МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ Аліна САВЧЕНКО
"____"_____2021 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
"МАГІСТРА"

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ "ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ"

Тема: "Розробка та дослідження технології аналітичної обробки даних корпоративної інформаційної системи"

Виконавець: Олійніченко Андрій Олексійович

Керівник: к.т.н., доцент Куклінський Максим Володимирович

Нормоконтролер: _____ Ігор РАЙЧЕВ

Київ - 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 "Інформаційні технології", 122 "Комп'ютерні науки", "Інформаційні управляючі системи та технології"

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Аліна САВЧЕНКО
«____»_____2021р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Олійніченка Андрія Олексійовича
(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка та дослідження технології аналітичної обробки даних корпоративної інформаційної системи» затверджена наказом ректора від 12.10.2021 за № 2228/ст.

2. Термін виконання роботи: з 11.10.2021 по 28.12.2021.

3. Вихідні дані до роботи: Схема процесу отримання знань із накопичених даних. Архітектура корпоративної інформаційно-аналітичної системи. Діаграми процесу побудови гіперкубу даних. Зображення процесу і результатів роботи програмного модуля аналітичної обробки даних для системи бізнес-аналітики.

4. Зміст пояснювальної записки: вступ, системи підтримки прийняття рішень, побудова сховища даних і OLAP системи, проект системи підтримки прийняття рішень для енергопідприємства.

5. Перелік обов'язкового ілюстративного матеріалу: слайди, презентація.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Підбір і вивчення літературних джерел, дослідження проблеми.	11.10.2021 – 15.10.2021	
2.	Аналіз існуючих технологій бізнес-аналітики	16.10.2021 – 19.10.2021	
3.	Розробка алгоритму проектування програмного комплексу системи бізнес-аналітики	20.10.2021 – 24.10.2021	
4.	Розробка проекту системи бізнес аналітики для енергопідприємства	25.10.2021 – 31.10.2021	
5.	Розробка алгоритму побудови гіперкубу даних	01.11.2021 – 07.11.2021	
6.	Розробка програмного модуля аналітичної обробки даних для системи бізнес-аналітики	08.11.2021 – 17.11.2021	
7.	Оформлення пояснювальної записки і графічного матеріалу дипломної роботи	18.11.2021 – 01.12.2021	
8.	Підготовка доповіді для захисту дипломної роботи	02.12.2021 – 11.12.2021	
9.	Підготовка доповіді для захисту дипломної роботи	12.12.2021 – 28.12.2021	

7. Дата видачі завдання: 11.10.2021р.

Керівник дипломної роботи _____ Максим КУКЛІНСЬКИЙ
(підпис керівника)

Завдання прийняв до виконання _____ Андрій ОЛІЙНІЧЕНКО
(підпис випускниці)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи “Розробка та дослідження технології аналітичної обробки даних корпоративної інформаційної системи” складається із вступу, трьох розділів, загальних висновків, списку бібліографічних посилань використаних джерел і містить 80 сторінок тексту, 2 таблиці, 22 рисунки. Список бібліографічних посилань використаних джерел містить 28 найменувань.

Мета роботи – автоматизація аналітичної обробки корпоративних даних енергопідприємства.

Предмет дослідження – процес формування звітності в корпоративній інформаційно-аналітичній системі.

Об’єкт дослідження – система підтримки прийняття рішень енергопідприємства.

Ключові слова. БІЗНЕС-АНАЛІТИКА, СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА, ОПЕРАТИВНИЙ АНАЛІЗ ДАНИХ, СХОВИЩЕ ДАНИХ, ВІТРИНИ ДАНИХ, КОРПОРАТИВНІ ДАНІ, ГІПЕРКУБ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BI (Business intelligence) - бізнес-аналітика.

ETL (Extraction, Transformation, Loading) - процес вибірки, перетворення і завантаження даних.

HOLAP (Hybrid OLAP) - гібридні OLAP системи.

MOLAP (Multidimensional OLAP) - багатовимірні OLAP системи.

OLAP (On - Line Analytical Processing) - оперативний аналіз даних.

OLTP (Online Transaction Processing) - обробка транзакцій в реальному часі.

ROLAP (Relational OLAP) - реляційні OLAP системи (віртуальні).

SQL(Structured Query Language) - мова структурованих запитів.

АСКОЕ - система автоматизованого комерційного обліку електроенергії.

ВД – вітрина даних

ІАС - інформаційні аналітичні системи.

ІС - інформаційне сховище.

ЛОЗПД - локальне облаштування збору і передачі даних.

ОБД - операційні бази даних. Основа OLTP систем.

САПР — системи автоматизованого проектування.

СППР – системи підтримки прийняття рішення.

СУБД - системи управління базами даних.

СД - сховища даних

ВСТУП

Сучасний рівень розвитку апаратних і програмних засобів протягом деякого часу дозволяв вести бази оперативної інформації повсюдно на різних рівнях управління. У ході своєї діяльності промислові підприємства, корпорації, відомчі структури, органи державної влади та управління збирали великі обсяги даних. Вони несуть в собі великі потенційні можливості для вилучення корисної аналітичної інформації, на основі якої можна виявити приховані тенденції, побудувати стратегію розвитку і знайти нові рішення для широкого спектру аналітичних і управлінських завдань.

Великий обсяг інформації, з одного боку, дозволяє отримувати більш точні розрахунки і аналіз, з іншого боку, перетворює пошук рішень в складну задачу. В результаті з'явився цілий клас програмних систем, призначених для полегшення роботи людей, що виконують аналіз (аналітиків). Такі системи зазвичай називають системами підтримки прийняття рішень СППР (DSS - Decision Support System).

В ідеалі робота аналітиків і менеджерів різних рівнів повинна бути організована таким чином, щоб вони могли:

- мати доступ до всієї інформації, що їх цікавить;
- використовуйте зручні та прості засоби подання та роботи з цією інформацією.

Інформаційні технології, об'єднані під загальною назвою сховище даних (Data Warehouse) і бізнес-аналітика (Business Intelligence), спрямовані на вирішення перерахованих вище проблем.

РОЗДІЛ 1.
СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ

1.1. Аспекти проблеми аналітичного прийняття рішень і їх реалізація в програмних продуктах

Вся проблема аналітичної підготовки прийняття рішень має наступні аспекти:

1. Отримання різнорідних даних, представлених в різних форматах, з багатьох джерел і приведення їх до єдиного формату і структурі, а також організація зберігання і надання користувачам інформації, необхідної для прийняття рішень;
2. Аналіз, в тому числі операційний та інтелектуальний, і підготовка планової або регулярної оцінки стану керованого об'єкта у вигляді паперових документів або екранних форм, підготовка результатів операційного та інтелектуального аналізу для їх ефективного сприйняття споживачами і прийняття на їх ефективного сприйняття споживачами та прийняття на їх основі адекватних рішень.

Аспект, що стосується збору та зберігання інформації з супутнім доопрацюванням, був сформований в концепцію інформаційних сховищ (Data Warehouse). Ця концепція полягає в тому, що інформація про діяльність підприємства або іншого об'єкта економічної чи іншої діяльності накопичується протягом тривалого періоду часу (років) в інформаційному сховищі відповідно до визначених правил. Накопичені дані використовуються в різних часових режимах для аналізу, як джерело даних для різних видів звітності, роботи з партнерами (Reporting) та обґрунтування управлінських рішень.

Аспект проблеми аналізу в силу її великого обсягу і складності має три напрямки:

1. Інформаційно-пошуковий аналіз виконується пошук необхідних даних. Характерною особливістю цього аналізу є виконання заздалегідь зумовлених запитів;

2. Оперативний аналіз даних (On — Line Analytical Processing — OLAP). Основне завдання оперативного або OLAP - аналізу полягає в швидкому (протягом декількох секунд) витяганні, групуванні та узагальненні даних в будь-якій формі, необхідної аналітику або особі, яка приймає рішення (ЛПР), для обґрунтування або прийняття рішення. На відміну від інформаційно-пошукового аналізу, в цьому випадку неможливо заздалегідь спрогнозувати необхідні аналітичні запити;

3. Інтелектуальний аналіз інформації (Data mining, так звана Knowledge Discovery In Data - виявлення знань в даних) призначений для фундаментального дослідження проблем в певній предметній області. Він шукає функціональні і логічні закономірності в накопичених даних, будує моделі і правила, які поясняють знайдені закономірності і/або (з певною ймовірністю) прогнозують розвиток певних процесів. Інтелектуальний аналіз даних знаходиться на перетині декількох наук, основними з яких є система баз даних, статистика і штучний інтелект.

Аспекти проблеми аналізу та функції, необхідні для їх вирішення, виражені у відповідних програмних продуктах. Відповідно, інструменти автоматизації аналізу представлені в різних типах.

Існують складні інформаційно-аналітичні системи, які в тій чи іншій мірі виконують функції відповідно до розглянутих аспектів. На ринку програмного забезпечення також представлені цільові програмні системи, які виконують будь-які функції, такі як оперативний або Інтелектуальний аналіз, в збільшеному обсязі, розширеному складі і підвищеної складності. ІАС інформативно подають системи підтримки прийняття рішень (СППР), і в літературі також використовується абревіатура DSS (Decision Support System).

В цілому, існує ринок інструментів для створення і підтримки OLAP-систем, сховищ інформації (DWH), СППР (DSS), інтелектуального аналізу Data mining (DMg), який отримав узагальнену назву — Business intelligence (BI).

Як правило, всі інструменти, призначені для автоматизації аналітичної роботи, адаптовані для обробки багатовимірних масивів інформації; вони також мають можливість імпорту/експорту даних в інші операційні середовища, розроблені засоби візуального двовимірного (2D) і тривимірного (3D) представлення інформації.

Модулі, призначені для виконання функцій аналізу OLAP, також входять до складу інтегрованих інформаційних систем (ІС) (систем автоматизації робіт, які виконує весь комплекс в інформаційному просторі економічного або будь-якого іншого об'єкта). Найбільш розвинені ІС виконують функції як оперативного, так і інтелектуального аналізу.

Функціональний склад і місце ІАС в системі інформаційних технологій, що використовуються на підприємстві, відображені на рис. 1.1.

Слід зазначити, що ІАС відіграє об'єднуючу роль, об'єднуючи розрізнені ІТ-технології в єдину інтегровану інформаційну систему для управління підприємством (корпорацією), як вона називається ІСУП.

АСУ ТП – автоматизовані системи управління технологічними процесами.

САПР – системи автоматизованого проектування.

ЕСУДО – електронні системи управління документообігом.

ІСУП – інтегровані системи управління підприємством.

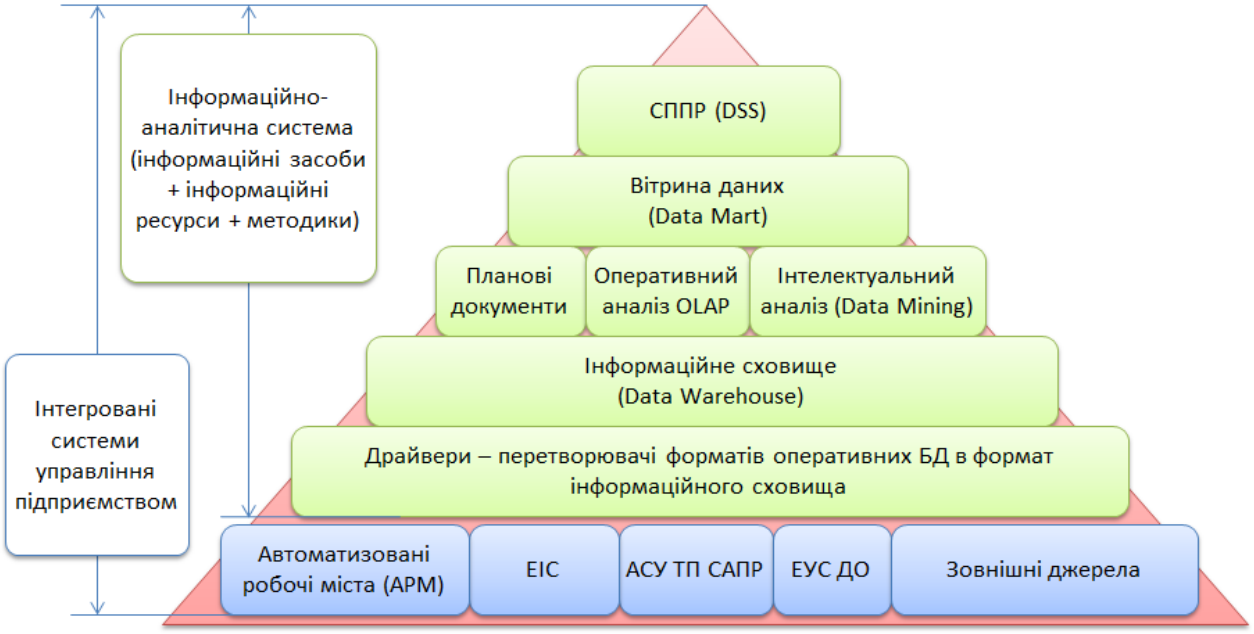


Рис. 1.1 Функціональний склад і місце ІАС в забезпеченні підприємства ІТ - технологіями

1.2. Неефективність використання OLTP-систем для аналізу даних

Практика використання OLTP-систем показала неефективність їх використання для повноцінного аналізу інформації. Такі системи досить успішно вирішують завдання збору, зберігання і пошуку інформації, але вони не відповідають вимогам, що пред'являються до сучасних СППР. Методи, пов'язані з підвищенням функціональності OLTP-систем, не дали задовільних результатів. Основною причиною збою є невідповідність вимог до систем OLTP і СППР. Перелік основних протиріч між цими системами наведено в таблиці 1.1.

Таблиця 1.1.

Порівняльна характеристика вимог до OLTP і OLAP системам.

Характеристика	**Вимоги до OLTP–системи**	**Вимоги до системи аналізу**
Міра деталізації даних, що зберігаються	Зберігання тільки деталізованих даних	Зберігання як деталізованих, так і узагальнених даних
Якість даних	Допускаються невірні дані із-за помилок введення	Не допускаються помилки в даних
Формат зберігання даних	Може містити дані в різних форматах залежно від програм	Єдиний погоджений формат зберігання даних
Допущення надмірних даних	Повинна забезпечуватися максимальна нормалізація	Допускається контрольована денормалізація (надмірність) для ефективного вилучення даних
Характеристика	Вимоги до OLTP–системи	Вимоги до системи аналізу
Управління даними	Має бути можливість у будь-який час додавати, видаляти і змінювати дані	Має бути можливість періодично додавати дані
Кількість даних, що зберігаються	Мають бути доступні усі оперативні дані, що вимагаються в даний момент	Мають бути доступні усі дані, накопичені впродовж тривалого інтервалу часу

Закінчення таблиці 1.1.

