твору одного або декількох доменів. Під доменом розуміється безліч значень конкретного інформаційного елемента. Декартово твір доменів $0_1, 0_2, \dots$ визначається як безліч всіх кортежів $\langle g_1, g_2, \dots, g_n \rangle$ довжини n , таких, що $g_i \in 0_i, \dots, g_n \in 0_n$. Ставлення зручно представити у вигляді двовимірної таблиці, кожен

рядок якої гея є кортеж, а кожен стовпець - домен відповідного інформаційного елемента $z / e \in \mathcal{L}$, що входить до складу відносини \mathcal{S} . Кожній таблиці Γ , що є формальним поданням відносини \mathcal{S} , притаманні такі властивості: відсутні повторюють рядки ; відсутні однакові стовпці; всі елементи відповідних позицій матриці Γ є атомарними.

3.8.Створення графічних моделей баз даних концептуального і логічного рівнів

3.8.1.Створення графічної моделі баз даних концептуального рівня

Визначимо основні вимоги до створюваної моделі:

- модель повинна включати опис сутностей і їх взаємозв'язків, що представляють інтерес в аналізованій предметній області (ПО) і виявляються в результаті аналізу даних;
- модель повинна бути поданням точки зору користувача на предметну область і не залежати від програмного забезпечення СУБД і технічних рішень;
- модель повинна бути стабільною: можуть змінюватися прикладні програми, що обробляють дані, може змінюватися організація їх фізичного зберігання - концептуальна модель залишається незмінною;
- модель повинна задовольняти критерію відкритості, тобто необхідно передбачити можливість її розширення при введенні додаткових даних.

Розглянемо безліч E - безліч сутностей, які визначимо, як: $E = \{E_1, E_2, \dots, E_k\}$, де k - загальна кількість сутностей. Кожному елементу множини E ставиться у відповідність безліч примірників відповідної суті, наприклад: e безліч екземплярів $\{E_1, E_2, E_3, \dots, E_k\}$ для сутності.

Наведемо предметну область бази даних у вигляді графа:

$G_k = \langle (E, A, I), R_1, R_2, R_3, R_4 \rangle$ Граф G_k - є концептуальна модель БД (рис.3.9.), що включає опис сутностей і їх взаємозв'язків і є поданням точки зору користувача на предметну область. Модель відповідає критерію відкритості: при введенні

додаткових даних необхідно лише розширити носій моделей за рахунок доповнення новими елементами множин E , α . При появі нових інформаційних запитів необхідно розширити носій моделі за рахунок доповнення новими елементами безлічі I . У обох випадках необхідно відповідне розширення сигнатури моделі.

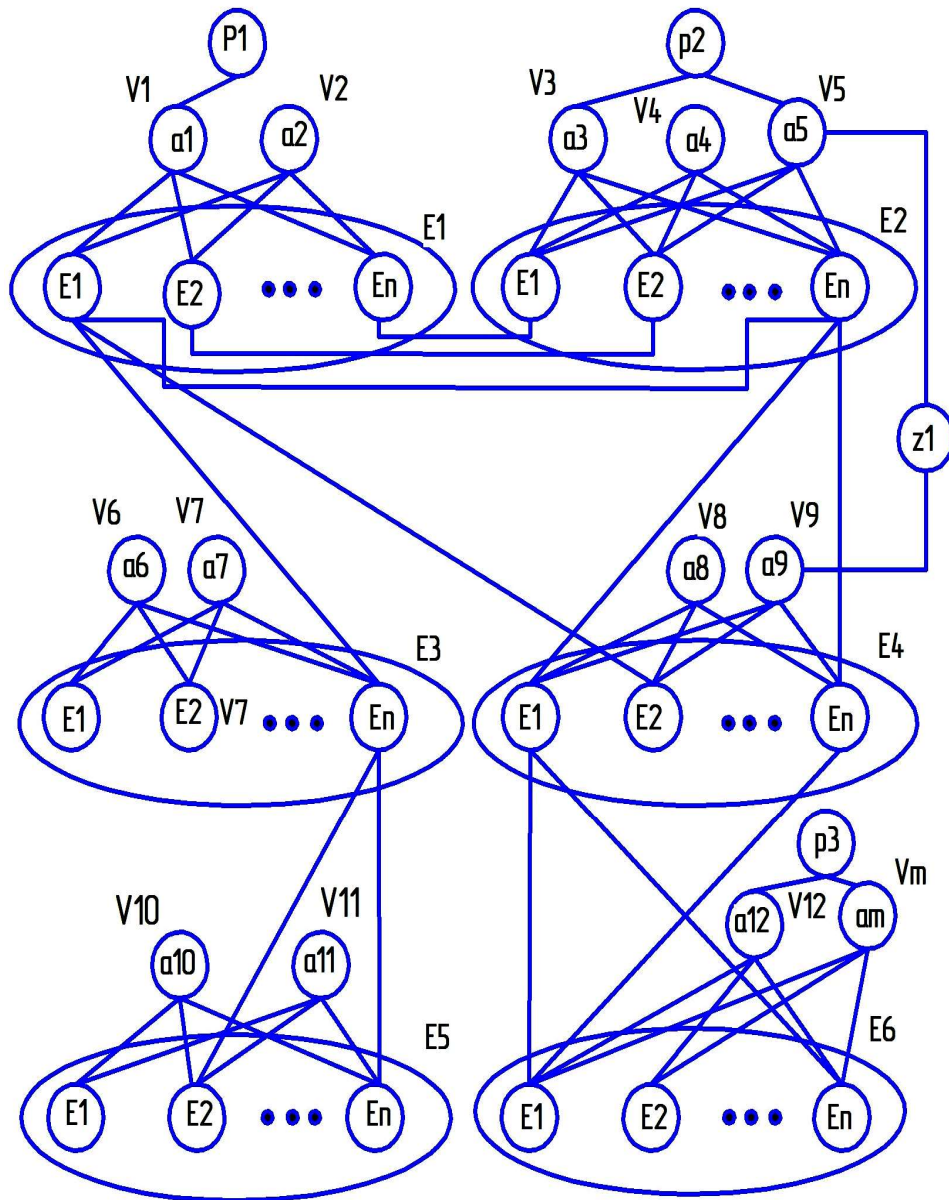


Рис.3.9. Граф G_k - концептуальна модель бази даних

В основу розробленої концептуальної графової моделі покладене широке застосовується в даний час модель П. Чена (ПМ - модель)

3.8.2. Створення графічної моделі баз даних логічного рівня

Основне призначення моделі логічного рівня - відображення логічних зв'язків між елементами даних незалежних від їх змісту і середовища зберігання. Розробляється логічна модель орієнтована на реляційну структуру БД.

Визначимо безліч T - безліч інформаційних масивів (ІМ), в яких будуть зберігатися дані проекрованої БД: $T = \{1, 2, \dots, n\}$. Таким чином, безліч T - це сукупність ІМ, що містять всю інформацію, яка повинна зберігатися в БД. Кожен ІМ - це реляційна таблиця або ж відношення. Математично відношення визначається наступним чином. Нехай дано 5 множин $\{A_1, A_2, \dots, A_n\}$, де A_i - елемент з T , $n \geq 2$ тоді L є ставлення над цими множинами, якщо L є безліч впорядкованих n -кортежів виду $\{(a_1, a_2, \dots, a_n)\}$, де a_i - елемент з A_i , $i = 1, 2, \dots, n$, називаються доменами відносини L .

Кожен ІМ (відношення) у БД зберігається в окремому файлі (рис.3.10).

Імена стовпців є атрибутами, а індивідуальні значення, що з'являються в окремих кортежі - значеннями атрибутів. Число стовпців щодо називають ступенем. Поточне число кортежів щодо називається потужністю. Ступінь відносини зазвичай не змінюється після його створення, але потужність буде коливатися в міру додавання нових і видалення старих кортежів. Число ІМ і конкретні атрибути, приписувані кожному ІМ, визначаються в процесі логічного проектування.

Зважимо кожен 1-ю вершину, відповідну елементу множини T числом X , що дорівнює кількості записів в 1-м ІМ. Зважимо кожен 1-ю вершину відповідну елементу множини A - числами i і U_i . V_i - кількість значень 1-го атрибута, A_i - довжина 1-го атрибута (в байтах). Припишемо ребрах $\{2\}$ аз ваги x_i , рівні кількості записів i -го ІМ, необхідних в 1-му інформаційному запиті.

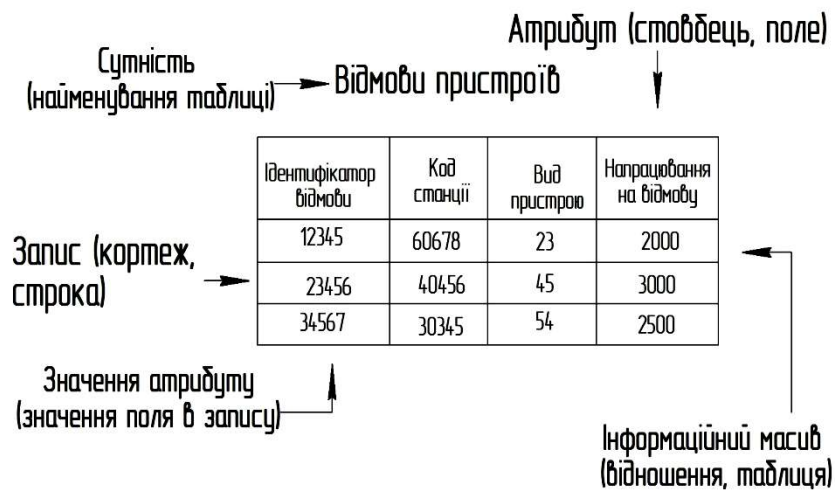


Рис.3.10. Інформаційний (логічний) масив

Відзначимо, що в графі не повинно бути подграфів виду (рис.3.8), тобто i -ий атрибут не може належати сам собі, а також i -й ІМ не може бути пов'язаний сам з собою.

У графі відображені логічні зв'язки між елементами даних, які не залежать від їх змісту і середовища зберігання. Дана логічна модель орієнтована на реляційну структуру БД.

3.8.3. Розробка правил перетворення моделі концептуального рівня в модель логічного рівня

Перетворення моделі концептуального рівня в логічну модель відбувається під управлінням методик, в яких досить чітко обумовлено всі етапи такого перетворення. Загально прийнятий підхід при відображенні ПМ - моделі на реляційну схему полягає в наступному: для будь-якої взаємозв'язку формується по одному відношенню для кожної сутності і по одному для кожної зв'язку, незалежно від ступеня зв'язку і класу приналежності сутностей.

Визначимо більш раціональний підхід до відображення концептуальної моделі на логічну, при якому генерація відносин проводиться за певними правилами, кожне з яких застосовується в залежності від ступеня зв'язку і класу приналежності кожної сутності. Так, наприклад, для взаємозв'язку виду

(рис.3.11) формується однакове число відносин - три (рис.3.12).

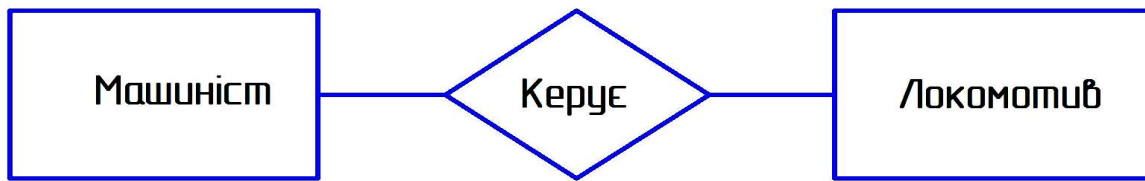


Рис.3.11. Взаємозв'язок виду 1:1

Машиніст			Керує		Локомотив		
Код машиніста	Прізвище	Дата народження	Код машиніста	Код локомотива	Код локомотива	Марка локомотива	Маса (т)
M12	Іванов	1959	M12	L23	L23	B/110	85
M57	Петров	1965	M57	L45	L45	B/111	90
M68	Сидоров	1963	M68	L57	L57	B/115	95

Рис.3.12. Перший варіант реалізації взаємозв'язку 1:1

Машиніст					
Код машиніста	Прізвище	Дата народження	Код локомотива	Марка локомотива	Маса (т)
M12	Іванов	1959	L23	B/110	85
M57	Петров	1965	L45	B/111	90
M68	Сидоров	1963	L57	B/115	95

Рис.3.13. Другий варіант реалізації взаємозв'язку 1:1

Правило 1. Якщо ступінь бінарної зв'язку дорівнює 1: 1 і клас приналежності обох сутностей є обов'язковим, то потрібно тільки одне відношення. Первинним ключем цього відношення може бути ключ будь-якої з двох сутностей.

Правило 2. Якщо ступінь бінарної зв'язку дорівнює 1: 1 і клас приналежності однієї сутності є обов'язковим, а другий - необов'язковим, то необхідно побудова двох відносин. Під кожен сутність виділяється одне

відношення. Ключ сутності, для якого клас приналежності є обов'язковим, додається в якості атрибута в відношення, що виділяється для сутності з обов'язковим класом приналежності

Правило 3. Якщо ступінь бінарної зв'язку дорівнює 1: 1 і клас приналежності жодної суті не є обов'язковим, то необхідно використовувати три відносини: по одному для кожної сутності і одне для зв'язку. Серед своїх атрибутів, що виділяється зв'язку, буде мати ключ сутності, що включає по одному ключу від кожної сутності.

Правило 4. Якщо ступінь бінарної зв'язку дорівнює 1: п і клас приналежності п - зв'язковий сутності є обов'язковим, то необхідно формування двох відносин. Ключ 1 - зв'язковий сутності повинен бути доданий як атрибут в відношення, що відводиться для п-зв'язковий сутності.

Правило 5. Якщо ступінь бінарної зв'язку дорівнює 1: п і клас приналежності п-зв'язковий сутності є обов'язковим, то необхідно формування трьох відносин: по одному для кожної сутності і одне для зв'язку. зв'язок повинна мати серед своїх атрибутів ключ сутності від кожної сутності.

Правило 6. Якщо ступінь бінарної зв'язку дорівнює ш: п, то для зберігання даних необхідно три відносини: по одному для кожної сутності і одне для зв'язку. Останнє ставлення має мати в числі своїх атрибутів ключ сутності від кожної сутності.

Так, наприклад, для сутностей і взаємозв'язків, представлених на рис 3.5 необхідне формування одного відношення (рис 3.13.).

З проведеного порівняння слід, що представлений метод є більш раціональним, тому що при такому підході не проводиться генерація зайвого числа відносин. Всі розглянуті правила, описані в термінах розроблених моделей зведені в табл. 3.1.

Проаналізувавши можливі результати перетворення правил перетворення GK -> GL, робимо висновок про те, що якщо для генерації відносин застосовуються правила 2-6 (табл. 3.1), то число інформаційних масивів, обсяг пам'яті, необхідний для зберігання проектованої бази даних і

час її первинного завантаження будуть мінімальні, а так як пошук всіх даних, необхідних для відповіді на інформаційні запити користувачів буде проводитися за допомогою сканування за вказівниками, то загальний час відповіді на запити буде мінімальним. цілі проектування

Таблиця 3.1.

Правила перетворення Gk - GL

Концептуальний рівень

Ступінь бінарний зв'язку	Клас приналежності	Концептуальний рівень	Логічний рівень
1	2	3	4
будь яка	будь який для і-ї і j-ї сутностей		
1:1	обов'язковий для і-ї і j-ї сутностей		

Тільки невеликі організації можуть усупільнити дані в одній повністю інтегрованій базі даних. Найчастіше адміністратор баз даних (навіть якщо це група осіб) практично не в змозі охопити і осмислити всі інформаційні вимоги, що пред'являються до системи. Тому інформаційні системи великих організацій містять кілька десятків БД, нерідко розподілених між кількома взаємопов'язаними ЕОМ різних підрозділів.

Окремі БД можуть об'єднувати всі дані, необхідні для вирішення однієї або декількох прикладних задач, інші БД можуть об'єднувати дані, що відносяться до будь-якої предметної області (наприклад, перевезень залізниць, фінансів, студентам, викладачам і т.д.). Перші зазвичай називають прикладними БД, а другі - предметними БД.

Предметні БД дозволяють забезпечити підтримку будь-яких поточних і майбутніх додатків, оскільки набір їх елементів даних включає в себе набори елементів даних прикладних БД. Внаслідок цього предметні БД створюють основу для обробки неформалізованих, змінюються і невідомих запитів та програм (додатків, для яких неможливо заздалегідь визначити вимоги до даних). Така гнучкість і пристосованість дозволяє створювати на основі предметних БД досить стабільні інформаційні системи, тобто системи, в яких більшість змін можна здійснити без вимушеного переписування старих додатків.

Засновуючи ж проектування БД на поточних і передбачуваних додатках, можна істотно прискорити створення вискоєфективної інформаційної системи, тобто системи, структура якої враховує найбільш часто зустрічаються шляхи доступу до даних. Тому прикладні епроектірованіє досі привертає деяких розробників. Однак у міру зростання числа додатків таких інформаційних систем швидко збільшується число прикладних БД, різко зростає рівень дублювання даних і підвищується вартість їх ведення.

Таким чином, кожен з розглянутих підходів до проектування впливає на результати проектування в різних напрямках. Бажання досягти і гнучкості, і ефективності призвело до формування методики проектування, що використовує як предметний, так і прикладної підходи. У загальному випадку предметний підхід використовується для побудови початкової інформаційної структури, а прикладний - для її вдосконалення з метою підвищення ефективності обробки даних.

На першому етапі проектування інформаційної системи створюється концептуальна модель даних. Необхідно провести аналіз цілей цієї системи і

виявити вимоги до неї окремих користувачів. Збір даних починається з вивчення сутностей організації бази і процесів, що використовують ці сутності. Сутності групуються по «схожості» (частоті їх використання для виконання тих чи інших дій) і за кількістю асоціативних зв'язків між ними (поїзд - пасажир, станція - стрілочний перевід, відмова пристрою - вигляд пристрою і т.д.). Сутності або групи сутностей, що володіють найбільшим подібністю і (або) з найбільшою частотою асоціативних зв'язків об'єднуються в предметні БД. (Нерідко сутності об'єднуються в предметні БД без використання формальних методик - по «здоровому глузду»).

Проектування АСУ БД здійснюється з використанням окремих реляційних предметних БД. Основна мета логічного проектування БД - це скорочення надмірності збережених даних, а отже, економія обсягу використовуваної пам'яті, зменшення витрат на багаторазові операції оновлення надлишкових копій і усунення можливості виникнення протиріч через зберігання в різних місцях відомостей про одне й те ж об'єкті. Так званий, «чистий» проект БД («Кожен факт в одному місці») можна створити, використовуючи методику нормалізації відносин. Для малих БД (що включають не більше 15 атрибутів) при проектуванні БД в якості відправної точки може використовуватися універсальне відношення.