Характеристика	Вимоги до OLTP–системи	Вимоги до системи аналізу
Час обробки звернень до даних	Час відгуку системи вимірюється в секундах	Час відгуку системи може складати декілька хвилин
Характер обчислювального навантаження на систему	Постійно середнє завантаження процесора	Завантаження процесора формується тільки при виконанні запиту, але на 100 %
Пріоритетність характеристик системи	Основними пріоритетами являються висока продуктивність і доступність	Пріоритетними є забезпечення гнучкості системи і незалежності роботи користувачів

1.3. Сховище даних

1.3.1. Технології вилучення, перетворення і завантаження даних

Об'єктом аналізу є дані, зосереджені в сховищі, і при необхідності витягнуті безпосередньо з первинних джерел. Вони повинні бути структуровані у вигляді системи «показників» досліджуваної предметної області.

Процеси переміщення і використання даних проходять кілька етапів:

- Етап вилучення, перетворення і завантаження даних (Extraction, Transformation, Loading – ETL процеси). На основі прийнятої системи показників, що характеризують діяльність компанії, організовується збір необхідних даних в сховище і опрацьовуються способи безпосереднього вилучення необхідних докладних даних з первинних джерел; цьому етапу передує робота по створенню необхідної структури даних;

- Етап накопичення, що забезпечує готовність даних до використання. У міру накопичення пам'яті у відповідних зонах пам'яті періодично завантажуються дані з функціональних (транзакційних) підсистем інтегрованої інформаційної системи (ІС) або автономних ІС. Це забезпечує необхідний рівень якості даних; іноді виконується незапланована завантаження даних.
- Етап застосування даних, що містяться в сховищі і витягнутих безпосередньо з первинних джерел. Для забезпечення процесу управління підприємством або іншим об'єктом дані використовуються в трьох основних режимах: створення планової звітності та інших документів (Reporting), оперативний аналіз в незапланованих ситуаціях (OLAP—аналіз), інтелектуальний або поглиблений аналіз (Data mining).

У процесі створення ІАС і її центральної підсистеми-сховища інформації, забезпечення необхідної якості даних, включаючи надійність, узгодженість, дотримання встановлених обмежень і бізнес-правил і так далі, виділяється як важлива проблема.

При зборі даних в інформаційному сховищі необхідно враховувати два основних аспекти: структурний і змістовний.

Структурний аспект полягає в поданні даних з джерел в певних форматах програмних середовищ, в яких вони були згенеровані. Вони повинні бути зведені до одного або групи форматів в системі збору і зберігання даних.

Аспект змісту полягає у змісті знакових структур даних. Навіть при використанні одних і тих же форматів даних можуть існувати різні інтерпретації одного і того ж або аналогічного типу записаних даних і інші типи розбіжностей. Такі ситуації повинні бути виключені ще на етапі формування структури ІС.

Процеси ETL, які реалізують вимоги до забезпечення якості, створення необхідної структури і підтримання характеристик вмісту даних, розділені на наступні етапи:

- висновок. На цьому етапі дані перевантажуються з джерела, як правило, в проміжну область зберігання. Кожне джерело в цій області створює свою власну таблицю. Дані в джерелах можуть мати різні формати, включаючи

неструктурований текст, табличні процесори і різні типи СУБД. Дані одного і того ж типу і структури в первинних джерелах об'єднуються в єдину таблицю, створюючи в ній додаткові поля.

- структурування. Йому доступні тільки неструктуровані дані. Вони зведені до відповідного типу для введення реляційних таблиць.
- обробка. Спочатку структуровані дані або ті, які зазнали структурування, обробляються, що полягає в очищенні, фільтрації і зіставленні даних.
- відправлення та імпорт даних. Сучасні СУБД надають можливість передачі даних як в межах одного сервера, так і в розподіленому режимі між серверами. Цей процес вимагає ретельного, кваліфікованого адміністрування.

Необхідно забезпечити захист передачі даних по каналах зв'язку. Може виявитися, що деякі дані не можуть бути вставлені в призначені таблиці через обмеження або невідповідності типів даних. У таких випадках їм слід призначити окрему область пам'яті, де вони зберігаються для подальшої оцінки.

1.3.2. Сховища даних і аналіз

Концепція СД не є закінченим архітектурним рішенням СППР, і тим більше це не готовий програмний продукт. Мета концепції СД-визначити вимоги до даних, розміщених в СД, загальні принципи і етапи побудови СД, основні джерела даних, а також дати рекомендації щодо вирішення потенційних проблем, що виникають при їх вивантаженні, очищенні, координації, транспортуванні і завантаженні. Концепція СД визначає тільки найзагальніші принципи побудови аналітичної системи і в першу чергу орієнтована на властивості і вимоги до даних, але не на способи їх організації та представлення в цільовій базі даних і режими їх використання.

СД - це концепція побудови аналітичної системи, але не концепція її використання. Це не вирішує жодної з наступних проблем:

- вибір найбільш ефективного способу організації даних для аналізу;
- організація доступу до даних;
- використовуючи технологію аналізу.

Проблеми з використанням зібраних даних вирішуються підсистемами аналізу. Такі підсистеми використовують такі технології:

- регульовані запити;
- аналіз оперативних даних;
- Інтелектуальний аналіз даних.

Якщо регульовані запити успішно застосовувалися задовго до появи концепції СД, то останнім часом оперативний та інтелектуальний аналіз все частіше асоціюється з СД.

1.4. OLAP-системи

У процесі прийняття рішень користувач генерує деякі гіпотези. Щоб перетворити ці гіпотези в закінчені рішення, їх необхідно перевірити. Перевірка гіпотез здійснюється на основі інформації про аналізованої предметної області. Як правило, найбільш зручним способом подання такої інформації для людини є наявність взаємозв'язку між певними параметрами.

OLAP (On-Line Analytical Processing) - це технологія оперативної аналітичної обробки даних, яка використовує методи та інструменти для збору, зберігання та аналізу багатовимірних даних для підтримки процесів прийняття рішень.

Основною метою систем OLAP є підтримка аналітичних дій і довільних (часто використовується термін ad-hoc) запитів від користувачів-аналітиків. Метою аналізу OLAP є перевірка виникаючих гіпотез.

Система OLAP включає в себе два основних компоненти:

- Сервер OLAP-забезпечує зберігання даних, виконання над ними необхідних операцій і формування багатовимірної моделі на концептуальному рівні. Тепер сервери OLAP об'єднані з СД або ВД (вітрина даних);
- Клієнт OLAP-надає користувачеві інтерфейс до багатовимірної моделі даних, надаючи йому можливість зручно маніпулювати даними для виконання завдань аналізу.

Сервери OLAP приховують від кінцевого користувача спосіб реалізації багатовимірної моделі. Вони утворюють гіперкуб, за допомогою якого користувачі використовують OLAP-клієнт для виконання всіх необхідних маніпуляцій, аналізу даних.

1.5. Інтелектуальний аналіз даних(Data Mining)

Для виявлення прихованих знань необхідно застосовувати спеціальні методи автоматичного аналізу, за допомогою яких необхідно практично витягувати знання з «завалів» інформації. Термін Data Mining або Інтелектуальний аналіз даних міцно утвердився в цій області. Класичне визначення цього терміна було дано в 1996 році одним із засновників цього напрямку П'ятецький-Шапіро.

Data Mining - дослідження і виявлення «машиною» (алгоритмами, інструментами штучного інтелекту) в необроблених даних прихованих знань, які раніше не були відомі, нетривіальні, практично корисні і доступні для інтерпретації людиною.

В Data Mining моделі використовуються для представлення отриманих знань. Типи моделей залежать від методів, що використовуються для їх створення. Найбільш поширеними з них є: правила, дерева рішень, кластери і математичні функції.

Основні завдання Data Mining:

- Завдання класифікації зводиться до визначення класу об'єкта на основі його характеристик. Зверніть увагу, що в цьому завданні набір класів, яким може бути присвоєно об'єкт, відомий заздалегідь.
- Завдання регресії, як і завдання класифікації, дозволяє визначити значення певного параметра на основі відомих характеристик об'єкта. На відміну від задачі класифікації, значення параметра являє собою не кінцевий набір класів, а набір дійсних чисел. При пошуку асоціативних правил мета полягає в тому, щоб знайти часті залежності (або асоціації) між об'єктами або подіями. Знайдені залежності представлені у вигляді правил і можуть бути використані як для кращого розуміння природи аналізованих даних, так і для прогнозування появи подій.
- Завдання кластеризації полягає в пошуку незалежних груп (кластерів) і їх характеристик у всьому наборі аналізованих даних. Вирішення цієї проблеми допоможе вам краще зрозуміти дані. Крім того, групування однорідних об'єктів зменшує їх кількість і, отже, полегшує аналіз.

Ці завдання поділяються на описові та прогностичні завдання за їх цільовим призначенням.

Описові завдання спрямовані на поліпшення розуміння аналізованих даних. Ключовим моментом в таких моделях є простота і прозорість результатів для людського сприйняття. Можливо, що виявлені закономірності будуть специфічною особливістю конкретних досліджуваних даних і не будуть знайдені ніде більше, але це все одно може бути корисно і тому має бути відомо. Цей тип завдань включає кластеризацію та пошук асоціативних правил.

Рішення задач прогнозування ділиться на два етапи. На першому етапі модель будується на основі набору даних з відомими результатами. На другому етапі він використовується для прогнозування результатів на основі нових наборів даних. У той же час, звичайно, потрібно, щоб побудовані моделі працювали якомога точніше. Цей тип проблем включає проблеми класифікації та регресії. Сюди також може входити завдання пошуку асоціативних правил,

якщо результати її вирішення можуть бути використані для прогнозування появи певних подій.

Висновки до першого розділу

Системи підтримки прийняття рішень стають все більш і більш життєво важливими для ефективної роботи підприємства. Компоненти СППР дозволяють ефективно і раціонально структурувати, збирати, зберігати і використовувати корпоративні дані підприємства. На практиці СППР являють собою замкнуті системи: Інформація, що є результатом аналізу і виявлення знань, поміщається в Сховище даних і використовується в якості вхідних даних. Таким чином, накопичуються знання і зберігається досвід прийняття управлінських рішень.

В результаті такі системи надають користувачеві єдину точку входу в інформаційний простір підприємства і на основі якісної і стабільної інформації про всі аспекти фінансово-господарської діяльності підприємства допомагають своєчасно приймати управлінські рішення як тактичного, так і стратегічного характеру.

РОЗДІЛ 2.
ПОБУДОВА СХОВИЩА ДАНИХ І OLAP СИСТЕМИ

2.1. Аналіз і обробка корпоративних даних підприємства

Аналіз корпоративних даних включає в себе кілька етапів, які показані на рис. 2.1.

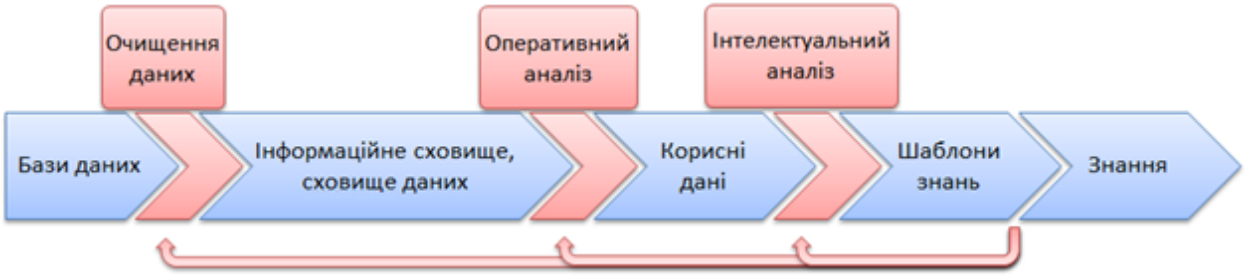


Рис. 2.1 Процес отримання знань з накопичених даних

Процес очищення, накопичення, аналізу та обробки даних називається бізнес-аналітикою (Business intelligence). Цей термін найчастіше відноситься до програмного забезпечення, створеного для того, щоб допомогти менеджеру аналізувати інформацію про свою компанію та її оточення.

2.2. Попередня обробка і очищення даних перед завантаженням в сховище

Абревіатура ETL (extraction, transformation, loading) відноситься до складного процесу передачі даних з однієї програми або автоматизованої інформаційної системи в іншу.

Процес ETL реалізується або шляхом розробки програми ETL, створення набору вбудованих програмних процедур, або за допомогою інструментів ETL. Додатки ETL витягують інформацію з вихідних баз даних, перетворюють її в формат, підтримуваний цільовою базою даних, а потім завантажують перетворені дані в цю базу даних.

Метою будь-якої програми ETL є своєчасна доставка даних із зовнішніх систем в систему, з якою працюють користувачі. Як правило, додатки ETL використовуються при передачі цих зовнішніх джерел в СД систем бізнес-аналітики. Тому організація процесу ETL є невід'ємною частиною проекту розробки практично будь-якого СД.

Процес ETL складається з трьох основних етапів:

- Вивантаження даних. На цьому етапі вибираються і описуються дані із зовнішніх джерел (починають генеруватися метадані СД), які повинні зберігатися в СД (відповідні дані).
- Перетворення даних. На цьому етапі відповідні дані перетворюються в формат представлення СД, правила перетворення зберігаються в метаданих СД, формуються ключові поля таблиць фізичної структури СД і виконується очищення даних.
- Для завантаження даних. На цьому етапі дані завантажуються в СД-карту і будуються агрегати.

Формування метаданих. Метадані - це набір елементів даних і специфікацій, які містять опис даних інформаційної системи і процесів їх обробки.

Процес створення метаданих складається з наступних етапів:

- ідентифікація об'єктів СД та їх атрибутів;
- ідентифікація джерел даних;
- опис семантики цих джерел і СД;
- опис алгоритмів перетворення і агрегування даних;
- опис шляхів доступу до даних тощо

Після визначення інформації для метаданих вона заноситься в спеціальне сховище, а потім використовується для перетворення, очищення і передачі даних.

Очищення даних. Очищення даних включає визначення стандартної фізичної характеристики для кожного елемента даних, його вихідної системи запису і його декодованої, самої базової структури. Необхідно, щоб всі дані в сховищі були еквівалентні за змістом, а не за змістом, його корпоративній системі запису.

Як правило, при створенні сховищ даних дуже мало уваги приділяється очищенню інформації, що надходить в них. Однак, оскільки інформація може бути неоднорідною і надходити з різних джерел, процес очищення є важливою частиною процесу формування сховища.

Мета включення цього механізму в BI полягає в тому, щоб запобігти можливості наступних типів помилок в даних:

- інформаційне протиріччя;
- прогалини в даних;
- аномальні значення;
- помилки при введенні даних.

Суперечність інформації. По-перше, визначаються критерії виявлення протиріч. Наприклад, у випадку з електрикою кожен запис даних має певний код якості, і в один і той же час значення не може бути ні надійним, ні ненадійним.

Тоді є кілька можливих варіантів дій :

1. Якщо ви виявите кілька конфліктуючих записів, видаліть їх.
2. Виправте суперечливі дані. Ви можете розрахувати ймовірність настання кожного з конфліктуючих подій і вибрати найбільш ймовірне з них.
3. Перед записом нового запису в сховище виконайте пошук за вказаними критеріями і поновіть її новими значеннями. Тоді буде збережено тільки поточне значення даних.