4. РОЗРОБКА СТРУКТУРИ БАЗИ ДАНИХ АНАЛІТИЧНОГО ЦЕНТРУ АСУ БД

4.1. Проектування загальної структури бази даних АСУ БД

Сучасна структура інформаційної взаємодії на залізничному транспорті передбачає багаторівневий інформаційний обмін: лінійне підрозділ - відділення / при наявності / - ІОЦ дороги - ГВЦ МІС. Ідеологія Системи передбачає безпосереднє надходження первинної інформації в Аналітичні центри дорожнього і мережевого рівнів для забезпечення її достовірності - захисту від навмисного і ненавмисного перекручування. Пропонується два варіанти інформаційного обміну. Структури бази даних первинної інформації для першого і другого варіантів наведені на рис.4.1 і рис.4.2 відповідно.

При виборі першого варіанту (рис.4.1) необхідно забезпечення заборони на зміну баз даних первинної інформації ІОЦ і ГВЦ. Іншим можливим варіантом (рис.4.2) є безпосереднє надходження первинної інформації від автоматичних пристроїв і відповідальних осіб по каналах передачі даних в Аналітичний центр мережевого рівня; її короткострокове (не більше року) архівування для забезпечення порівнянності з базами даних дорожнього і мережевого рівнів (ІОЦ і ГВЦ).

Перевагами першого варіанту є:

- підтримка сучасної структури інформаційного обміну відповідно до інформаційної структури залізничного транспорту (ІСЗТ);
- зменшення обчислювального та інформаційного навантаження на Аналітичний центр мережевого рівня, що функціонує на базі ПЕОМ;

Перевагами другого варіанту є:

велика ступінь захисту інформації від навмисного і ненавмисного перекручування.

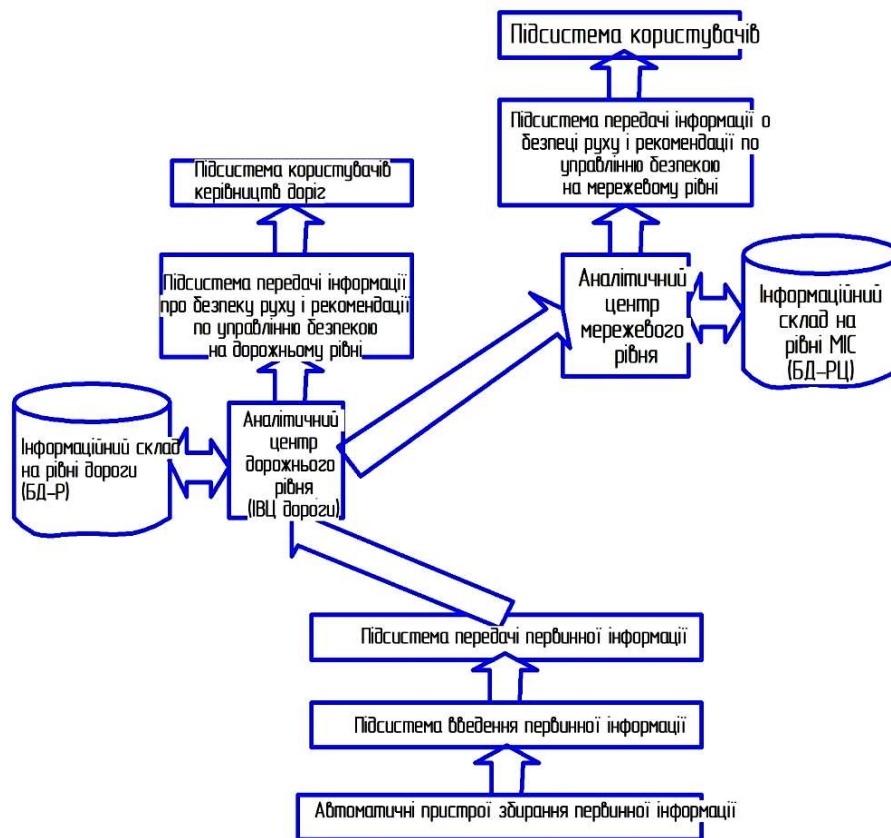


Рис.4.1. Структурна схема бази даних АСУ БД (1-й варіант)

У той же час інформація з усіх доріг передається в аналітичних центр мережевого рівня, де вона міститься в інформаційному сховищі рівня МІС (БД-РЦ). Після обробки первинної інформації і розрахунку фактичних показників безпеки руху виробляються рекомендації з управління безпекою руху на мережевому рівні.

Перший варіант побудований за принципом ієрархії (лінійні підприємства - відділення - міністерство). Тут інформація передається послідовно спочатку на ІОЦ, потім в ГВЦ. Використовуючи перший варіант, ми розвантажуюмо аналітичний центр мережевого рівня від додаткового навантаження, пов'язаної з обробкою інформації, отриманої від різних джерел (компонентів підсистеми введення первинної інформації). Тут вся інформаційна навантаження накладається на ІОЦ, а аналітичний центр мережевого рівня буде працювати тільки з тими даними, які необхідні безпосередньо для розрахунку показників безпеки руху поїздів і видачі вихідних форм.

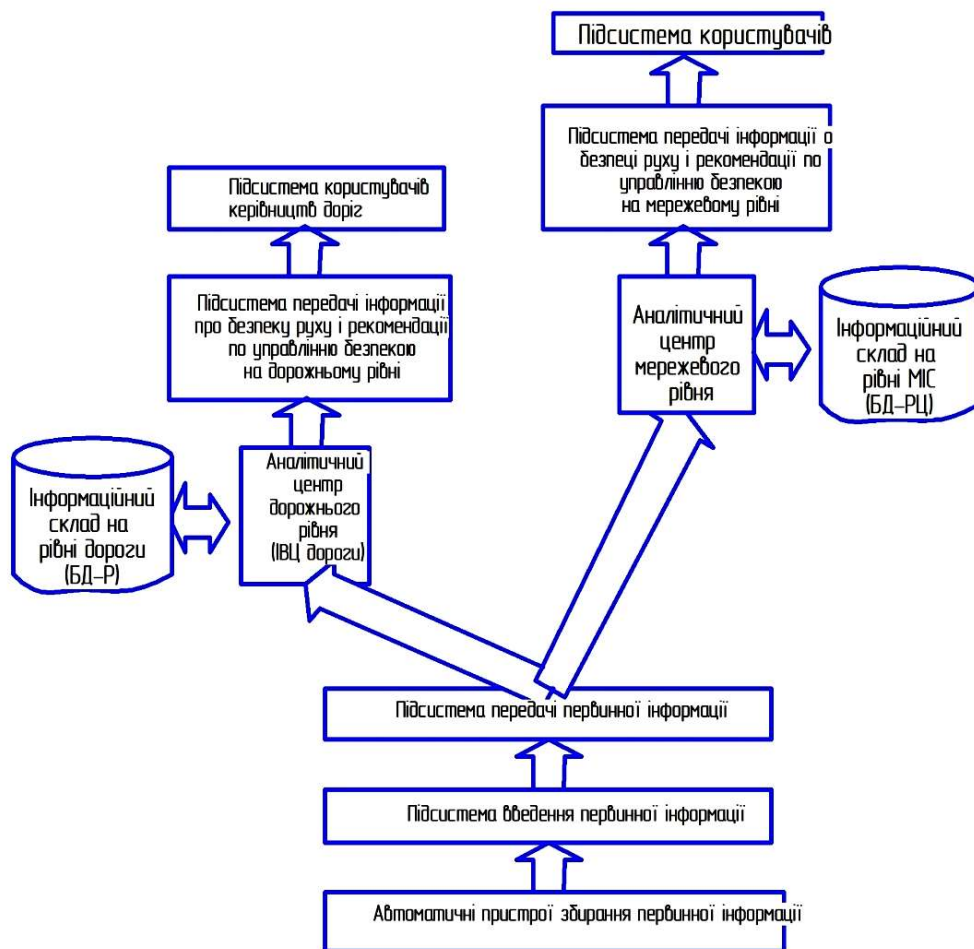


Рис.4.2. Структурна схема бази даних АСУ БД (2-й варіант)
 БД-Р – інформаційний склад дорожнього рівня

Другий варіант не передбачає отримання інформації з «залізних доріг». Стає очевидним, що немає необхідності отримувати необхідну інформацію для аналітичного центру мережевого рівня від підсистеми передачі первинної інформації. Використовуючи перший варіант побудови АСУ БД, всю необхідну інформацію для розрахунку показників безпеки на мережевому рівні аналітичний центр мережевого рівня буде отримувати з ІОЦ доріг.

Для запобігання можливості спотворення інформації, що передаються дані необхідно перевіряти використовуючи сучасні методи контролю передачі інформації, зокрема за допомогою використання відомих функції агрегування структурованого мови запитів).

Виходячи з вище викладеного, впливає висновок, що для розробки АСУ БД використовуватиметься перший варіант. Узагальнена схема банку даних аналітичного центру представлена на рис.4.3.

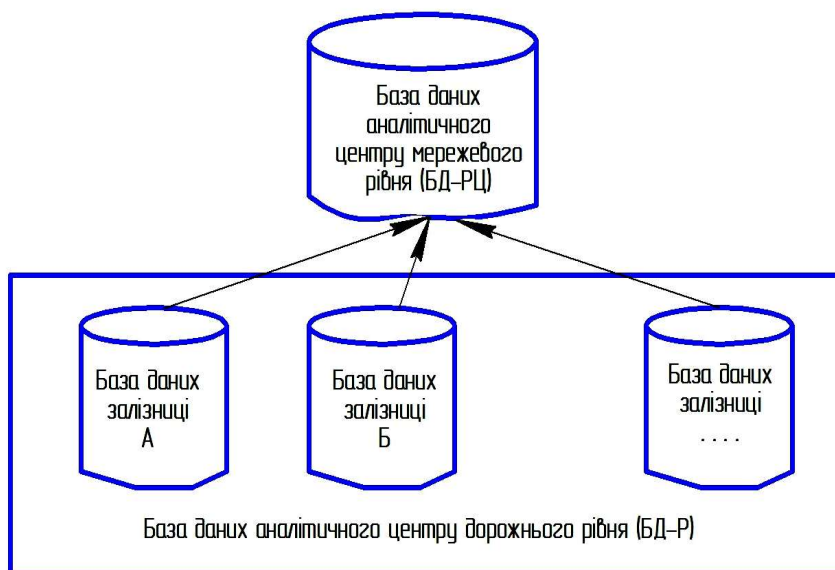


Рис.4.3. Узагальнена схема банку даних аналітичного центру

Вихідними даними для розробки структури банку даних аналітичного центру мережевого рівня системи АСУ БД (БД-РЦ) є форми статистичної звітності [65]. Через великий обсяг об'єктів розглянутої предметної області, виділеної за допомогою аналізу форм статистичної звітності, було прийнято рішення про розбиття бази динних за смисловим змістом на окремі області, для спрощення проектування БД-РЦ.

БД-РЦ включає в себе:

- Область полігонів (ПОЛІГОНИ);
- область функціонування технічних засобів (ТЗ);
- область функціонування персоналу (ПЕРСОНАЛ);
- область безпеки руху поїздів (БДП);
- область кількості рухомих одиниць, технічних засобів, обсягів вантажів, пасажирів (КОЛ);
- область втрат вантажів, технічних засобів, здоров'я пасажирів (ВТРАТИ);

- область гірок сортувальних станцій (ГОРКИ).

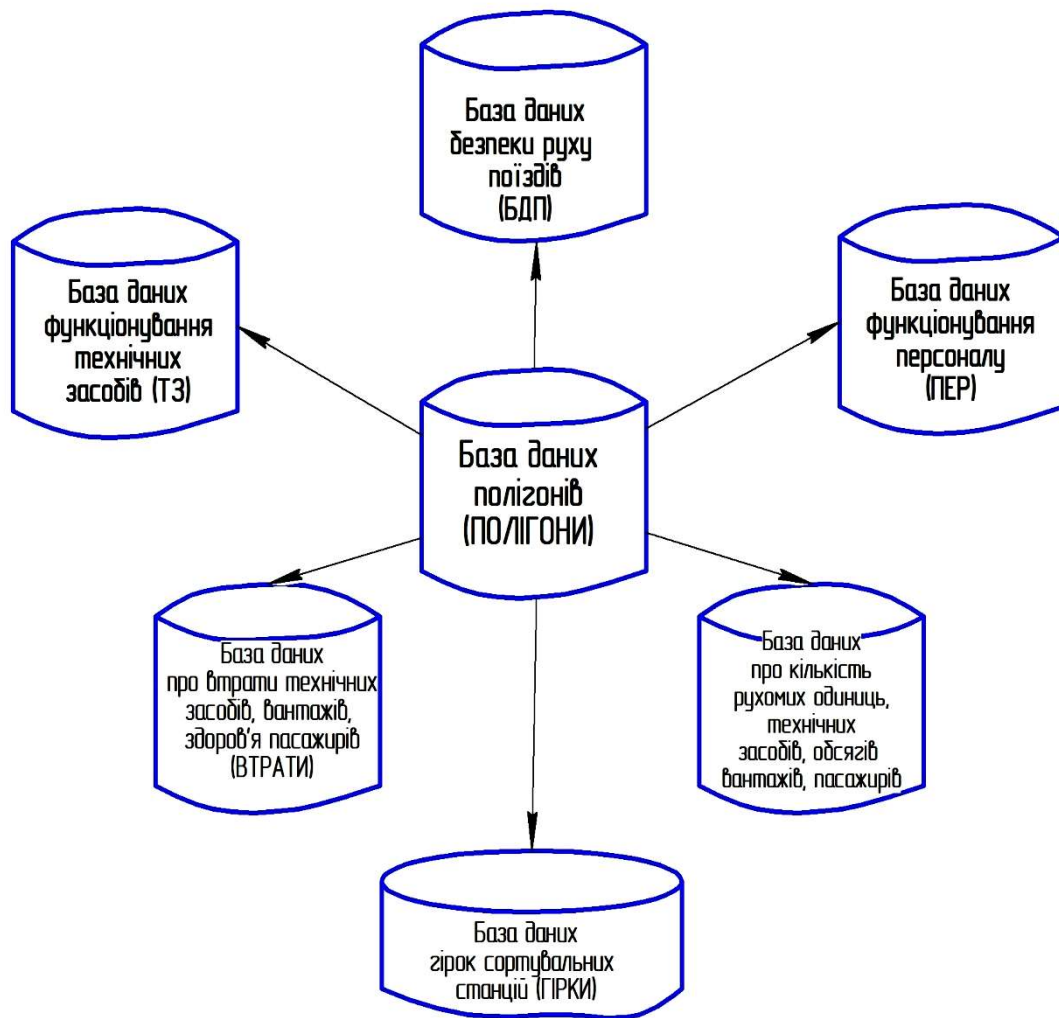


Рис.4.4. Структурна схема організації БД-РЦ

Таким чином, був зроблений вибір загальної структури передачі даних в АСУ БД і визначено структурну схему БД-РЦ АСУ БД, виходячи з аналізу розглянутої предметної області. Важливою структурною складовою АСУ БД є нормативно-довідкова інформація (НДІ).

При розробці бази даних АСУ БД використовуються загальногалузеві класифікатори та довідники залізниць (УЗ), а також специфічні для АСУ БД довідники. Поле «Ідентифікатор станції» таблиці «Стрілочні переводи» є зовнішнім ключем до поля «Ідентифікатор станції» таблиці «Станції».

При зміні коду станції таблиці «Станції», в інших таблицях посилаються на таблицю «Станції» ніяких змін проводити не потрібно.

При аналізі існуючих систем на залізничному транспорті була виявлена ще одна проблема пов'язана зі зміною інформації в НДІ. У ряді систем не передбачена можливість зберігання відкоригованої інформації, тобто неможливо отримати інформацію про дані, які існували до змін.

Для того, щоб вести історію зміни інформації в НДІ було вирішено завести ще додаткові поля в таблиці НДІ: «Дата початку дії запису», «Дата закінчення дії записи», додатковий «Унікальний ідентифікатор».

Відомі випадки перекодування довідників, в результаті чого виникають проблеми, пов'язані з порушенням посилальної цілісності (для значень зовнішнього ключа в екземплярі батьківської суті не існує відповідне значення батьківської сутності). Тому крім кодів в таблицях бази даних АСУ БД пропонується ввести додаткові поля - унікальні ідентифікатори, які будуть первинними ключами в таблицях НДІ [5].

Ефективність цього методу можна продемонструвати на таких прикладах. Таким чином, при створенні бази даних аналітичного центру АСУ БД будуть використовуватися таблиці НДІ з використанням ідентифікаторів і зовнішніх ключів, а для необхідності ведення історії зберігання інформації, паралельно будуть створені таблиці з додатковими полями «Унікальний ідентифікатор», «Дата початку», «Дата закінчення» дії записі. Определеніє предметної області БД-РЦ АСУ БД. Для розрахунку фактичних показників безпеки руху поїздів і функціонування технічних засобів, а також показників безпеки функціонування персоналу в межах полігону за заданий період часу необхідна наявність інформації про відмови руху поїздів, технічних засобів і помилок персоналу.

4.2. Поточне планування поїздоутворення та відправлення поїздів

План формування потягів (ПФП) - це план організації вагонопотоків у спеціалізовані потяги. На даній станції передбачено ПФП 24 сортувальних

колій.

Поїздоутворення на станції включає розформування - формування поїздів на гірці, накопичення вагонів і закінчення формування поїздів. Всі ці процеси між собою і багато в чому залежать від числа колій в сортувальному парку і їхнього використання відповідно до плану формування потягів, а також взаємодії в роботі гірки і витяжок. Варто враховувати суперечливість вимог, запропонованих до спеціалізації сортувальних колій. Крім того, у зв'язку з різною потужністю окремих призначень поїздів і нерівномірного підходу вагонів доводиться в оперативному порядку змінювати спеціалізацію колій, використовуючи вільні в даний момент сортувальні колії як для вагонів тих призначень, спеціалізовані колії які зайняті накопиченими поїздами в очікуванні й у процесі формування. Тому в оперативних умовах використовується принцип гнучкої спеціалізації сортувальних колій.

Однак після завершення накопичення составів у сортувальному парку доводиться виконувати операції по закінченню їхнього формування і перестановці в парк відправлення. Дуже ефективним виявляється двостороннє закінчення формування складу з гірки і з боку витяжки.

Для дослідження роботи системи необхідно розрахувати показники і нерівномірність надходження потягів у систему формування, розрахунковий параметр обслуговування, кількість обслуговуючих пристроїв і оптимальний рівень завантаження обслуговуючого пристрою.

Нерівномірність надходження потягів у систему формування залежить від інтенсивності накопичення вагонів на коліях сортувального парку.