Пропуски в даних. Більшість методів прогнозування засновані на припущенні, що дані надходять в рівномірному, постійному потоці. Однак

недосконале технічне оснащення і перебої в зв'язку призводять до прогалин в даних. Щоб виправити такі помилки, ви можете використовувати наступні методи:

1. Наближення. Тобто, якщо в будь-якій точці немає даних, її околиця визначається і обчислюється з використанням формул для апроксимації наближених значень в цій точці, додаючи відповідний запис в сховище. Цей метод може бути ефективно застосований до упорядкованих даних. Однак ви повинні бути відзначені із зазначенням точності розрахованих даних.

2. Визначте дефектну інформацію та повторно запитайте дані. У цьому випадку точність даних буде максимальною, але потрібен час і виправлення технічних проблем на стороні джерела, щоб знову отримати інформацію.

Аномальні значення. Періодично можуть виникати ситуації, коли в сховище надходять дані, різко відрізняються від попередньої інформації. Інструменти прогнозування не ефективні для аналізу таких значень без заздалегідь визначених правил. Такі значення небезпечні при використанні в аналізі OLAP, і особливо при програмному формуванні "знань" з використанням інструментів Data Mining.

Існують методи вирішення цієї проблеми-це надійні оцінки. Ці методи стійкі до сильних збурень. Існуючі дані оцінюються до значень, що перевищують допустимі межі, і застосовується одна з наступних дій:

1. Значення видаляється;

2. Замінено найближчим граничним значенням;

3. Визначено правило для "відсікання" записів, значення яких перевищують допустимі межі.

Помилки введення даних. Ці помилки включають в себе:

- друкарські помилки,
- навмисне спотворення даних,
- невідповідність формату,
- помилки, пов'язані з функціями Введення даних програми

- та інші.

Для виправлення таких помилок визначаються правила введення даних в сховище і визначаються допустимі формати і межі значень. Алгоритми автоматичного виправлення помилок також іноді використовуються для визначення найбільш ймовірних і точного визначення характеру виправлення.

Якщо очищення даних не виконується на етапі завантаження в сховище, ці алгоритми слід включити в аналітичні запити, які створюють звіти BI.

2.3. Концепції організації сховища даних

Сховища виконують завдання накопичення інформації про діяльність підприємства, партнерів та інших інформаційних ресурсів з різних джерел, включаючи бази даних, що відображають окремі бізнес-процеси, автоматизовані робочі місця, інформаційні системи та інші джерела інформації, в тому числі з глобальних інформаційних мереж, таких як інтернет.

За визначенням Білла Інмона Сховище даних – це «орієнтований на предмет, інтегрований, незмінний, хронологічний набір даних, організований для цілей підтримки прийняття рішень».

Предметна орієнтація. У традиційній схемі реалізації інформаційної системи джерелом даних для інструментів аналізу є операційні бази даних (ОБД), а дані орієнтовані на обробку і функціональність систем збору інформації.

У сховищах даних метод зберігання орієнтований на вирішення завдань аналізу та подання даних. Предметна орієнтація-це фундаментальна відмінність між ОБД і СД.

Саме ця властивість дозволяє кінцевому користувачеві працювати з даними, що охоплюють діяльність організації в цілому. Різні програми ОБД можуть описувати одну і ту ж предметну область з різних точок зору, і рішення,

прийняте на основі даних, що відображають тільки одну сторону проблеми, може бути неефективним, а іноді і ненадійним.

Предметна орієнтація також дозволяє значно прискорити доступ до даних за рахунок попередньої структуризації даних під час завантаження і зберігати в сховищі тільки ті дані, які необхідні для інструментів аналізу, що значно знижує вартість носіїв і підвищує безпеку доступу до даних.

Інтеграція. Різні ОБД розробляються різними командами розробників, часто в різний час і з допомогою різних інструментів розробки. Це призводить до того, що об'єкти, що відображають одну і ту ж сутність, мають різні назви і одиниці виміру. Обов'язкова інтеграція даних в СД дозволяє вирішити цю проблему.

Важливість цієї ключової властивості СД може бути продемонстрована в різних аспектах:

- єдині правила найменування об'єктів
- єдині одиниці виміру для об'єктів одного і того ж типу
- єдине фізичне представлення об'єктів одного і того ж типу
- загальні атрибути для представлення об'єктів одного і того ж типу
- та інші.

Це також означає, що дані об'єднуються для задоволення всіх вимог підприємства в цілому, а не однієї бізнес-функції.

Підтримка хронології. Вимоги до ефективності ОБД диктують досить суворі рамки для періоду часу, протягом якого безпосередньо доступні дані. Деякі дані в ОБД взагалі тимчасово не пов'язані. Хронологія даних в різних ОБД може виконуватися по-різному. Наприклад, щоб одне і те ж значення дати в двох ОБД мало різну інтерпретацію.

Сувора і однакова хронологія в СД дозволяє вирішувати всі ці проблеми протягом усього періоду існування даних. В результаті кінцевий користувач завжди має точне і єдине уявлення про тимчасову прив'язку всіх даних.

Незмінність. Дані в ОБД можуть бути додані, видалені і змінені. Дані в СД можна тільки завантажувати і зчитувати. Ця властивість СД дозволяє вирішити дві проблеми:

- після того, як результати, отримані на основі вихідних даних, завжди залишаються актуальними;
- підвищити швидкість доступу до даних.

Структура інформаційного сховища. Таким чином, ідея сховищ даних - це не просто єдиний підхід до зберігання необхідних даних, а створення єдиного багатопрофільного інформаційного ресурсу підприємства, напрямку досліджень, корпоративної структури і так далі в рамках однієї концептуальної ідеї

Дані в сховищі з джерел накопичуються протягом певного періоду часу в зоні накопичення. Протягом цього часу проводиться робота по забезпеченню необхідної якості даних відповідно до правил, описаних вище. У процесі перекачування даних з джерел вони перетворюються в єдиний формат, перевіряється їх семантична узгодженість, перевіряються помилки і вживаються заходи щодо підвищення якості даних. Після досягнення необхідного рівня якості і часу, зазначеного графіком роботи, дані передаються в область зберігання (рис. 2.2).

Рис. 2.2 Структура інформаційного сховища

В області зберігання вони можуть бути представлені у вигляді реляційної або багатовимірної моделі (представлення об'єкта). При використанні реляційної моделі необхідно мати зону представлення об'єктів у сховищі, щоб досягти рівня системних характеристик, що відповідає вимогам до систем OLAP.

2.4. Концепція OLAP

Концепція OLAP заснована на принципі багатовимірного представлення даних. У 1993 році Е.Ф.Кодд у статті «Забезпечуючи аналітиків технологією OLAP» розглянув недоліки реляційної моделі, в першу чергу вказавши на неможливість «об'єднувати, переглядати і аналізувати дані з точки зору множинності вимірів, тобто найзрозумілішим для корпоративних аналітиків способом», і визначив загальні вимоги до систем OLAP, які розширюють функціональність реляційних СУБД і включають багатовимірний аналіз однією з їх характеристик.

2.4.1. Вимоги, що пред'являються до OLAP -системам

Едвард Кодд визначив 12 правил, яким повинен відповідати програмний продукт класу OLAP.

Таблиця 2.1.

Правила оцінки OLAP системи.

№	Правило	Опис
1.	Багатовимірне концептуальне представлення даних (Multi - Dimensional	Концептуальне представлення моделі даних в продукті OLAP має бути багатовимірним за своєю природою, тобто дозволяти аналітикам виконувати інтуїтивні операції "аналізу вздовж і поперек" ("slice and dice"),

	Conceptual View)	обертання (rotate) і розміщення (pivot) напрямів консолідації.
2.	Прозорість (Transparency)	Користувач не повинен знати про те, які конкретні засоби використовуються для зберігання і обробки даних, як дані організовані і звідки беруться.
3.	Доступність (Accessibility)	Аналітик повинен мати можливість виконувати аналіз у рамках загальної концептуальної схеми, але при цьому дані можуть залишатися під управлінням СУБД, що залишилися від старого спадку, будучи при цьому прив'язаними до загальної аналітичної моделі. Тобто інструментарій OLAP повинен накладати свою логічну схему на фізичні масиви даних, виконуючи усі перетворення, що вимагаються для забезпечення єдиного, погодженого і цілісного погляду користувача на інформацію.

Продовження таблиці 2.1.

№	Правило	Опис
4.	Стійка продуктивність (Consistent Reporting Performance)	Зі збільшенням числа вимірів і розмірів бази даних аналітики не повинні зіткнутися з яким би то не було зменшенням продуктивності. Стійка продуктивність потрібна для підтримки простоти використання і свободи від ускладнень, які потрібно для доведення OLAP до кінцевого користувача.

5.	Клієнт - серверна архітектура (Client - Server Architecture)	Велика частина даних, що вимагають оперативної аналітичної обробки, зберігається в мэйнфреймовых системах, а витягається з персональних комп'ютерів. Тому однією з вимог є здатність продуктів OLAP працювати в середовищі клієнт-сервер. Головною ідеєю тут є те, що серверний компонент інструменту OLAP має бути досить інтелектуальним і мати здатність будувати загальну концептуальну схему на основі узагальнення і консолідації різних логічних і фізичних схем корпоративних баз даних для забезпечення ефекту прозорості.
6.	Рівноправ'я вимірів (Generic Dimensionality)	Усі виміри даних мають бути рівноправні. Додаткові характеристики можуть бути надані окремим вимірам, але оскільки усі вони симетричні, ця додаткова функціональність може бути надана будь-якому виміру. Базова структура даних, формули і формати звітів не повинні спиратися на якийсь один вимір.

Продовження таблиці 2.1.

№	Правило	Опис
7.	Динамічна обробка розріджених матриць (Dynamic Sparse	Інструмент OLAP повинен забезпечувати оптимальну обробку розріджених матриць. Швидкість доступу повинна зберігатися незалежно від розташування осередків даних і бути постійною величиною для

	Matrix Handling)	моделей, що мають різне число вимірів і різну розрідженість даних.
8.	Підтримка розрахованого на багато користувачів режиму (Multi - User Support)	Частенько декілька аналітиків мають необхідність працювати одночасно з однією аналітичною моделлю або створювати різні моделі на основі одних корпоративних даних. Інструмент OLAP повинен надавати їм конкурентний доступ, забезпечувати цілісність і захист даних.
9.	Необмежена підтримка кроссмерных операцій (Unrestricted Cross - dimensional Operations)	Обчислення і маніпуляція даними по будь-якому числу вимірів не повинні забороняти або обмежувати будь-які стосунки між осередками даних. Перетворення, що вимагають довільного визначення, повинні задаватися на функціонально повній формульном мові.
10.	Інтуїтивне маніпулювання даними (Intuitive Data Manipulation)	Переорієнтація напрямів консолідації, деталізація даних в колонках і рядках, агрегація і інші маніпуляції, властиві структурі ієрархії напрямів консолідації, повинні виконуватися в максимально зручному, природному і комфортному призначеному для користувача інтерфейсі.

Закінчення таблиці 2.1.

№	Правило	Опис

11.	Гнучкий механізм генерації звітів (Flexible Reporting)	Повинні підтримуватися різні способи візуалізації даних, тобто звіти повинні представлятися у будь-якій можливій орієнтації.
12.	Необмежена кількість вимірів і рівнів агрегації (Unlimited Dimensions and Aggregation Levels)	Настійно рекомендується допущення в кожному серйозному OLAP інструменті як мінімум п'ятнадцяти, а краще двадцяти, вимірів в аналітичній моделі. Більше того, кожен з цих вимірів повинен допускати практично необмежену кількість визначених користувачем рівнів агрегації по будь-якому напряму консолідації.

У 1995 році на основі вимог, викладених Е.Ф.Коддом, був сформульований так званий тест FASMI (Fast Analysis of Shared Multidimensional Information) що включає наступні вимоги до додатків для багатовимірного аналізу:

- надавати користувачеві результати аналізу в прийнятний час (зазвичай не більше 5 секунд), навіть ціною менш докладного аналізу;
- можливість виконувати будь-який логічний і статистичний аналіз, специфічний для даного додатка, і зберігати його у формі, доступній кінцевому користувачеві;
- багатокористувацький доступ до даних з підтримкою відповідних механізмів блокування та інструментів авторизованого доступу;
- багатовимірне концептуальне представлення даних, включаючи повну підтримку ієрархій і множинних ієрархій (це ключова вимога OLAP);
- можливість доступу до будь-якої необхідної інформації, незалежно від її обсягу і місця зберігання.

Слід зазначити, що функціональність OLAP може бути реалізована різними способами, починаючи з простих інструментів аналізу даних в офісних додатках і закінчуючи розподіленими аналітичними системами на основі серверних продуктів.

2.4.2. OLAP-системи

В даний час на ринку існує велика кількість продуктів, які в тій чи іншій мірі забезпечують функціональність OLAP. Надаючи багатовимірне концептуальне уявлення від інтерфейсу користувача до вихідної бази даних, всі продукти OLAP розділені на три класи відповідно до типу вихідної БД.

1. Найперші операційні системи аналітичної обробки (наприклад, Essbase компанії Arbor Software, Oracle Express Server компанії Oracle) відносилися до класу MOLAP, тобто могли працювати тільки зі своїми власними багатовимірними базами даних. Вони засновані на запатентованих технологіях для багатовимірних СУБД і Є найдорожчими. Ці системи забезпечують повний цикл обробки OLAP. Вони або включають в себе, на додаток до серверного компоненту, власний інтегрований клієнтський інтерфейс, або використовують зовнішні програми для роботи з електронними таблицями для зв'язку з користувачем. Обслуговування таких систем вимагає спеціального штату співробітників, що беруть участь в установці, обслуговуванні системи і формуванні уявлень даних для кінцевих користувачів.

2. Операційні системи аналітичної обробки реляційних даних (ROLAP) дозволяють представляти дані, що зберігаються в реляційній базі даних, в багатовимірній формі, забезпечуючи перетворення інформації в багатовимірну модель через проміжний шар метаданих. Цей клас включає в себе набір DSS від MicroStrategy, Metacube від Informix, DecisionSuite від information Advantage та інші. Системи ROLAP добре адаптовані для роботи з великими складськими приміщеннями. Як і системи MOLAP, вони вимагають значних витрат на

технічне обслуговування фахівцями з інформаційних технологій і забезпечують багатокористувацький режим роботи.

3. Нарешті, гібридні системи (hybrid OLAP, HOLAP) призначені для об'єднання переваг і мінімізації недоліків, властивих попереднім класам. Цей клас включає в себе програмне забезпечення Media/MR компанії Speedware. За словами розробників, він поєднує в собі аналітичну гнучкість і швидкість відгуку MOLAP з постійним доступом до реальних даних, властивим ROLAP.

На додаток до цих інструментів існує ще один клас-інструменти для створення запитів і звітів для настільних ПК, які доповнені функціями OLAP або інтегровані з зовнішніми інструментами, що виконують такі функції. Ці добре розвинені системи відбирають дані з вихідних джерел, перетворюють їх і поміщають в динамічну багатовимірну базу даних, яка працює на клієнтській станції кінцевого користувача. Основними представниками цього класу є однойменні бізнес-об'єкти, BrioQuery від Brio Technology і Powerplay від Cognos.

Багатовимірний OLAP (MOLAP). У спеціалізованих СУБД, заснованих на багатовимірному поданні даних, дані організовані не у вигляді реляційних таблиць, а у вигляді впорядкованих багатовимірних масивів:

1. Гіперкуби (всі осередки, що зберігаються в базі даних, повинні мати однаковий вимір, тобто перебувати в найбільш повній базі вимірювань)

2. Полікуби (кожна змінна зберігається зі своїм власним набором вимірювань, і всі пов'язані з цим труднощі обробки передаються внутрішнім механізмам системи).

Використання багатовимірних баз даних в системах оперативної аналітичної обробки має наступні переваги.

1. У разі використання багатовимірної СУБД пошук і вибірка даних виконуються набагато швидше, ніж при багатовимірному концептуальному поданні реляційної бази даних, оскільки багатовимірна база даних денормалізована, містить попередньо агреговані показники і забезпечує оптимізований доступ до зшитих осередків.

2. Багатовимірні СУБД легко справляються із завданнями включення різних вбудованих функцій в інформаційну модель, в той час як об'єктивно існуючі обмеження мови SQL роблять виконання цих завдань на основі реляційних СУБД досить складним, а іноді і неможливим.

З іншого боку, існують істотні обмеження.

1. Багатовимірні СУБД не дозволяють працювати з великими базами даних. Крім того, через денормалізацію і попередньо виконаної агрегації обсяг даних в багатовимірній базі даних зазвичай відповідає (за оцінками Кодда) в 2,5-100 разів менше, ніж вихідні докладні дані.

2. Багатовимірні СУБД використовують зовнішню пам'ять дуже неефективно в порівнянні з реляційними. У переважній більшості випадків інформаційний гіперкуб дуже розріджений, і оскільки дані зберігаються в упорядкованій формі, невизначені значення можуть бути видалені тільки шляхом вибору оптимального порядку сортування, що дозволяє організувати дані в максимально можливі безперервні групи. Але навіть в цьому випадку проблема вирішена лише частково. Крім того, оптимальний порядок сортування з точки зору зберігання розріджених даних, швидше за все, не буде відповідати порядку, який найчастіше використовується в запитах. Тому в реальних системах доводиться знаходити компроміс між продуктивністю і надмірністю дискового простору, займаного базою даних.

Тому використання багатовимірних СУБД виправдано тільки при наступних умовах.

1. Обсяг вихідних даних для аналізу не надто великий (не більше декількох гігабайт), тобто рівень агрегації даних досить високий.

2. Набір інформаційних вимірювань стабільний (оскільки будь-яка зміна їх структури майже завжди вимагає повної перебудови гіперкуба).

3. Час відгуку системи на нерегульовані запити є найбільш важливим параметром.

4. Для виконання багатовимірних обчислень в осередках гіперкуба потрібне широке використання складних вбудованих функцій, включаючи можливість написання користувацьких функцій.

Реляційний OLAP (ROLAP). Пряме використання реляційних баз даних в операційних системах аналітичної обробки має наступні переваги.

1. У більшості випадків корпоративні сховища даних реалізуються з використанням інструментів реляційної СУБД, а інструменти ROLAP дозволяють виконувати аналіз безпосередньо над ними. Однак розмір сховища не є таким критичним параметром, як у випадку з MOLAP.

2. У разі проблеми зі змінним розміром, коли зміни в структурі вимірювань доводиться вносити досить часто, системи ROLAP з динамічним поданням розмірів є оптимальним рішенням, оскільки такі модифікації не вимагають фізичної реорганізації бази даних.

3. Реляційні СУБД забезпечують значно вищий рівень захисту даних і хороші можливості розмежування прав доступу.

Основним недоліком ROLAP в порівнянні з багатовимірними СУБД є більш низька продуктивність. Для забезпечення продуктивності, порівнянної з OLAP, реляційні системи вимагають ретельного вивчення схеми бази даних і конфігурації індексу, тобто великих зусиль з боку адміністраторів баз даних. Тільки при використанні схем у формі зірки продуктивність добре зарекомендували себе реляційних систем може бути ближче до продуктивності систем, заснованих на багатовимірних базах даних.

2.5. Куби даних OLAP

Термін «Куб» використовується для опису багатовимірного простору даних, хоча це не куб в геометричному сенсі слова. Геометричний куб має тільки три виміри. Багатовимірний простір даних може мати будь-яку кількість вимірювань, і ці вимірювання не обов'язково повинні бути однакового (або навіть схожого) розміру.

Одна з найважливіших відмінностей між геометричним простором та багатовимірним простором даних полягає в тому, що геометричний сегмент складається з нескінченного числа точок, тоді як багатовимірний простір є дискретним і містить дискретне число значень у кожному вимірі.

2.5.1. Опис багатовимірного простору

Нижче наведено визначення термінів, що використовуються для опису багатовимірного простору:

- Вимір (Dimension) – описує елемент даних, який використовується для аналізу. Наприклад, досить поширеним елементом аналізу є час;
- Елемент (Member) – відповідає одній точці вимірювання. Наприклад, у вимірі часу (Time) понеділок буде елементом вимірювання;
- Значення Елементу (Member Value) є унікальною характеристикою елемента. Наприклад, у вимірі часу (Time) це може бути певна дата;
- Атрибут (Attribute) – це повна колекція елементів одного і того ж типу. Наприклад, всі дні тижня будуть атрибутом вимірювання часу (Time);
- Розмір (Size), або Кардинальність (Cardinality) виміру – це кількість елементів, які воно містить. Наприклад, вимірювання часу (Time), що складається з днів тижня, матиме розмір 7.

Щоб проілюструвати ці терміни, Я наведу опис тривимірного простору управління живленням. На рис. 2.1 показані три вимірювання:

- час в місяцях (1);
- напрям і вид електроенергії (Актив прийом (А+), Актив віддача(А-), Реактив прийом (R+), Реактив віддача (R+)) (2);
- енергопідприємства, що входять в енергоуправління (3)

Ці три вимірювання використовуються для визначення обсягу електроенергії, виробленої і споживаної Херсонською енергетичною адміністрацією за період часу, вимірюваний в місяцях.

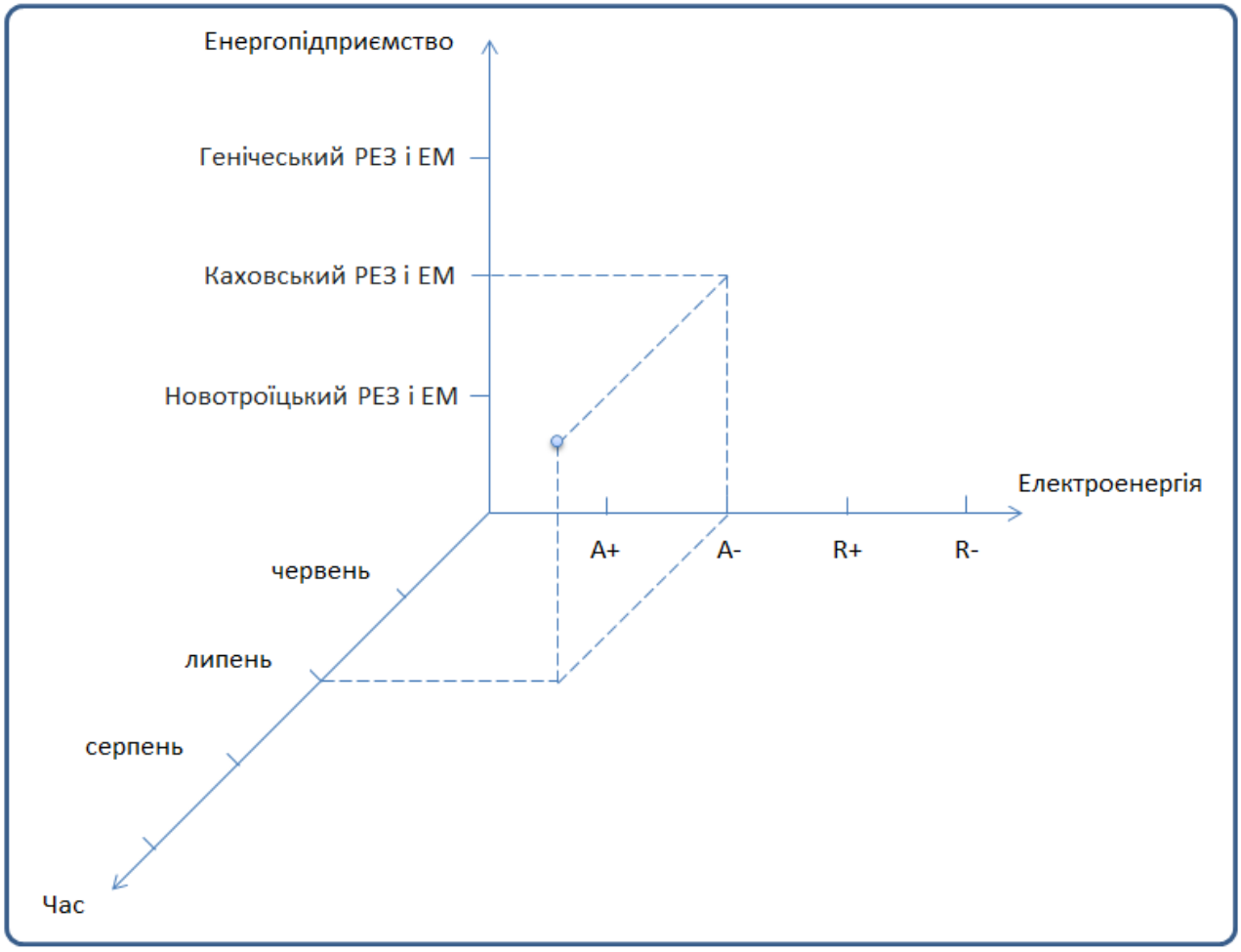


Рис. 2.3 Простір даних по електроенергії Херсонського обленерго.

На рис. 2.3 показано лише одне значення електроенергії, показане точкою в просторі даних. Якщо кожне значення енергії в певний момент часу представлено у вигляді точки в багатовимірному просторі, то всі ці точки утворюють фактичний простір даних або фактичні дані.

Природно, фактична кількість виробленої або споживаної енергії відрізняється від можливої кількості. Таким чином, кількість точок, що відповідає можливій кількості електроенергії, формує теоретичний простір даних. Розмір теоретичного простору математично визначається шляхом множення розмірів всіх вимірювань. Якщо існує велика кількість вимірювань, теоретичний простір може бути дуже великим, але незалежно від його розміру він залишається обмеженим, оскільки кожне вимірювання дискретно і обмежено своїм власним набором елементів.

Наступний список визначає кілька інших поширених термінів, що використовуються для опису багатовимірного простору :

- Кортеж (Tuple) — це координата в багатовимірному просторі;
- Зріз (Slice) — це ділянка багатовимірного простору, який може бути визначений кортежем.

Кожна точка в геометричному просторі визначається набором координат. у тривимірному просторі це X, Y і Z. Як і геометричний простір, багатовимірний простір визначається набором координат. Цей набір називається кортежем. Кортеж відіграє важливу роль у визначенні багатовимірних даних і при маніпулюванні ними.

Наприклад, точка в просторі, показана на рис. 2.4, визначається кортежем ([Актив віддача], [Каховський РЕЗ і ЕМ], [Липень]). Якщо елемент в одному або декількох вимірах в кортежі замінюється зірочкою «*», яка грає роль підстановочного знака і вказує на всі елементи цього вимірювання, Ви отримуєте підпростір (фактично звичайний підпростір). Цей тип нормального підпростору називається зрізом.

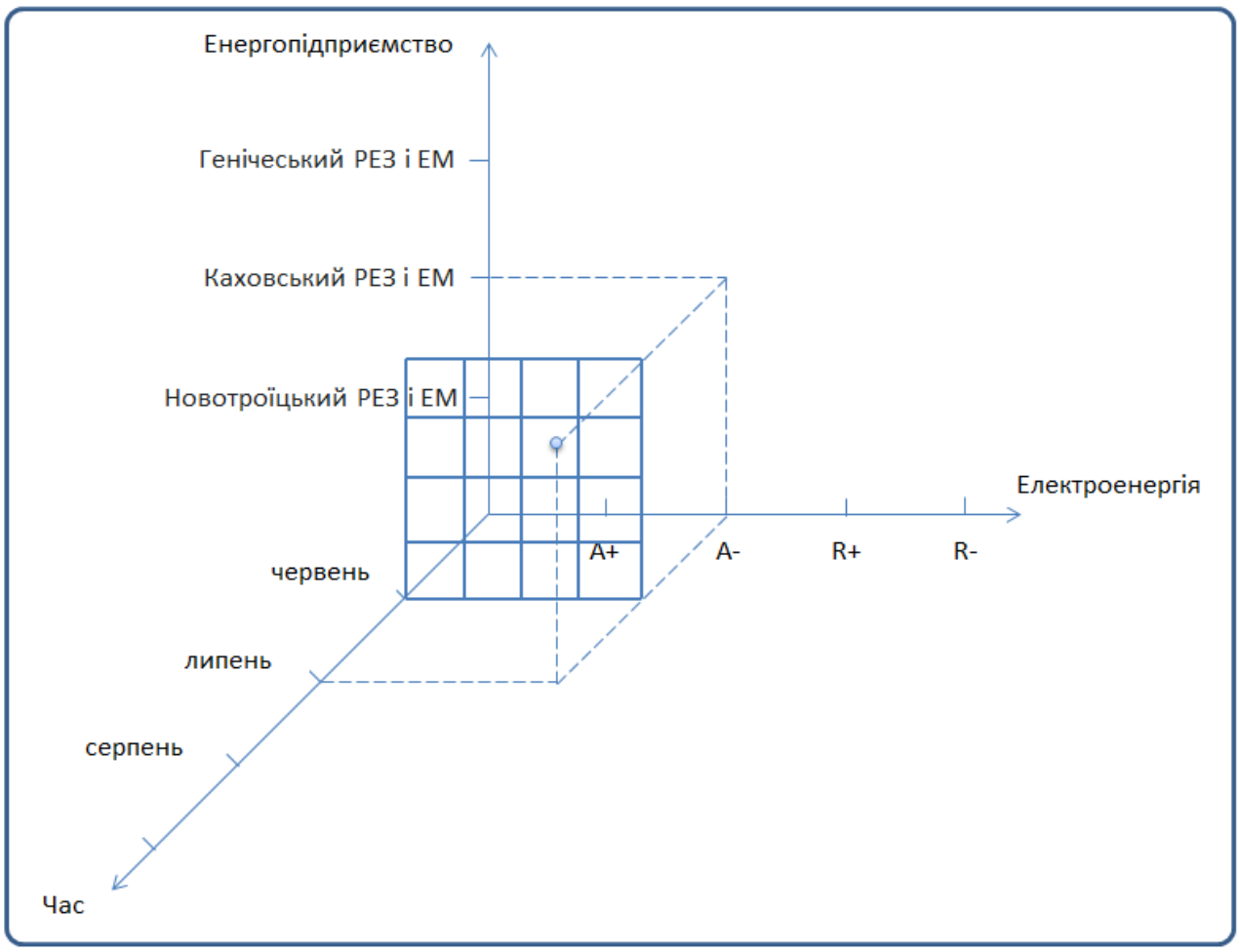


Рис. 2.4 Зріз даних за червень (*, [Червень]).