Шляхом розрахунків знайдена величина сукупності моментів завершення формування одnogрупних , двогрупних і збірних потягів і оптимальний середній час на закінчення формування всіх категорій потягів.

$$T_{\text{ЕФ}}^{\text{ОД}} = T_{\text{ОС}} + T_{\text{ПДТ}} + T_{\text{ПТЕ}},$$

$$T_{\text{ЕФ}}^{\text{ДВ}} = T_{\text{ОС}} + T_{\text{ПДТ}} + T_{\text{ПТЕ}}^{\text{ХВ}} + T_{\text{ПТЕ}}^{\text{ГОЛ}},$$

$$T_{\text{ЕФ}}^{\text{ЗБ}} = T_{\text{ОС}} + T_{\text{ПДТ}} + T_{\text{С}} + T_{\text{ЗБ}},$$

$$T_{E\Phi} = \frac{N_{OD}}{\sum N} \times T_{E\Phi}^{OD} + \frac{N_{DB}}{\sum N} \times T_{E\Phi}^{DB} + \frac{N_{ЗБ}}{\sum N} \times T_{E\Phi}^{ЗБ},$$

де $T_{E\Phi}^{OD}$ - тривалість закінчення формування одногрупного поїзда;

$T_{E\Phi}^{DB}$ - тривалість закінчення формування двогрупного поїзда;

$T_{E\Phi}^{ЗБ}$ - тривалість закінчення формування збірного поїзда;

$T_{E\Phi}$ - середньозважений час на закінчення формування одного составу по станції "К";

T_{OC} - час на осаджування вагонів з боку витяжних колій;

$T_{ПДГ}$ - час на підтягування вагонів з боку витяжних колій:

$$T_{ПДГ} = 0,08 \times m_{\phi},$$

$T_{ПТЕ}$ - технологічний час на виконання операцій, пов'язаних із розстановкою вагонів по ПТЕ:

$$T_{ПТЕ} = B + E \times m_{\phi},$$

де В, Е - нормативні коефіцієнти, значення яких залежать від середньої кількості розчеплень на один сформований состав;

m_{ϕ} - розмір составу сформованого поїзда;

ТС - тривалість сортування вагонів:

$$T_C = A \times g_0 + B \times m_{\phi},$$

де А, Б — нормативні коефіцієнти, що залежать від умов сортування вагонів;

g - кількість відчепів у составі;

ТЗБ - тривалість збирання вагонів:

$$T_{ЗБ} = 1,8 \times P + 0,3m_{\phi},$$

де Р — кількість колій, із яких здійснюється перестановка вагонів на колію збирання.

Таким чином робимо висновок, що середній час на закінчення формування одногрупного потяга склав - 12 хвилин (0,2 години), для двогрупного — 24 хвилини (0,4 години), для збірного — 30 хвилин (0,5

години), що дало можливість визначити середній час на закінчення формування всіх категорій потягів 15 хвилин (0,25 години).

До елементів розрахункового параметра обслуговування системи відносять:

- вхідний потік - сукупність моментів завершення накопичення на групі сортувального парку прикріплених до витяжки (локомотиву); черга - число накопичених составів, очікуючих початку формування; обслуговуючі пристрої - витяжні колії (маневрові локомотиви); вихідний потік - сукупність моментів закінчення виставки сформованих на даній групі сортувального парку составів у парк відправлення.

Інтенсивність вхідного потоку розраховується по формулі:

$$\lambda_{\text{вх}} = \frac{N_{\phi}}{24 \times M_{\phi}},$$

де N_{ϕ} - кількість сформованих потягів, $N_{\phi} = 50$ потягів;

M_{ϕ} - кількість витяжних локомотивів, $M_{\phi} = 2$ локомотиви.

$$\lambda_{\text{вх}} = \frac{50}{24 \times 2} = 1,02$$

Інтенсивність обслуговування локомотива з обліком часу на закінчення формування з боку гірки становить:

$$\mu = \frac{1}{t_{\text{оф}}^{\text{с}}},$$

$$\mu = \frac{1}{0,38} = 2,63$$

Система працює в стаціонарному режимі, тому що умова $\mu > \lambda$ виконується.

Після накопичення та закінчення операцій по формуванню состави потягів відправляються на ділянки в основному з колій сортувального парку, де спочатку підлягають технічному огляду. Хоча в тому випадку коли процес накопичення по не дає резерв часу, необхідний для проведення операцій по відправленню, состав поїзда переставляється в Ковельський парк.

Із составом поїзда свого формування виконуються такі операції: технічне

обслуговування і безвідчіпний ремонт вагонів, комерційний огляд і усунення несправностей, причеплення поїзного локомотива, випробування гальм, здача запломбованого пакета з перевізними документами локомотивній бригаді.

Про перестановку составу поїзда в парк відправлення черговий по станції сповіщає працівників ПТО, ПКО із зазначенням номера колії. Після перестановки він пред'являє оператору ПТО состав до технічного обслуговування із зазначенням колії кількості вагонів у поїзді, номерів головного і хвостового вагонів і часу відправлення поїзда. При відсутності парку відправлення сформовані поїзди обробляються на коліях сортувального парку після закінчення їх формування, звідки потім вони відправляються. Працівники ПТО, відгородивши поїзд сигналами установленим ТРА порядком, проводять технічне обслуговування і ремонт вагонів одночасно кількома групами (3-4 груп). Знайшовши несправності, оглядачі вагонів наносять на вагони помітки крейдою, а слюсарі, що йдуть слідом за ними, виконують ремонт. Оглядачі та ремонтники перевіряють гальмівну магістраль та гальмові прилади вагонів, усувають виявлені несправності, наповнюють гальмівну магістраль стисненим повітрям і проводять повну пробу гальм, а після причеплення поїзного локомотива - пробу гальм.

Паралельно технічному обслуговуванню приймальники поїздів і робітники оглядають поїзд з двох боків у комерційному відношенні, усувають комерційні несправності. Про готовність поїзда у комерційному відношенні приймальники по доповідають черговому по станції з наступним записом у книзі форми ГУ-98. Контрольна перевірка сформованого поїзда оператором СТЦ (фактично наявності та розташування в ньому вагонів) проводиться списуванням вагонів у процесі перестановки состава із сортувального парку в парк відправлення. При відправленні поїзда з колій накопичення він перевіряється оператором СТЦ, який проходить вздовж поїзда і повідомляє по радіозв'язку номери вагонів у СТЦ.

Під час накопичення вагонів оператор СТЦ по відправленню за даними листків обліку наявності та розташування вагонів на сортувальних коліях (веде

оператор-накопичувач) і даних перевізних документів на вагони оформляє натурний лист поїзда. Послідовність номерів вагонів у натурному листі повинна відповідати фактичній наявності й розташуванню вагонів у поїзді. Натурний лист поїзда складають у кількості, встановленій залізницею, але не менше ніж у трьох примірниках. Перший примірник вкладають у пакет з перевізними документами, другий вручають машиністу поїзного локомотива разом з пакетом, третій залишається на станції для обліку і передачі інформації. Якщо станція формування не передає інформацію, натурні листи складають у чотирьох примірниках. При цьому машиністу поїзного локомотива вручають два примірники натурального листа (не враховуючи примірника, вкладеного в пакет з документами). Один із них призначений для станції передавання інформації а інший - для станції призначення поїзда, четвертий примірник залишається на станції формування. Для збірних, передаточних та вивізних поїздів натурний лист складають у двох примірниках - один прямує з составом, інший залишається на станції формування.

Пакет з перевізними документами в запечатаному вигляді й один (або два) примірники натурального листа СТЦ пересилає черговому по парку відправлення. Він вручає його під розписку машиністу поїзного локомотива або головному кондукторові збірного поїзда. Типова норма часу обробки в парку відправлення поїзда свого формування становить 30 хв.

Елементами системи підготування потягів до відправлення є проведення необхідних операцій на коліях сортувального парку, технічний та комерційний огляди. При відправленні поїздів з Подільського парку обслуговування відбувається тими ж бригадами, що обслуговують транзитний потік.

Вихідний потік розраховується по формулі:

$$\lambda = \frac{N_{сф}}{24},$$

де $N_{сф}$ - середнє число потягів за добу відповідно свого формування, що відправляються на ділянки, $N_{сф}=50$ потягів.

$$\lambda = \frac{50}{24} = 2,04.$$

Операцією, що лімітує, технічний огляд і безвідчіпний ремонт. Тривалість цієї операції залежить від числа вагонів у составі, числа вагонів, що підлягають ремонту, тривалості ремонту і числа бригад робітників, що обслуговують окремі частини составу. Кількість бригад ПТО та завантаження їх визначимо:

$$\varphi = \frac{N_{сф} \times t_{ТО}}{24},$$

де $t_{ТО}$ - середня тривалість обслуговування составу;

$$t_{ТО} = (1 - \lambda) \times \frac{\tau \times m}{\chi} + \lambda \left(t_{рем} + \frac{\tau \times m}{2 \times \chi} \right),$$

де λ - частка составів, які потребують безвідчепного ремонту, $\lambda = 0,2$;

$t_{рем}$ - час безвідчепного ремонту, $t_{рем} = 0,25$ години.

$$t_{ТО} = (1 - 0,2) \times \frac{0,016 \times 57}{3} + 0,2 \left(0,25 + \frac{0,016 \times 57}{2 \times 3} \right) = 0,34 \text{ години.}$$

$$\varphi = \frac{50 \times 0,34}{24} = 0,62$$

Аналогічний розрахунок робиться і для двох бригад ПТО, що склало 0,46 години, а завантаження склало 0,97, що не задовольняє бажаному діапазону. Тоді інтенсивність обслуговування при $\lambda = 1,29$ для однієї групи складає:

$$\mu = \frac{1}{t_{ТО}},$$

$$\mu = \frac{1}{0,34} = 2,9$$

Умова $\lambda < \mu$ виконується при роботі трьох бригад оглядачів.