2.5.2. Атрибути вимірів

Але як визначити простір споживання електроенергії по кварталах, а не по місяцях. Якщо для вимірювання часу використовується один атрибут (місяці), то необхідно вручну згрупувати місяці по кварталах. Якщо це розглядається протягом декількох років, ручне угруповання стає незручним.

У цьому випадку вам потрібен якийсь спосіб візуалізації місяців, кварталів і років (і будь-яких інших типів поділу часу, можливо, днів) по відношенню один до одного-аналогічно шкалі на лінійці, яка дозволяє відображати поділ розміру на метри, сантиметри і міліметри.

Насправді вам потрібні додаткові атрибути-квартали, роки і т.д. Тепер можна використовувати місяці в якості ключового атрибута і пов'язувати інші

атрибути (залежні атрибути, пов'язані атрибути) з місяцями — 3 місяці в кварталі, 12 в році.

Наприклад, вам потрібно показувати окремі місяці в кожному кварталі і в кожному році. Для цього вам потрібно додати два залежних атрибута в вимір часу (квартал і рік) і створити зв'язок між цими атрибутами і ключовим атрибутом. Тепер ви можете створити шкалу, як показано на рис. 2.5, для вимірювання року-квартал-місяць.

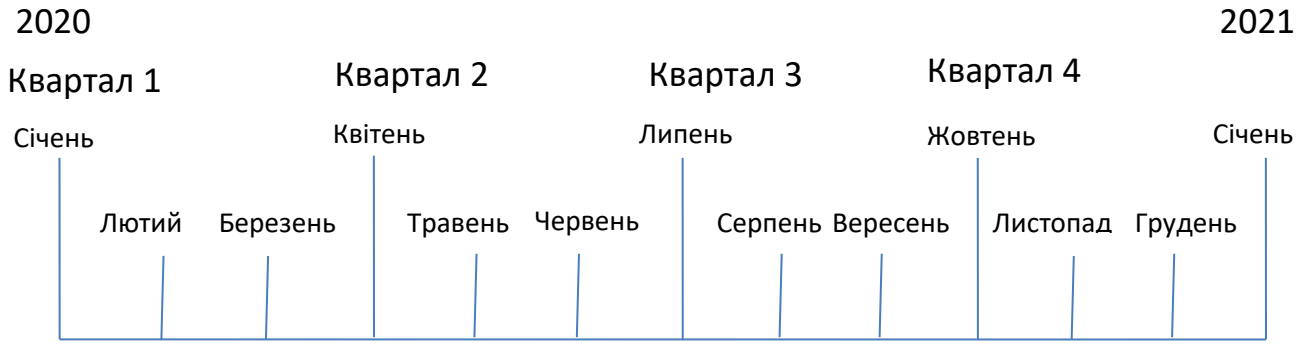


Рис. 2.5 Залежних атрибутів (рік, квартал) відкалібровані відносно ключового атрибуту (місяць)

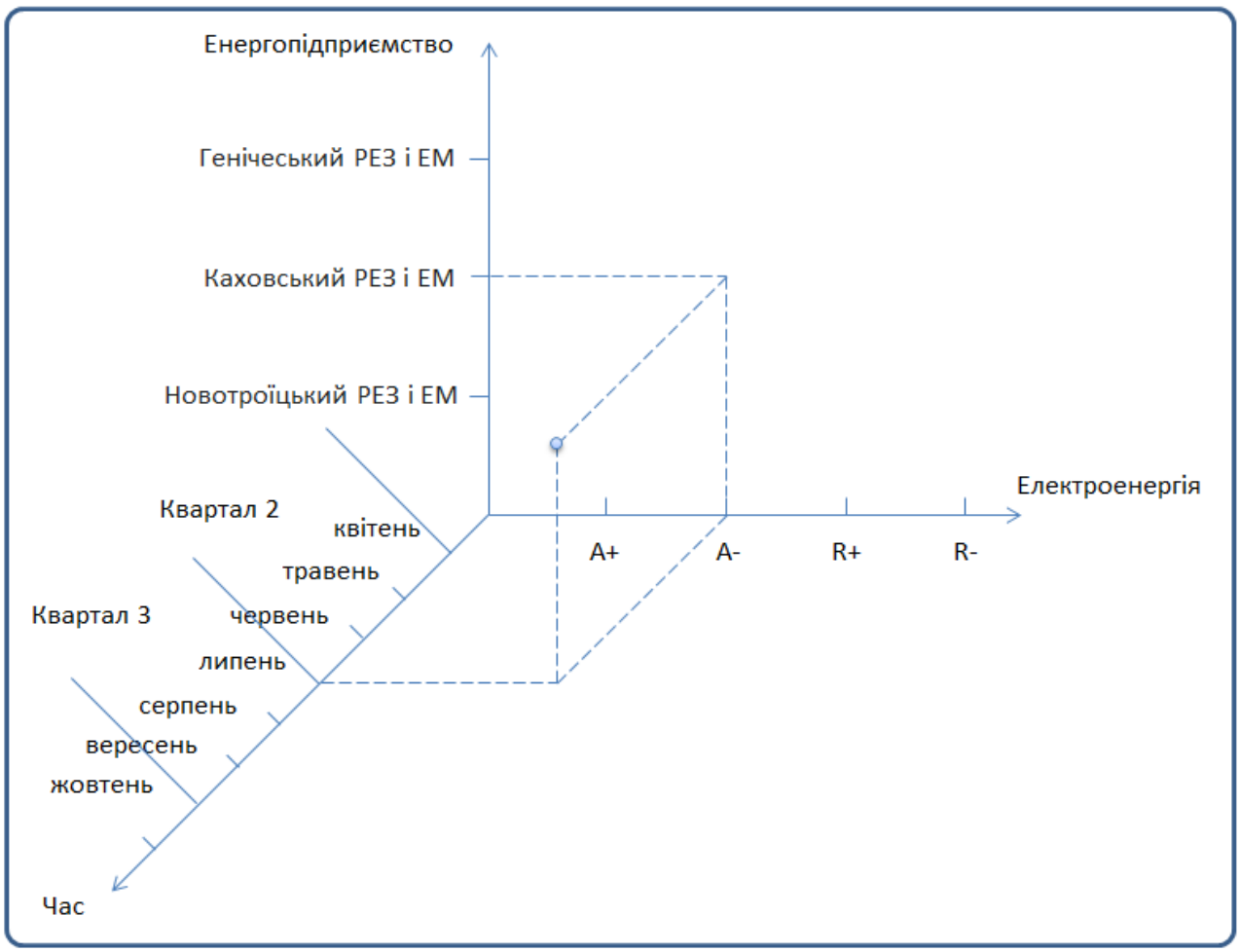


Рис. 2.6 Залежних атрибутів створюють нові точки у багатовимірному просторі

2.5.3. Осередки

Після додавання шкали до вимірювання багатовимірного простору (рис. 2.6) на вимірі з'явилися нові позиції, відповідні елементам залежних атрибутів (квартал, рік).

Ці елементи, в свою чергу, створюють безліч нових точок в багатовимірному просторі. Але для цих нових точок немає значень, тому що дані, введені в базу даних, містили тільки значення по місяцях. Показання для цих точок можуть бути розраховані тільки на основі значень, встановлених фактичними даними.

На цьому етапі з'являється новий простір даних-логічний простір, який, на відміну від простору фактів, містить тільки точки, що представляють реальні

значення, містить точки, які можна обчислити. Повний набір точок у просторі, що об'єднує фактичний і логічний простір, називається багатовимірною моделлю або багатовимірним кубом, який скоріше є багатовимірним гіперкубом, а точки в просторі куба називаються осередками.

2.5.4. Міри

Значення в комірці того ж типу називаються мірою. Міра-це значення, що описує показання в осередку. Наприклад, електрика може вимірюватися в кВт або МВт. Іншим заходом може бути вартість 1 кВт в певний час доби (за відповідним тарифом).

Ці заходи разом складають вимір міри. Кожен елемент цього вимірювання (заходи) має набір властивостей, таких як тип даних, одиниця виміру і, що найбільш важливо, тип обчислення для агрегатної функції.

2.5.5. Функції агрегації

Тип обчислення – це зв'язок, який з'єднує фізичний і логічний простір куба. Функція агрегування даних дозволяє обчислювати значення осередків в логічному просторі за значеннями осередків в реальному просторі.

Функція агрегування може бути як простою-адитивною, так і складною – напівадитивною. Список функцій адитивної агрегації досить обмежений-сума даних (SUM), мінімальні (MIN) і максимальні (MAX) значення даних і обчислення кількості (COUNT), яке по суті є просто сумою. Всі інші функції є складними і використовують складні формули і алгоритми.

2.5.6. Елемент All

На відміну від геометричного простору, в якому відправною точкою відліку є точка, в якій всі координати рівні 0, початкову точку для

багатовимірного простору визначити складніше. Наприклад, немає значення 0 для вимірювання часу по місяцях, а Січень-це тільки перший місяць. Тому вам необхідно задати початок багатовимірного простору, використовуючи спеціальний атрибут, який об'єднує всі елементи вимірювання. Цей атрибут містить тільки один елемент - все. Для простих функцій агрегування, таких як суми, елемент All еквівалентний сумі значень всіх елементів фактичного простору. для складних функцій агрегування елемент all обчислюється за формулою, пов'язаною з функцією.

Висновки до другого розділу

Таким чином, програмний пакет BI включає інструменти для інтеграції та очищення даних (ETL), сховища аналітичних даних, обробки та аналізу даних (OLAP-технонолонія), інструменти Data Mining, автоматичне формування звітів, а також механізми візуального представлення інформації та отриманих «знань».

Система двотипного типу повинна складатися з безлічі програмних модулів, які забезпечать повний цикл роботи з даними: від отримання з фізичних джерел і зберігання їх у певній структурованій формі до інтелектуального аналізу інформації та формування на її основі «знань». Системні модулі повинні бути незалежними один від одного з точки зору реалізації, оскільки вони можуть бути встановлені на різних апаратних обчислювальних пристроях.

РОЗДІЛ 3.
ПРОЕКТ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ ДЛЯ ЕНЕРГОПІДПРИЄМСТВА

3.1. Архітектура підприємства

На підприємстві енергетичного ринку інформаційно-аналітичною системою повинна бути автоматизована комерційна система обліку електроенергії (АСКУЄ). Така система АСКУЄ являє собою багаторівневий повний набір апаратних і програмних засобів для автоматизації процесу збору та обробки інформації з комерційного та технічного обліку електричної енергії і потужності на базі електронних багатофункціональних лічильників електричної енергії, а також пристроїв збору, обробки, зберігання, відображення, прийому і передачі інформації (рис. 3.1).

Завдання АСКУЄ:

- вимірювання, збирання, обробка, накопичення, відображення, документування та розповсюдження надійної, захищеної та легалізованої інформації про вироблену, передану, розподілену та вивільнену електричну енергію та потужність;
- контроль основних показників якості електроенергії;
- ведення архівів виміряних значень енергії, потужності і показників якості електричної енергії заданої дискретності і для заданої ретроспективи;
- обробка даних і звітність;
- надання інформації з обліку енергії зацікавленим користувачам;
- моніторинг та діагностика технічного стану підсистем бухгалтерського обліку.

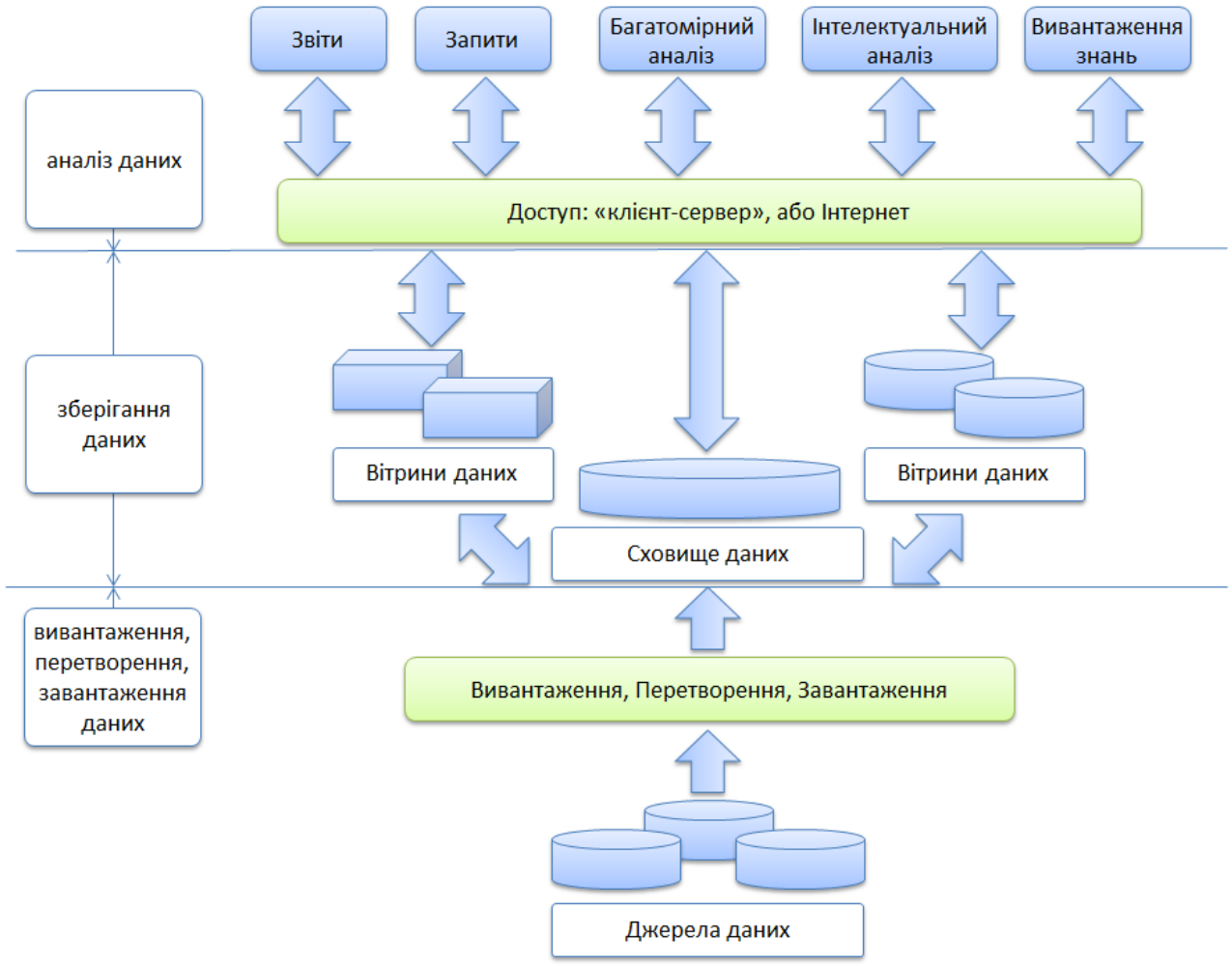


Рис. 3.1 Архітектура корпоративної інформаційно-аналітичної системи

3.2. Джерела даних

Бази даних цих внутрішніх транзакційних систем(рівень підстанції), інформаційних систем підпорядкованих організацій і дані, що надходять від зовнішніх організацій (пов'язаних ліцензіатів), можуть використовуватися в якості джерел інформації для зберігання.

З організаційної точки зору цей рівень включає підрозділи і структури організації на всіх рівнях, які підтримують бази даних оперативного доступу. Це низовий рівень генерації інформації, рівень внутрішніх і зовнішніх джерел інформації, які виробляють «сиру» інформацію. Ця інформація є первинною і використовується для щоденної звітності енергетичної компанії.

З системно-технічної точки зору цей шар представлений обчислювальними системами всіх підрозділів всіх рівнів, до яких підключені спеціалізовані технічні комплекси, що зберігають інформацію. Такі технічні комплекси можуть включати:

1. Реляційні (орієнтовані на SQL) сервери баз даних на базі комп'ютерів під управлінням Windows NT, Unix та інші.
2. Файлові сервери, на яких встановлена будь-яка система обробки даних (наприклад, Btrieve) або мережева версія персональної СУБД (наприклад, Paradox, FoxPro і так далі).
3. Персональні комп'ютери з локальними персональними базами даних або файлами.

На рівні підстанції збирається і зберігається первинна інформація з лічильників електроенергії (графіки навантаження, поточні показання, журнали подій лічильників і т.д.), обробляється і передається на верхні рівні АСКУЄ. При необхідності локальне клієнтське місце розташування оператора може бути організовано на базі персонального комп'ютера.

В якості локального облаштування збору і передачі даних (ЛОЗПД) може бути використано пристрій на базі промислового контролера серійного виробництва, такому як Think Core, RISC, e-Box, Моха і т.д. зі стандартною операційною системою (Windows CE, Linux або Unix) і спеціальним програмним забезпеченням для реалізації функцій сервера зв'язку і сервера зберігання даних може використовуватися в якості локального пристрою збору і передачі даних. Цей пристрій забезпечує в автоматичному режимі:

- збір інформації з лічильників електроенергії на базі спеціалізованих мікропроцесорів через цифровий інтерфейс (RS-485, RS-232, ИПРС, Ethernet), використовуючи протоколи різних типів лічильників, встановлених на підстанції;
- передача даних за запитом на верхній рівень або безпосередньо в центр збору та обробки даних;

• синхронізація лічильників часу. Як джерело часу використовується GPS-приймач (наприклад, GARMIN). Метрологічна атестація ЛОЗПД проводиться в рамках метрологічної атестації місцевого комплексу АСКУЄ.

ЛОЗПД налаштовується з персонального комп'ютера (через оптопару) або за допомогою вбудованої клавіатури і плати дисплея.

Організація інформаційної мережі може здійснюватися з використанням комутаторів локальної мережі, серверів з асинхронним портом (N-Port).

Для організації каналів зв'язку з верхніми рівнями можна використовувати модеми (провідні 2/4, комутовані, Радіо, GSM / GPRS) або бездротові точки доступу (технології WiMAX, Wi-Fi). Для каналів зв'язку має бути передбачено резервування. В якості резервного каналу передбачається використовувати GSM-канал.

Вимога щодо забезпечення безперебійного живлення виконується при використанні стандартних джерел резервного живлення від батарей.

У місцях розташування клієнтів в якості операційної системи надається Windows XP або Windows Vista, а «пакет програмного забезпечення NOVASYS АСКУЄ» надається в якості спеціалізованого програмного забезпечення.

3.3. Вилучення, перетворення і завантаження даних в сховищі

Інформація переміщується з джерел даних на основі певних правил в централізоване сховище. У системах транзакційних джерел даних дані, необхідні для зберігання, Не зберігаються в остаточному вигляді. Ці дані можуть бути отримані з вихідних баз даних шляхом спеціальних перетворень, обчислень і агрегування.

Крім того, незважаючи на різну функціональну орієнтацію, вихідні транзакційні системи часто "перекриваються" в даних, тобто їх локальні бази даних містять один і той же тип інформації за змістом. Це в першу чергу відноситься до нормативної та довідкової інформації, яка в тій чи іншій формі використовується в будь-якій операційній системі. Ця інформація включає в себе

дані про лічильниках, трансформаторах, структурі енергетичного підприємства і т. д. однак важливо відзначити, що одне і те ж значення даних зазвичай має різні формати, типи представлення, ідентифікації, одиниці виміру і так далі в різних системах. Наприклад, інший формат визначається тим, як організована система збору даних, а одиниці виміру електроенергії залежать від прийнятих стандартів для енергетичного підприємства: Вт, кВт, МВт.

Вся ця інформація повинна бути узгоджена перед завантаженням в сховище, щоб забезпечити цілісність і узгодженість аналітичних даних. Зіставлення даних також потрібно при завантаженні даних з одного джерела. Справа в тому, що в сховищі зберігаються Історичні дані, тобто дані за досить тривалий період часу.

В операційній системі дані зберігаються в повному вигляді протягом обмеженого періоду часу, після чого відправляються в архів. У разі змін в структурі або самих даних архіви не піддаються будь-якої додаткової обробки, а зберігаються в їх первісному вигляді. Тому, якщо вам необхідно мати дані за досить тривалий період часу, вам необхідно узгодити архівовану інформацію з поточною.

Таким чином, завантаження даних з джерел в сховище здійснюється спеціальними процедурами, які дозволяють :

- як витягти дані з різних баз даних цих текстових файлів;
- виконувати різні типи зіставлення і очищення даних;
- як конвертувати дані при переміщенні їх з джерел в сховище;
- завантажувати узгоджені і" очищені " дані в структури зберігання.

Інструментами для створення процедур є окремі модулі програмного пакету, які забезпечують автоматичну генерацію процедур завантаження на основі декларативної інформації про джерела, правила затвердження і перетворення. Процедури створюються на мові запитів SQL для різних систем управління базами даних MSSQL, PostgreSQL, Oracle та ін., а також на мовах програмування (в залежності від реалізації програмного пакету). Декларативна

інформація вводиться адміністратором передачі даних і зберігається у вигляді метаданих в системному сховищі.

Витяг, перетворення і завантаження даних можуть виконуватися або безпосередньо шляхом ручного виклику відповідних процедур, або автоматично на основі сценаріїв і розкладів, складених на етапі розробки системи.

Метадані. З технічної точки зору метадані являють собою набір специфікацій і даних, які в цілому дають відповіді на питання про те, яка міра охоплення предметної області в інформаційній системі (ІС), які дані в ній представлені, яка архітектура системи і т. д.

Зокрема, метадані містять семантичну інтерпретацію або інтерпретацію змісту елементів даних, що циркулюють в ІС, а також опис обчислювального середовища, предметних областей, інформаційної безпеки і т. д.

Основними компонентами метаданих сховища є описи таблиць, їх атрибутів, ключів і так далі. Крім того, метадані включають описи перетворень:

- ідентифікація полів джерела даних;
- в відповідність між атрибутами сутностей джерела даних і атрибутами об'єктів СД;
- перетворення атрибутів;
- фізичні характеристики перетворень;
- перетворення таблиць кодування та довідкових таблиць;
- зміни імен (збіг імен джерел і об'єктів СД);
- зміна ключових атрибутів;
- значення полів за замовчуванням;
- логіка (алгоритми) генерації СД-даних з декількох джерел (пріоритет джерела);
- алгоритми перетворення даних і т. д.

Метадані також включають алгоритми агрегування і підсумовування даних, критерії вибірки з джерел, правила перетворення цих джерел перед їх завантаженням в сховище, описи взаємозв'язків між об'єктами зберігання, їх потужність і т. д

У програмному пакеті метадані реалізовані у вигляді правил і взаємозв'язків між джерелами даних з відповідним набором визначень і характеристик і поміщені в окреме сховище. Далі розробляється інтерфейс для їх редагування адміністратором. Таким чином, після прийняття відповідних правил вони включаються в механізми обробки і відправки даних, які є частиною планувальника завдань і працюють автоматично.

Перетворення і очищення даних. У програмному пакеті АСКУЄ перетворення даних включає в себе:

- приведення даних до одиниць вимірювання, прийнятих для СД (кВт);
- агрегування даних до 30 хвилинних значень;
- розрахунок тарифних зон;
- очищення даних від неправдивої інформації (для значень визначені маркери якості);
- перерахунки даних: втрати електроенергії, різні агрегації, розрахунок балансу, розрахунки в МВт з використанням спеціального методу;
- інформація про замовлення за часом і т. д.

3.4. Сховище даних

СД призначений безпосередньо для зберігання значущої, перевіреної, послідовної, несуперечливої і хронологічно повної інформації, яку можна вважати достовірною з досить високим ступенем впевненості.

Наше власне сховище даних не орієнтоване на вирішення будь-якої конкретної функціональної аналітичної задачі.

Мета сховища-забезпечити цілісність і підтримувати часову шкалу всіх видів корпоративних даних, і з цієї точки зору воно не залежить від додатків.

Сховище даних для АСКУЄ організовано у вигляді реляційної бази даних, і багатовимірне представлення буде надано вже на етапі аналізу, що дозволяє забезпечити стабільну швидкість виконання запитів незалежно від розміру сховища, а також забезпечити високий рівень захисту даних і хороші можливості

розмежування прав доступу. Це також дозволяє зробити аналітичні процедури та функції більш структурованими та гнучкими порівняно з використанням багатовимірної бази даних.

Для забезпечення високої надійності даних на сервері СД встановлена операційна система UNIX, а спеціалізоване програмне забезпечення комплексу АСКУЄ створено на кроссплатформенних мовах (наприклад, C++, з використанням QT). В якості сервера управління базами даних обраний PostgreSQL. Програмне забезпечення виконується як планувальник завдань, який відповідає за запит і отримання даних з джерел, а також за виконання реплікації даних у вітринах магазинів.

3.5. Вітрини даних

По суті, вітрина-це відносно невелике, але функціонально орієнтоване сховище, в якому інформація зберігається особливим чином, оптимізованим з точки зору вирішення конкретних аналітичних завдань певного підрозділу або групи аналітиків.

Інформація надходить у вітрини магазинів зі сховища, і вони називаються залежностями. Для заповнення вітрин даних в програмному пакеті реалізований механізм «реплікації» (копіювання) даних.

Опис механізму реплікації даних. Вітрини зберігання даних – це сервери енергетичних підрозділів, підпорядкованих конкретному енергетичному підприємству, сервер якого є сховищем даних.

У сховищі метаданих створюються певні правила з мережевими настройками кожного управління живленням, а також із зазначенням діапазону реплікованих даних.

Коли дані надходять в СД, автоматично створюється список змін, і для кожного з них визначається відповідність правилу реплікації. Механізм реплікації запускається планувальником завдань з певною частотою. як

Механізм реплікації реалізований на кроссплатформенном мовою програмування, а також з можливістю роботи з різними СУБД: PGSQL, MSSQL, Oracle. Це гарантує, що цей механізм не залежить від того, як база даних реалізована на сервері демонстрації даних.

3.6. Візуалізація даних

Візуалізація даних – це візуальне представлення великих обсягів числової та іншої інформації, що можливо завдяки використанню комп'ютерної графіки.

Використовуючи візуальні елементи на панелі керування, користувач повинен мати можливість створити запит для побудови таблиці або графічного відображення даних. Крім того, для підвищення зручності використання результатів запиту повинна бути передбачена можливість експорту даних у файли відповідно до певних форматів. Наприклад, файли MS Excel або звіти за шаблоном 30XXX.

Система також повинна забезпечувати автоматичну генерацію звітів на основі попередньо згенерованих шаблонів і ручне створення форм звітів на основі даних СД.

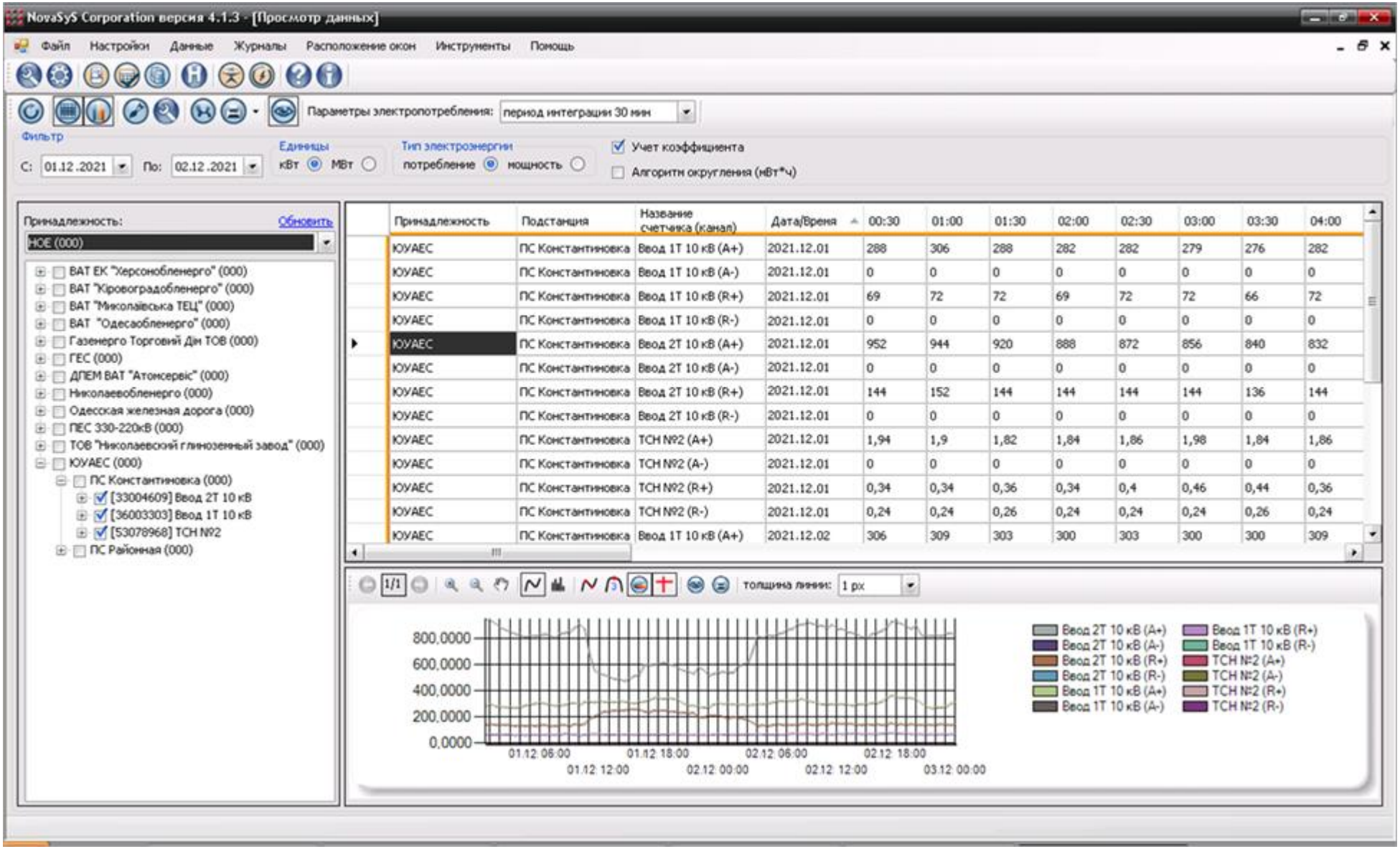
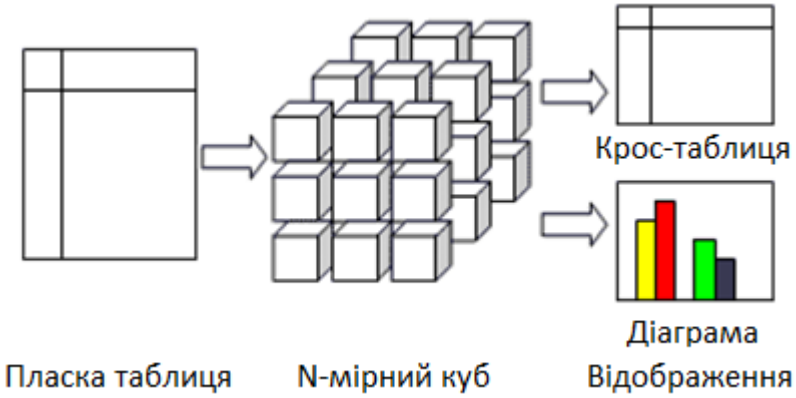


Рис. 3.2 Графічне і табличне відображення даних за запрошеними користувачем критеріями

3.7. OLAP аналіз

Схема роботи

Загальну схему роботи настільної OLAP-системи можна представити наступним чином:



3.3 Схема роботи OLAP системи

Алгоритм роботи :

1. Отримання даних у вигляді плоскої таблиці або результату виконання SQL-запиту.
2. Кешування даних і перетворення їх в багатовимірний куб.
3. Відобразіть побудований куб за допомогою перехресної таблиці або діаграми і т.д. у загальному випадку до одного куба може бути підключено довільну кількість відображень.