Таким чином достатня кількість бригад — три.

Робота системи відправлення поїздів за своєю технологією роботи схожа з технологією роботи системи прийому поїздів і повністю залежить від її роботи. В систему відправлення включені потоки, що надходять із системи прийому та системи маневрової роботи, тобто транзитний поїздопотік і поїздопотік свого формування.

$$\lambda_{вих} = \lambda_{ех}^{mp} + \lambda_{вих}^{C\Phi}$$

Оскільки вихідний потік складається з транзитного поїздопотіку і поїздопотіку в розформування, а вхідний вагонопотік дорівнює вихідному вагонопотіку, то

$$\lambda_{\text{вх}} = \lambda_{\text{вих}}.$$

За аналітичними розрахунками система масового обслуговування (тобто станція) цілком справляється з задоволенням заявок інформації, що надходить.

4.3. Можливості оперативного прогнозування виконання простою місцевого вагона на станції

Часто в практиці потрібно оцінити, наскільки випадкова варіація показника, що спостерігається закономірно, чи знаходяться коливання в межах нормативу закону або розподілу, чи виходять за рамки випадкових відхилень.

Методи теорії ймовірностей дозволяють з відомим ступенем точності відповісти на ці питання.

Використовуючи закон розподілу, можна оцінити на базі вибіркової сукупності можливі коливання (довірчі інтервали) математичного очікування і дисперсії в генеральній сукупності [20].

Точне знаходження довірчих інтервалів можливо тоді, коли вид закону розподілу випадкової величини заздалегідь відомий [23]. Як показало приведені дослідження, переважним законом розподілу значень більшості об'ємних і якісних показників є нормальний закон.

Довірчий інтервал будь-якого розглянутого параметра розподілу знаходиться з умови ймовірності виконання деяких нерівностей, у який входить цікавлячий нас параметр, отриманий на основі вибіркової даних.

Закон розподілу вибіркового параметра залежить від невідомих параметрів розподілу показника в генеральній сукупності. Умовно приймається за генеральну сукупність річний відрізок часу спостереження показника, а в деяких випадках - квартал місяця.

Отже, закон розподілу вибіркової характеристики параметра залежить від

невідомих параметрів генеральної сукупності. Однак, за допомогою деяких математичних прийомів вдається перейти в нерівностях від випадкової величини вибіркового параметра до іншої функції значень показника, що спостерігаються $x_1, x_2 \dots x_n$. Закон розподілу функції залежить не від невідомих параметрів генеральної сукупності, а тільки від числа спостережень – n (у нашому випадку від кількості днів спостереження) і від виду закону розподілу x [23].

Розглянемо практичний приклад оперативного прогнозування виконання плану простою місцевого вагона на станції.

При плановому завданні на січень $x_{пл} = 51$ год. виконання склало:

1 січня - 65 год.;

2 січня - 68 год.;

3 січня - 70 год.;

4 січня - 57 год.;

5 січня - 74 год.;

6 січня - 61 год.

Необхідно по шести днях оцінити, чи знаходяться виконання простою на рівні планового завдання.

Відповісти на це питання можна, розрахувавши довірчий інтервал I_β математичного чекання фактичного виконання плану простою [18].

Як видно з таблиці на рисунку 6.1, вибіркоче значення дисперсії $D(x)$ для шести днів з розрахунку складає 38 год².

Потрібно знайти оцінку $x_{сер}$ для $M(x)$ і побудувати для нього 95 % -вий I_β . Виходячи з прийнятих умов, довірча ймовірність $\beta = 0,95$.

По табульованим значенням інтеграла $\psi(\chi) = 2 \int_0^\chi f_{n-1}(t) dt$ зворотною інтерполяцією знаходиться величина t_β . t_β - це таке число, при якому ймовірність випадкової величини T , меншої t_β , дорівнює довірчій ймовірності [22].

Величина T розподіляється за законом Ст'юдента [22]. Тому

$$P(|T| < t_{\beta}) = \beta$$

t_{β} визначається по таблицях у залежності від β і числа ступенів волі $n-1=6-1=5$.

Установивши по таблицях $t_{\beta} = 2,57$, розраховуємо ε_{β} - половину ширини довірчого інтервалу I_{β}

$$\varepsilon_{\beta} = t_{\beta} \sqrt{D(x)/n}$$

$$\varepsilon_{\beta} = 2,57 \sqrt{38/6} = 6,5.$$

Довірчий інтервал I_{β} для математичного чекання $M(x)$ добового простою складе

$$I_{\beta} = (x_{\text{сер}} - \varepsilon_{\beta}; x_{\text{сер}} + \varepsilon_{\beta})$$

$$I_{\beta} = (66 - 6,5; 66 + 6,5);$$

$$I_{\beta} = (59,5; 72,5).$$

Неважко помітити, що довірчий інтервал математичного очікування не накриває добове планове завдання ($x_{\text{пл}} = 51$ год.).

Тому можна зробити висновок, що протягом шести днів план не виконується і варто очікувати невиконання плану простою місцевого вагона за місяць.

Тепер дамо оцінку $I_{\beta} M(x)$ простою по оперативним даним за п'ятнадцять перших днів місяця.

Обробка даних дала наступні результати:

$$x_{\text{сер}} = 857/15 = 57 \text{ год.}; D(x) = 1820/14 = 130 \text{ год}^2;$$

$$\beta = 0,95; t_{\beta} = 2,14 \text{ при } n-1=14;$$

$$\varepsilon_{\beta} = 2,14 \sqrt{130/15} = 6,3;$$

$$I_{\beta} = (57 - 6,3; 57 + 6,3);$$

$$I_{\beta} = (50,7; 63,3).$$

Довірчий інтервал хоча і накриває значення планового завдання $x_{\text{пл}} = 51$ год, однак, планове завдання знаходиться в лівому кінці інтервалу. Мається небезпека невиконання місячного плану.

Розглянемо $I_{\beta} M(x)$ на кінець другої декади. Обробка даних дала наступні параметри виборки за дві декади місяця

$$\bar{x} = 53 \text{ год.}; D(x) = 158 \text{ год}^2;$$

$$\beta = 0,95; t_{\beta} = 2,09 \text{ при } n-1=19;$$

$$\varepsilon_{\beta} = 2,09 \sqrt{158/20} = 5,9;$$

$$I_{\beta} = (53-5,9; 53+5,9);$$

$$I_{\beta} = (47,1; 58,9)$$

а) план не виконується:

Дата	x	x- \bar{x}	(x- \bar{x}) ²	n-1	$D(\bar{x})$
1	65	-1	1		
2	68	+2	4		
3	70	+4	16		
4	57	-9	81		
5	74	+8	64		
6	61	-5	25		
Разом	395	-	191	5	191/5=38

б) мається небезпека не виконання плану

Дата	x	x- \bar{x}	(x- \bar{x}) ²
1	65	+8	64
2	68	+11	121
3	70	+13	169
4	57	0	0
5	74	+17	289
6	61	+4	16
7	54	-3	9
8	76	+19	361
9	51	-6	36
10	47	-10	100
11	38	-19	361
12	52	-5	25
13	49	-8	64
14	51	-6	36
15	44	-13	169
Разом	857	-	1820

$$\bar{x} = 395/6 = 66 \text{ год.};$$

$$D(x) = 191/5 = 38 \text{ год}^2;$$

$$\beta = 0,95; t\beta = 2,57 \text{ при } n-1 = 6-1 = 5;$$

$$\varepsilon\beta = 2,57\sqrt{38/6} = 6,5;$$

$$I\beta = (66-6,5; 66+6,5);$$

$$I\beta = (59,5; 72,5).$$

$$\text{б) } x_{\text{сер}} = 857/15 = 57 \text{ год};$$

$$D(x) = 1820/14 = 130 \text{ год}^2;$$

$$\beta = 0,95; t\beta = 2,14 \text{ при } n-1 = 14;$$

$$\varepsilon\beta = 2,14\sqrt{130/15} = 6,3;$$

$$I\beta = (57-6,3; 57+6,3);$$

$$I\beta = (50,7; 63,3).$$

Довірчий інтервал математичного очікування накриває добове планове завдання. Значить у плинні двадцяти днів виконання показника простою місцевого вагона знаходилося приблизно на рівні Хпл.

Простежимо за поведженням $I\beta$ $M(x)$ середньодобового простою стосовно планового завдання на кінець місяця. При обробці даних отримані наступні параметри розподілу:

$$x_{\text{сер}} = 47 \text{ год}; D(x) = 179 \text{ год}^2;$$

$$\beta = 0,95; t\beta = 2,04 \text{ при } n-1 = 30;$$

$$\varepsilon\beta = 2,04\sqrt{179/31} = 4,9;$$

$$I\beta = (47-4,9; 47+4,9);$$

$$I\beta = (42,1; 51,9).$$

$I\beta$ значно змістився вліво, що свідчить про перевиконання планового завдання.

В результаті розрахунків було встановлено, що коливання простою місцевих вагонів підпорядковано нормальному закону розподілу, що дає змогу отримати середньоочікуваний простій вагонів на станції. За допомогою методів теорії ймовірностей було розглянуто практичний приклад оперативного прогнозування виконання плану простою місцевого вагона на станції. На основі отриманих результатів можна зробити висновок про виконання чи невиконання

планового завдання. Користуючись методами прогнозування можна заздалегідь розробляти і проводити заходи щодо покращення різноманітних показників і нормативів, що дасть змогу повністю контролювати ситуацію і у разі потреби втручатися з необхідними змінами у різноманітні процеси.

а) план виконується

Дата	x	x-хсер	(x-хсер) ²
16	53	0	0
17	42	-11	121
18	38	-15	225
19	27	-26	676
20	40	-13	169
Разом	1057	-	3001

б) перевиконання плану

Дата	x	x-хсер	(x-хсер) ²
21	23	-24	576
22	38	-9	81
23	49	+2	4
24	37	-10	100
25	51	+4	16
26	42	-5	25
27	19	-28	184
28	39	-8	64
29	34	-13	169
30	43	-4	16
31	24	-23	529
Разом	1456	-	5375

$$а) \text{хсер} = 1057/20 = 53 \text{ год};$$

$$D(x) = 3011/19 = 158 \text{ год}^2;$$

$$\beta = 0,95; t\beta = 2,09 \text{ при } n-1 = 19;$$

$$\varepsilon\beta = 2,09\sqrt{158/20} = 5,9;$$

$$I\beta = (53-5,9; 53+5,9);$$

$$I\beta = (47,1; 58,9).$$

$$б) \text{хсер} = 1456/31 = 47 \text{ год};$$

$$D(x) = 5375/30 = 179 \text{ год}^2;$$

$$\beta = 0,95; t\beta = 2,04 \text{ при } n-1 = 30;$$

$$\varepsilon\beta = 2,04\sqrt{179/31} = 4,9;$$

$$I\beta = (47-4,9; 47+4,9);$$

$$I\beta = (42,1; 51,9).$$

4.4. Розрахунок економічної ефективності впровадження АСК

Автоматизація процесів супроводжується зміною технології процесів, що автоматизуються, і дозволяє: раціоналізувати процес керування, спростити та здешевіти його, сприяє підвищенню якості і надійності.

Упровадження нової управлінської технології припускає:

- визначення економічної ефективності від упровадження системи;
- доцільність зміни існуючої технології.

Розглядаючи ефект від створення і функціонування системи оптимального оперативного розподілу потоків вагонів, потрібно визначати загальний економічний результат зміни технології перевезень.

Складові економічного ефекту:

- економія експлуатаційних витрат від скорочення простоїв (підвищення ефективності використання перевізних ресурсів),

- непряма ефективність.

У свою чергу, перераховані складові містять у собі:

- економія експлуатаційних витрат від скорочення вагоно-годин простою під різними операціями та на під'їзних коліях, що приводить до зменшення загального простою вагонів на станції;

- непряма ефективність - це річна вигода, що не піддається точному грошовому виміру – запобігання збитку від помилкових рішень, упущена вигода від не прийнятих під час рішень, скорочення часу на вироблення рішення, використання схованих технологій контролю виконавців.

Визначимо економічний ефект за рік від впровадження автоматизованих систем управління: Скорочення простою вагонів по відправленню, визначається за формулою [22]:

$$E_1 = N_{відпр}^{ваг} \cdot \Delta t \cdot e_{г-г} \cdot k ,$$

де, $N_{відпр}^{ваг}$ – кількість відправлених вагонів. ваг;

Δt – скорочення часу простою, ваг-год.;

ев-г – витратна ставка однієї вагоно-години, грн.;

k – коефіцієнт затримки з причин недостатньої якості оформлення документів та підготування попередньої інформації («людський» фактор), 1;

$$E = 1359634 \cdot 0,8 \cdot 1,689 \cdot 1 = 1837 \text{ тис.грн.}$$

Скорочення простою вагонів під однією вантажною операцією. Визначається за формулою [22]:

$$E = N_{місц}^{ваг} \cdot \Delta t \cdot e_{г-г} \cdot k ,$$

де, $N_{місц}^{ваг}$ – кількість місцевих вагонів. ваг;

Δt – скорочення часу простою місцевих вагонів, ваг-год.;

ев-г – витратна ставка однієї вагоно-години, грн.;

k – коефіцієнт затримки з причин недостатньої якості оформлення документів та підготування попередньої інформації («людський» фактор), 1;

$$E = 63078 \cdot 0,9 \cdot 1,689 \cdot 1 = 96 \text{ тис.грн.}$$

Скорочення часу користування вагонами на під'їзних коліях. Визначається за формулою [22]:

$$E = N_{відпр}^{ваг} \cdot \Delta T \cdot e_{г-г} \cdot k$$

де, $N_{відпр}^{ваг}$ – кількість вагонів забраних з під'їзних колій, ваг;

ΔT – скорочення часу користування вагонами на п/к, ваг-год.;

ев-г – витратна ставка однієї вагоно-години, грн.;

k – коефіцієнт затримки з причин недостатньої якості оформлення документів та підготування попередньої інформації («людський» фактор), 1;

$$E = 30660 \cdot 3,1 \cdot 1,689 \cdot 1 = 160 \text{ тис.грн.}$$

Статистика показує, що 30 % відсотків часу оперативно - диспетчерський апарат витрачає на інтелектуальну діяльність пов'язану з переробкою рішень, оцінкою поточної ситуації, пошуком раціональних рішень.

Скорочення часу на знаходження рішення, створення умов для запобігання прийняття помилкових рішень з мінімальними витратами складають непряму ефективність, формалізація і калькулювання якої неможлива.

Отже, автоматизація процесів дозволяє: скоротити експлуатаційні витрати (зменшення простою вагонів), підвищити якість і доступність інформації, вести контроль виконання, спростувати процес прийняття рішень, підвищити якість умов праці.

ВИСНОВКИ

В результаті виконання дипломної роботи отримані наступні основні результати:

1. Проаналізовано системи безпеки залізничного транспорту. Показана актуальність створення АСУ БД.

2. Виконано аналіз методів проектування баз даних. Показано, що, незважаючи на різноманіття методів створення БД в даний час відсутня єдина методологія їх проектування. На базі описаних досліджень встановлені переваги і недоліки різних підходів до проектування структур баз даних і виконано висновок про необхідність удосконалення методики їх проектування.

3. На підставі проведеного аналізу існуючих підходів і методик проектування баз даних обґрунтовано застосування комбінованого підходу до проектування систем баз даних, що дозволяє знизити розмірність розв'язуваних завдань. Встановлено склад завдань концептуального етапу проектування системи баз даних.

4. Розроблено методику проектування систем баз даних, що передбачає взаємозв'язок і рішення сформульованих завдань концептуального проектування СБД, що включає методику побудови організаційної моделі бази даних, методику побудови моделі інформаційного проходження і методу нормалізації інформаційних структур.

5. Виконано формалізований опис основних компонентів концептуальної і логічної структур баз даних з розробкою методу розрахунку основних характеристик концептуальних структур баз даних.

6. Розроблено методику створення графічних моделей концептуального і логічного рівнів з розробкою правил перетворення моделі концептуального рівня в модель логічного рівня.

7. Наведено послідовність кроків загальної методики логічного проектування.

8. Розроблено загальну структуру побудови бази даних АСУ БД,

використовуючи багаторівневий інформаційний обмін, з розробкою структурної схеми бази даних аналітичного центру мережевого рівня АСУ БД.

9. Розроблено метод підтримки таблиць НДІ в актуальному стані за допомогою додавання додаткових полів в структури таблиць.

10. Виконано проектування БД-РЦ АСУ БД, включаючи визначення предметної області системи, побудова концептуальної, логічної та фізичної моделей даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бабанова Н.И. Применение двухуровневой графовой модели для проектирования реляционных баз данных // Труды СКГТУ, выпуск 7, Владикавказ: Терек 2000.
2. Базы и банки данных знаний: Учебник для вузов по спец. “Автоматизированные системы обработки информации и управления” / Г.И. Ревунков, Э.Н. Самохвалов, В.В. Чистов; Под ред. В.Н. Четверикова. – М.: Высшая школа, 1992. – 367 с.
3. Беляев С.Н. Смысловое содержание информации, используемой в банке данных автоматизированной системы управления безопасности движения поездов (АСУ БД) сетевого уровня // Труды четвертой научно-практической конференции «Безопасность движения поездов» . – М.: МИИТ, 2003 г. – с. II-2.
4. Беляев С.Н. Основные критерии качества проектирования баз данных // Журнал «Объединенный научный журнал» . – М.: 2003 г. № 24. С. 70-71.
5. Беляев С.Н. Структура таблиц нормативно-справочной информации базы данных аналитического центра сетевого уровня автоматизированной системы управления безопасности движения поездов // Труды четвертой научно-практической конференции «Безопасность движения поездов», второе издание . – М.: МИИТ, 2003 г. – с. II-40.
6. Беляев С.Н., Лисенков В.М. Методика концептуального проектирования систем баз данных автоматизированной системы управления безопасностью движения поездов // Труды четвертой научно-практической конференции «Безопасность движения поездов», второе издание . – М.: МИИТ, 2003 г. – с. II-41.
7. Беляев С.Н. Основные этапы методики проектирования баз данных автоматизированной системы управления безопасностью движения поездов // IV Международная научно-практическая конференция «Участие молодых ученых, инженеров и педагогов в разработке и