Дисплеї, що використовуються в системах OLAP, найчастіше бувають двох типів: перехресні таблиці і діаграми. Розглянемо перехресну таблицю, яка є основним і найбільш поширеним способом відображення куба.

Крос-таблиця

На рисунку нижче рядки і стовпці, що містять агреговані результати, показані жовтим кольором, осередки, що містять факти, відзначені світло-сірим кольором, а осередки, що містять дані вимірювань, відзначені темно-сірим кольором.

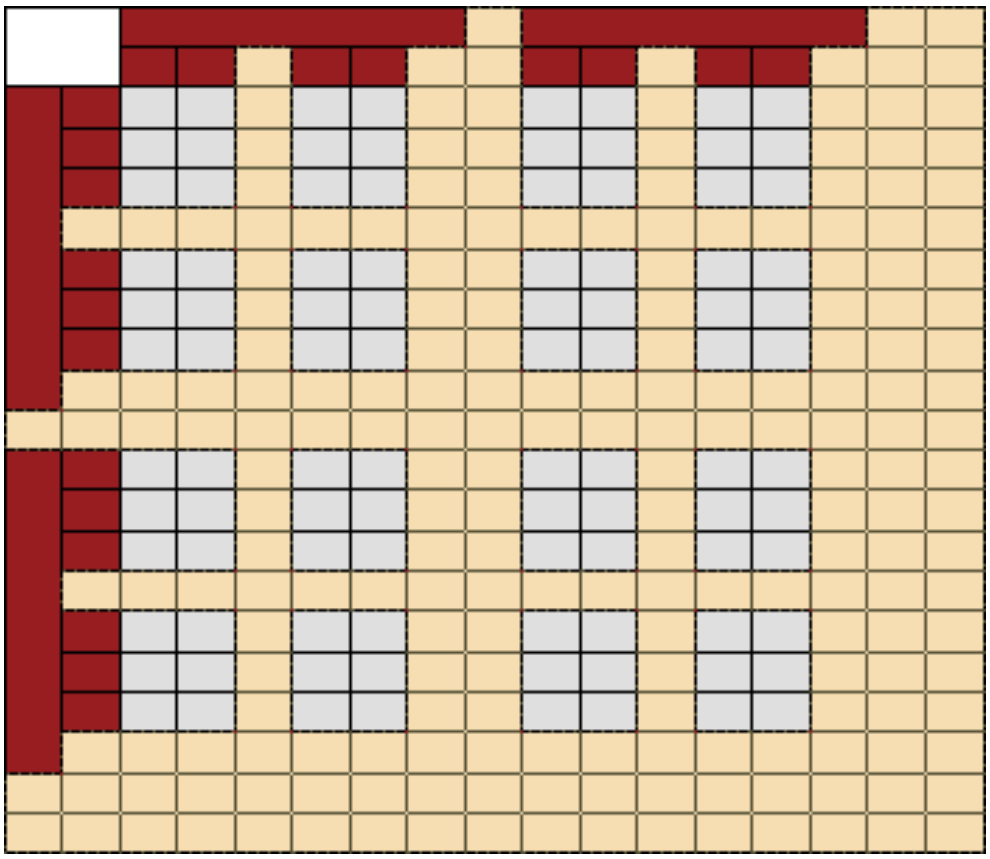


Рис. 3.4 Крос-таблиця

Таким чином, таблицю можна розділити на наступні елементи, з якими ми будемо працювати надалі:

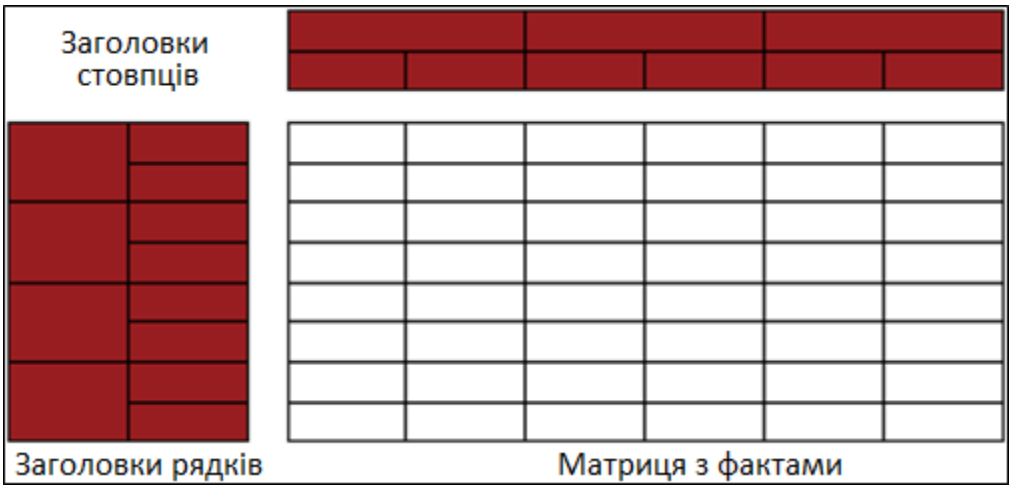


Рис. 3.5 Виділення ключових елементів в крос-таблиці

При заповненні матриці фактами виконуються наступні дії:

- На основі даних вимірювань визначте координати прикріпленого елемента в матриці.
- Визначте координати стовпців і рядків підсумків, на які впливає доданий елемент.
- Додайте елемент в матрицю і відповідні стовпці і рядки підсумків.

У той же час слід зазначити, що отримана матриця буде дуже розрідженою, саме тому її організація у вигляді двовимірного масиву (варіант, що лежить на поверхні) не тільки нераціональна, але, швидше за все, неможлива через велику розмірності цієї матриці, для зберігання якої недостатньо оперативної пам'яті. Наприклад, якщо наш куб містить інформацію про електроенергетичну компанію за один рік, і якщо він має тільки 3 Вимірювання-місце розташування (500), Код якості (10) і дату (365*48), то ми отримуємо матрицю фактів наступних вимірювань:

Кількість елементів = 500 x 10 x 365 x 48 = 87600000

І це при тому, що в матриці може бути всього кілька тисяч заповнених елементів. Більш того, чим більше число вимірювань, тим більше розрідженої буде матриця.

Тому для роботи з цією матрицею необхідно застосовувати спеціальні механізми для роботи з розрідженими матрицями.

Розріджена матриця – матриця з такою великою кількістю нульових елементів, що застосування спеціальних методів обробки виправдано. Зберігання таких матриць в одній з компактних форм дозволяє вирішувати завдання значно більших розмірів у порівнянні з методами загального призначення. Метод використання розрідженості очевидний для ітераційних методів обчислювальної лінійної алгебри, основною операцією якої є множення матриці на вектор. Був розроблений ряд алгоритмів для вирішення лінійних систем з розрідженими матрицями більш загального вигляду. Вони засновані на добре відомих прямих методах-методі Гауса, ортогональних методах - і

спрямовані на те, щоб максимально зменшити заповнення матриці (тобто поява нових ненульових елементів, які також вимагають зберігання під час вирішення завдання).

Давайте тепер розглянемо, як ми можемо визначити координати факту, знаючи відповідні йому вимірювання. Для цього давайте докладніше розглянемо структуру заголовка:

Изм. 1			Изм. 2			Итог
Изм. 1	Изм. 2	Итог	Изм. 1	Изм. 2	Итог	

У той же час ви можете легко знайти спосіб визначити номери відповідної комірки і підсумкові значення, в які вона потрапляє. Тут ви можете запропонувати кілька підходів. Один з них-використовувати дерево для пошуку відповідних осередків. Це дерево можна побудувати, пройшовши через зразок. Крім того, ви можете легко визначити аналітичну рекурентну формулу для обчислення необхідної координати.

Підготовка даних

Дані, що зберігаються в таблиці, повинні бути перетворені для її використання. Таким чином, для підвищення продуктивності при побудові гіперкуба рекомендується знаходити унікальні елементи, що зберігаються в стовпцях, які є розмірами куба. Ви також можете попередньо агрегувати факти для записів, що мають однакові значення вимірювань. Як згадувалося вище, унікальні значення, Доступні в полях вимірювань, важливі для нас. Тоді ми можемо запропонувати наступну структуру для їх зберігання:

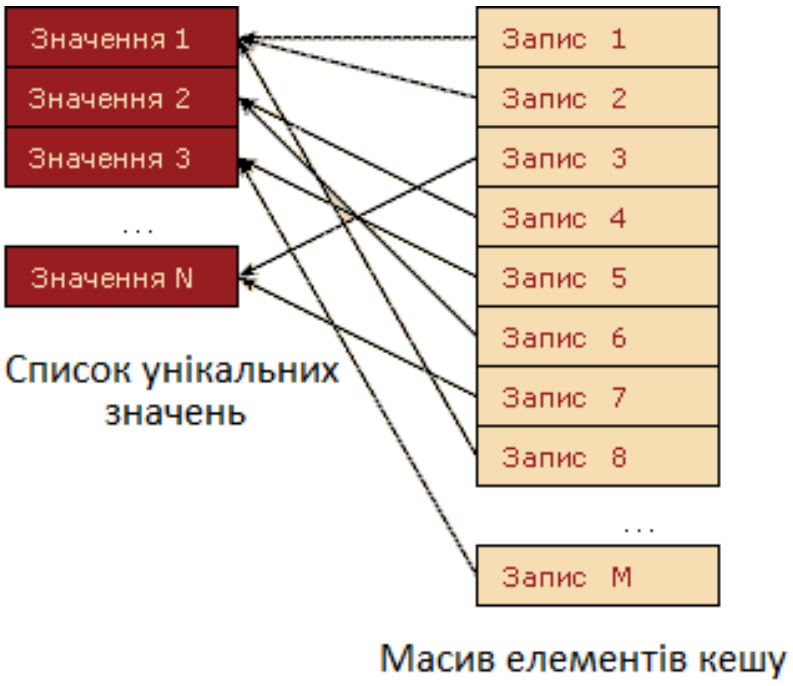


Рис. 3.6 Структура елементів

При використанні цієї структури ми значно знижуємо потребу в пам'яті. Це цілком актуально, так як для збільшення швидкості роботи бажано зберігати дані в оперативній пам'яті. Крім того, ви можете зберегти тільки масив елементів і завантажити їх значення на диск, так як вони знадобляться нам тільки при відображенні перехресної таблиці.

Завантаження даних в гіперкуб

Першим кроком системи є завантаження даних і перетворення їх у внутрішній формат.

Для системи OLAP стовпці таблиці можуть бути або фактами, або вимірами. Однак логіка роботи з цими стовпцями буде іншою. У Гіперкубі розміри фактично є осями, а значення вимірювань є координатами на цих осях. У цьому випадку Куб буде заповнений дуже нерівномірно - будуть комбінації координат, які не будуть відповідати ніяким записам, і будуть комбінації, відповідні декільком записам у вихідній таблиці, і перша ситуація більш поширена, тобто Куб буде виглядати як Всесвіт - порожній простір, в деяких місцях якого є скупчення точок (фактів). Таким чином, якщо ми попередньо зберемо дані при початковому завантаженні даних, тобто. об'єднуйте записи, які

мають однакові значення вимірювань, при обчисленні попередніх агрегованих значень фактів, тоді в майбутньому нам доведеться працювати з меншою кількістю записів, що збільшить швидкість роботи і знизить вимоги до обсягу оперативної пам'яті.

Для побудови розділів гіперкуба нам потрібні наступні функції: визначення координат (фактично значень вимірювань) для записів таблиці, а також визначення записів, які мають конкретні координати (значення вимірювань).

Найпростіший спосіб зберігання гіперкуба-використовувати базу даних власного внутрішнього формату. Схематично перетворення можна представити так, як показано на рис. 3.7.