- реализации инновационных технологий» – М.: МГИУ, 2003 г. – С. 177-179.
8. Богуславский Л.Б., Дрожжинов В.И. Основы построения вычислительных сетей для автоматизированных сетей. - М.: Энергоатомиздат, 1990. – 235 с.
 9. Бойко В.В., Савинков В.М. Проектирование информационной базы автоматизированной системы на основе СУБД: Практическое руководство. – М.: Финансы и статистика, 1982. – 174 с.
 10. Болтинский В.Г. Математические методы оптимального управления. М.: Наука, 1968. – 408 с., ил.
 11. Брешников А.В. Оперативная разработка баз данных средствами системы CLARION – М.: Машиностроение, 1995. – 63 с.
 12. Бритов П.А., Липчинский Е.А. Практика построения Хранилищ Данных: Система SAS. // СУБД, № 4-5, 1998, с. 60.
 13. Вейнеров О.М., Самохвалов Э.Н. Проектирование баз данных. – М.: Высшая школа, 1990. – 140 с.
 14. Дедегкаев А.Г., Яровая-Бабанова Н.И. Анализ информационных потоков и построение канонической структуры базы данных для подсистем “База данных городских инфраструктур”. / Труды СКГТУ, выпуск 6, Владикавказ: Терек, 1999.
 15. Дедегкаев А.Г., Бабанова Н.И. Запрещенные фигуры оптимизации структур локальных баз данных на этапе логического проектирования. / Сб. трудов аспирантов, Владикавказ, 2000.
 16. Диго С.М. Проектирование и использование баз данных: Учеб. для студентов вузов по направлению и спец. “Информационные системы в экономике” – М.: Финансы и статистика, 1995. – 208 с.
 17. Дюран Б., Одел П. Кластерный анализ – М.: Статистика, 1977. – 128 с.
 18. Ерема-Еременко А.А., Мииховский С.Д. Модель и язык описания предметной области // Банки данных: Тез. докл. 2 Всесоюз. Конф. Секция

3. Методы и системы проектирования баз данных. – Ташкент, 1983. – С. 20-22.
- 19.Зайцев Н.Г., Завадский П.И., Еремеев Л.Г. Автоматизация структурной декомпозиции информационной базы АСУ на основе модульной технологии // Ус и М.- 1985. – № 2. С. 16-21.
- 20.Замулин А.В. Типы данных в языках программирования и базах данных. – Новосибирск: Наука. Сиб. отд-ние, 1987. - 150 с.
- 21.Интерактивная разработка информационных систем / Геловани В.А., Безруков Д.И., Братков В.Г. и др. // Техническая кибернетика. – 1986. № 2. – С. 48-70.
- 22.Инфологическая модель в системе автоматизированного проектирования баз данных: концепции построения и реализации / Вейнеров О.М., Савинков В.М., Жадан Н.В., Казаров М.С. // Прикладная информатика, 1987. – Вып. 2. – С. 59-75.
- 23.Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. – М: Наука, 1983. - 424 с.
- 24.Крахт В.А., Росталу Э.П. Проектирование баз данных на основе реляционно-решетчатой концептуальной модели предметной области // УС и М. –1981. – № 4. – С. 22 – 28.
- 25.Кульба В.В., Ковалевский С.С., Косяченко С.А., Сиротюк В.О. Теоретические основы проектирования оптимальных структур распределенных баз данных. –М.: Российская академия наук. Институт проблем управления, 1999.- 660 с.
- 26.Мартин Дж. Планирование развития автоматизированных систем. – М.: Финансы и статистика. 1984, – 196 с.
- 27.Мейер Д. Теория реляционных баз данных. – М.: Мир, 1987. – 608 с.
- 28.Савинков В.М., Вейнеров О.М., Казаров М.С. Основные концепции автоматизации проектирования баз данных // прикладная информатика. – М.: Финансы и статистика, 1982. – Вып. 1. – С. 30 – 41.

- 29.Тиори Т., Фрай Дж. Проектирования структур баз данных: в 2-х кн. Кн. 1. Пер. с англ. – М.: Мир, 1985. – 287 с.
- 30.Филиппов В.Н. Теоретико - множественный подход к моделям данных // Банки данных: Материалы 3 Всесоюз. конф. – Таллин. 1985. С. 16 – 25.
- 31.Хаббард Дж. Автоматизированное проектирование баз данных. – М.: Мир, 1984. – 296 с.
- 32.Цаленко М.Ш. Семантические и математические модели баз данных. – М.: ВИНТИ, 1985. – 208 с.
- 33.Церелик Г.Г. Системы распределения баз данных. – Львов: Свит, 1990. – 167 с.
- 34.ANSI/X3/SPARC. Study group on data base management systems interim report FDT // Bull. ASM-SIGMOD. 1975. Vol. 7, N 2.
- 35.Abrial I.R. Data semantics // Data Base Management / - Amsterdam: North – Holland, 1974. P. 1 – 59.
- 36.Atzeni D., Carboni E. INCOD (A system for interactive conceptual design) revisited after the implementation of a prototypen // Entity – Relationship Approach to Information modeling and Analysis. – Amsterdam: North – Holland, 1983. - P. 449 – 464.
- 37.Batini C., Lenzerini M., Navathe S.B. A Comparative Analisis of Methodologies for Database Shema Integration //ACM Computing Survies. – 1986. – V.18. - N 4. – P. 332 - 364.
- 38.Bekke J.H. Semantic data modeling in relation environments: Diss. – Delft, 1991. – VIII, 140 p.
- 39.Borgida A., Mylopolus I., Wong H.K.T. Generalization/Specialization as a Basis for Software Specification // On Conceptual Modelling. – Berlin: Springer – Verlag, 1984. – P.87 – 117.
- 40.Bravaco R.R., Yadav S.B. A Methodology to model the dynamic structure of an organization // Inf. Syst. – 1985. – V.10. - N 3. – P. 299 – 317.
- 41.Brodie M.L. On the development of Data Models // On Conceptual Modelling. – Berlin: Springer – Verlag, 1984. – P. 19 – 47.

42. Brodie M.L., Silva E. Active and Passive Component Modelling: ACM / PCM // Information Systems design methodologies: a comparative review – Amsterdam: North – Holland, 1982. – P. 41 – 92.
43. Burns R.N., Dennis A.R. Selecting the Appropriate application development methodology // Data Base. – 1985. – V.17. – N 1. – P. 19 – 23.
44. Chen P.P. Application of the entity – relationship model // Lect. Notes in Comp. S. – 1982. – V.132. – P.87 – 113.
45. Chen P.P. Preliminary framework for Entity – Relationship models // Entity – Relationship Approach to Systems Analysis and Design. – Amsterdam: North – Holland, 1981. – P. 103 – 119.
46. Chen Q.A Rule – Based objects/task Modelling approach // SIGMOD Record. – 1986. – V.5. – N2. – P.72 – 83.
47. Chiang T.C., Beryeron R.I. A data base management system with an ER conceptual model // Entity-Relationship Approach to System Analysis and Design.- Amsterdam: North-Holland, 1981. - P. 467- 476.
48. Codd E.F. Extending the data base relational model to capture more meaning // ACM Trans. on Database Systems. – 1979. – V.4. – N 4. – P. 397 – 434.
49. Comparison of analysis techniques for information Requirement determination / Yadav S.B., Bravaco R.R., Chatfield A.T., Rajkumar T.M. // Comm. ACM. – 1988. – V. 31. – N 9. – P. 1090 – 1097.
50. Dawson K.S., Parker L.M.P. From Entity-Relationship Diagrams to Forth Normal Form: A Pictorial Aid to Analysis // Computer Journal. – 1988. – V. 31. – N 3. – P. 258 – 268.
51. Elmasri R., Weeldreyer J., Hevnel A. The category concept: An extension to the entity-relationship model // Data and Knowledge Engineering. – 1985. – V. 1. – N 1. – P. 75 – 116.
52. Ellis V. A Refined model for definition as System requirements // Data Base Journal. – 1982. – V. 2. – N 3. – P. 2 – 9.
53. Entity-relationship approach – ER'93: 12th Intern. Conf. on Entity-Relationship Approach. – Berlin: Springer, 1994. – X. 530 p.

54. Feldman D., Miller P. Entity Model Clustering: Structuring a Data Model by Abstraction // *The Computer Journal*. – 1986. – V. – 29. – N 4. – P. 348 – 361.
55. Fitzgerald G., Stokes N., Wood J.R.G. Feature Analysis of Contemporary Information Systems Methodologies // *The Computer Journal*. – 1985. – V. 28. – N 3. – P. 223 – 230.
56. Grodolla M. An extended Entity-Relationship model: Fundamentals and pragmatics. – Berlin et. al.: Springer, 1994. – X, 136 p.
57. Hammer M., Mcleod D. Database Description with SDM: A Semantic Database Model // *ACM Trans on Database Systems*. – 1981. – V.6. – N 3. – P. 351-386.
58. ISO TS 97/SC5/W63 N 095. Concept and terminology for the conceptual schema and information base, 1982.
59. Knuth E., Halasz F. SDLA System Description and Logical Analysis // *Information system design methodologies: comparative review*. – Amsterdam: North-Holland, 1982. – P. 143 – 172.
60. Marti R.W. Integrating Database and Program Description using an ER-Data Dictionary // *Entity-Relationship Approach to Software Engineering*. – Amsterdam: North-Holland, 1983. – P. 797 – 391.
61. Ng A., Paul J.F. A Formal definition of entity-relationship Model // *Entity-Relationship Approach to System Analysis and Design*. – Amsterdam: North-Holland, 1981. – P. 211 – 230.
62. Parrelo B., Overbeek R., Lusk E. The design of entity-relationship models for general lager systems // *Data and Knowledge Engineering*. – 1985. – V. 1. – N 1. – P. 155 – 180.
63. Smith J.M., Smith D.C.P. Principles of data base Conceptual design // *Lect. Notes in Comp Sci*. – 1982. – V. 132. – P. 144 – 146.
64. Zhao L., Roberts S.A. An Object – Oriented Data Model for Data Modelling. Implementation and Access // *The Computer Journal*. – 1998. – V. 31. – N 2. – P. 116 – 124.