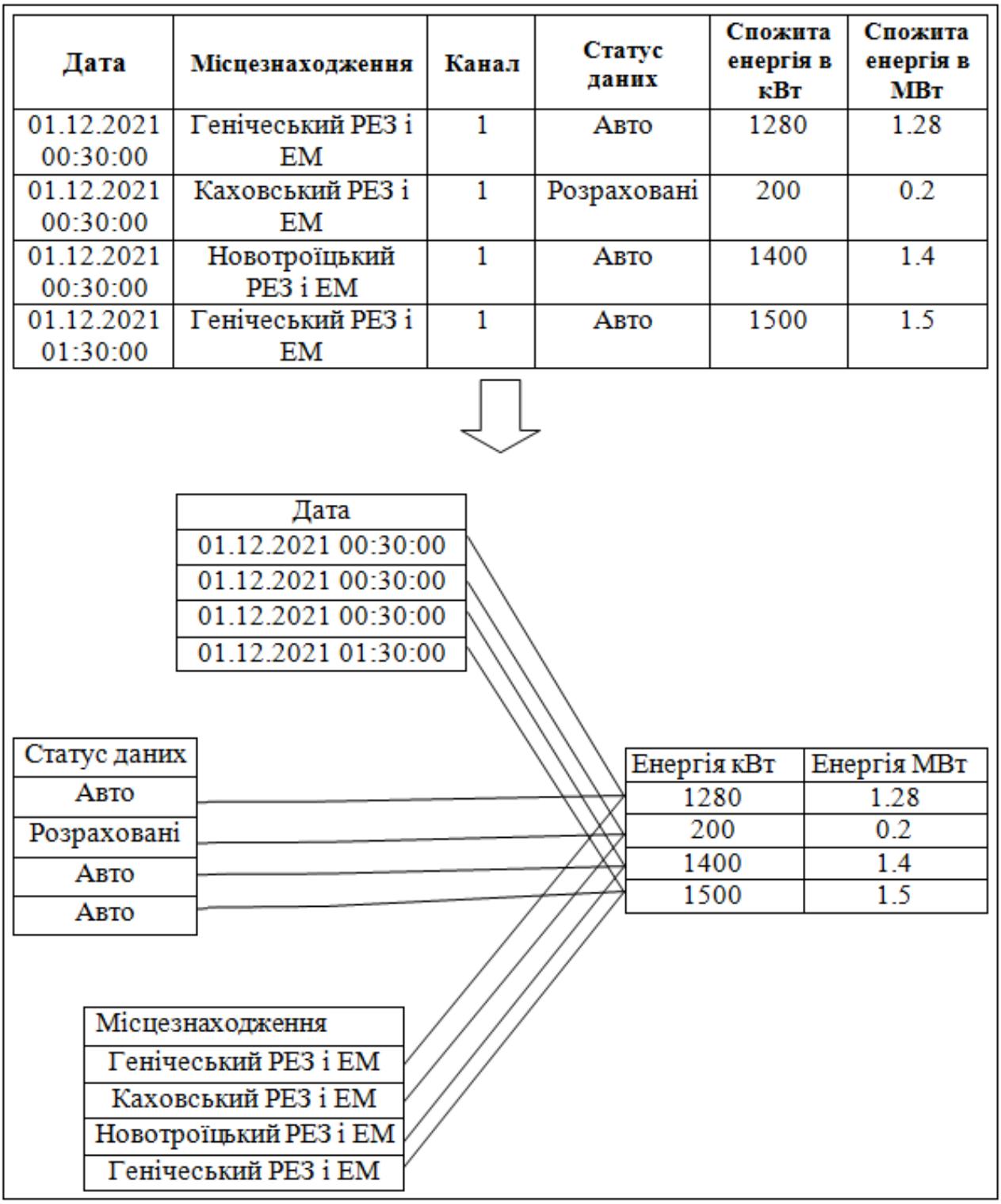


Рис. 3.7 Схема перетворень таблиць

Іншими словами, замість однієї таблиці ми отримали нормалізовану базу даних. Насправді нормалізація знижує швидкість роботи системи, - можуть сказати фахівці з баз даних, і в цьому вони, безумовно, будуть праві, коли нам знадобиться отримати значення для елементів словника (в нашому випадку,

значення вимірювань). Але вся справа в тому, що нам взагалі не потрібні ці значення на етапі побудови зрізу. Як уже згадувалося вище, нас цікавлять тільки координати в нашому Гіперкубі, тому ми визначимо координати для значень вимірювань. Найпростіший спосіб-перенумерувати значення елементів. Щоб забезпечити однозначну нумерацію в межах одного виміру, ми заздалегідь відсортуємо списки значень вимірювань (словники, виражені в термінах БД) в алфавітному порядку. Крім того, перенумерувавши факти, а факти попередньо агреговані, ми отримаємо схему, показану на рис. 3.8.

№	Дата
1	01.12.2021 00:30:00
2	01.12.2021 01:30:00

№	Енергія кВт	Енергія МВт
1	1280	1.28
2	200	0.2
3	1400	1.4
4	1500	1.5

№	Статус даних
1	Авто
2	Розраховані

№	Місцезнаходження
1	Геніческий РЕЗ і ЕМ
2	Каховський РЕЗ і ЕМ
3	Новотроїцький РЕЗ і ЕМ

Рис. 3.8 Схема проіндексованих таблиць вимірів і фактів

Тепер все, що залишається, – це зв'язати елементи різних таблиць разом. У теорії реляційних баз даних це робиться за допомогою спеціальних проміжних таблиць. Досить для кожного запису в таблицях вимірювань помістити в рядок список, елементами якого будуть номери фактів, які були сформовані з використанням цих вимірювань (тобто визначити всі факти, які мають однакове значення координати, описуваної цим виміром). Для фактів, відповідно до

кожного запису, ми будемо зіставляти значення координат, за якими вона розташована в Гіперкубі. У майбутньому скрізь координати запису в Гіперкубі будуть розумітися як номери відповідних записів в таблицях значень вимірювань. Потім для нашого гіпотетичного прикладу ми отримуємо набір, що визначає внутрішнє представлення гіперкуба, показаного на рис. 3.9.

№	Дата	№ фактів
1	01.12.2021 00:30:00	1, 2, 3
2	01.12.2021 01:30:00	4

№	Статус даних	№ фактів
1	Авто	1, 3, 4
2	Розраховані	2

№	Місцезнаходження	№ фактів
1	Генічеський РЕЗ і ЕМ	1, 4
2	Каховський РЕЗ і ЕМ	2
3	Новотроїцький РЕЗ і ЕМ	3

№	Дата	Місцезнаходження	Канал	Статус даних	Спожита енергія в кВт	Спожита енергія в МВт
1	1	1	1	1	1280	1.28
2	1	2	1	2	200	0.2
3	1	3	1	1	1400	1.4
4	2	1	1	1	1500	1.5

Рис. 3.9 Зіставлення індексів фактів і вимірів

Це внутрішнє уявлення гіперкуба. Оскільки ми не робимо це для реляційної бази даних, ми просто використовуємо поля змінної довжини в якості полів зв'язку для значень вимірювань (ми не змогли б зробити це в РБД, так як кількість стовпців таблиці там визначено заздалегідь).

Реалізація гіперкуба

Ви можете спробувати використовувати набір тимчасових таблиць для реалізації гіперкуба, але цей метод забезпечить занадто низьку продуктивність, тому ми будемо використовувати наші власні структури зберігання даних.

Для реалізації гіперкуба нам необхідно використовувати структури даних, які забезпечать максимальну продуктивність і мінімальне споживання оперативної пам'яті. Очевидно, що наші основні структури будуть призначені для зберігання словників і таблиць фактів. Давайте розглянемо завдання, які словник повинен виконувати з максимальною швидкістю:

- перевірка наявності елемента в словнику;
- додавання елемента до словника;
- пошук номерів записів, що мають певне значення координат;
- пошук координат за значенням вимірювання;
- пошук значення вимірювання по його координаті.

Для задоволення цих вимог можуть використовуватися різні типи і структури даних. Наприклад, ви можете використовувати масиви структур. У реальному випадку ці масиви вимагають додаткових механізмів індексування, які збільшать швидкість завантаження даних і отримання інформації.

Щоб оптимізувати роботу гіперкуба, необхідно визначити, які завдання необхідно вирішити в першочерговому порядку, і за якими критеріями нам необхідно підвищити якість роботи. Головне для нас-збільшити швидкість роботи програми, і бажано, щоб був потрібний невеликий обсяг оперативної пам'яті. Продуктивність може бути підвищена за рахунок впровадження додаткових механізмів доступу до даних, таких як індексування. На жаль, це збільшує навантаження на оперативну пам'ять. Тому ми визначимо, які операції нам необхідно виконувати з максимальною швидкістю. Для цього давайте розглянемо окремі компоненти, що реалізують гіперкуб. Ці компоненти мають два основних типи-Вимірювання і таблиця фактів. Для вимірювання типовим завданням буде:

- додавання нового значення;
- визначення координати на основі значення вимірювання;
- визначення значення по координаті.

При додаванні нового значення в елемент нам потрібно перевірити, чи є у нас вже таке значення, і якщо є, то не додавайте нове, а використовуйте існуючу координату, в іншому випадку нам потрібно додати новий елемент і визначити його координату. Для цього вам потрібен спосіб швидкого пошуку наявності потрібного елемента (крім того, ця проблема виникає при визначенні координати на основі значення елемента). Для цього оптимально використовувати хешування. У цьому випадку оптимальною структурою є використання хеш-дерев, в яких ми будемо зберігати посилання на елементи. В цьому випадку елементами будуть рядки словника вимірювань.

І ми будемо зберігати посилання на унікальні елементи в хеш-дереві. Крім того, нам необхідно вирішити задачу зворотного перетворення - визначити значення вимірювання за координатою. Для максимальної продуктивності вам необхідно використовувати пряму адресацію. Тому Ви можете використовувати інший масив, індекс в якому є координатою вимірювання, а значення-посиланням на відповідний запис у словнику. Однак ви можете зробити щось простіше (і заощадити пам'ять), якщо розташуєте масив елементів відповідним чином, щоб індекс елемента був його координатою.

Організація масиву, що реалізує список фактів, не представляє особливої проблеми через його просту структуру. Єдине, що слід зазначити, це те, що бажано розрахувати всі методи агрегування, які можуть знадобитися і які можуть бути розраховані поступово (наприклад, сума).

Побудова зрізів куба.

Розглянутий раніше метод зберігання даних у вигляді гіперкуба дозволяє формувати набір точок в багатовимірному просторі на основі інформації, що

зберігається в сховищі даних. Для того щоб людина могла працювати з цими даними, вони повинні бути представлені у формі, зручній для обробки. При цьому в якості основних видів подання даних використовуються зведена таблиця і графіки. Більш того, обидва ці методи насправді є проекціями гіперкуба. Щоб забезпечити максимальну ефективність при побудові уявлень, ми почнемо з того, що являють собою ці проекції.

Заголовки зведеної таблиці мають чітку ієрархічну структуру, тому природно припустити, що для їх зберігання буде використовуватися дерево. В цьому випадку схематичну структуру вузла дерева можна представити наступним чином:

Батьківський вузол	**Значення виміру**	**N (Кількість дочірніх вузлів)**	**Стовпець (рядок) зі значенням**
Дочірній вузол 1	Дочірній вузол 2	...	Дочірній вузол N

У той же час логічно Зберегти посилання на відповідний елемент таблиці вимірювань багатовимірного куба в якості значення вимірювання. Це зменшить споживання пам'яті для зберігання зрізу і прискорить вашу роботу. Посилання також використовуються як батьківські та дочірні вузли.

Кожен рівень дерева буде відповідати одному виміру в тому порядку, в якому вони використовуються для побудови зрізу. В цьому випадку самий верхній вузол дерева буде відповідати повному результату в зведеній таблиці.

Щоб додати елемент в дерево, необхідно мати інформацію про його місцезнаходження в Гіперкубі. В якості такої інформації вам необхідно використовувати її координату, яка зберігається в словнику значень вимірювань.

Розглянемо схему додавання елемента в дерево заголовків зведеної таблиці. У цьому випадку ми використовуємо значення координат вимірювання в якості вихідної інформації. Порядок, в якому перераховані ці вимірювання, визначається необхідним методом агрегування і збігається з рівнями ієрархії

дерева заголовків. В результаті вам необхідно отримати список стовпців або рядків зведеної таблиці, в яку ви хочете додати елемент (рис. 3.10).

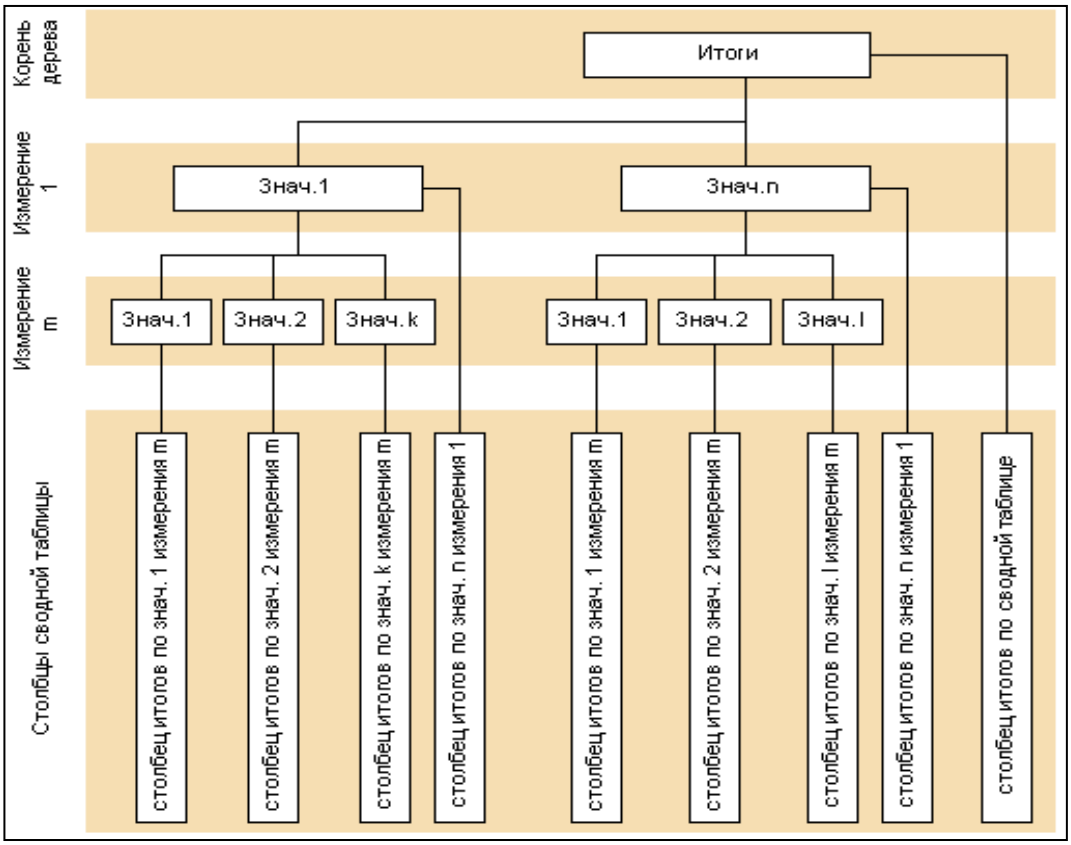


Рис. 3.10 Схема додавання елементу в дерево

Використовуємо координати вимірювання як вихідні дані для визначення цієї структури. Крім того, для визначеності ми будемо вважати, що визначаємо цікавить стовпець в матриці (ми розглянемо, як визначити рядок пізніше, так як там зручніше використовувати інші структури даних, і причину цього вибору див.нижче). В якості координат візьмемо цілі числа-числа значень вимірювань.

Отже, після вибірки з масивів вимірювань ми отримуємо масив посилань на стовпці розрідженої матриці. Тепер вам потрібно виконати всі необхідні дії з лініями. Для цього знайдіть потрібний елемент всередині кожного стовпця і додайте туди відповідне значення.

Тепер давайте розглянемо форму, в якій значення повинні бути представлені всередині стовпців, тобто як визначити необхідну рядок. Для цього

ви можете використовувати кілька підходів. Найпростіше було б представити кожен стовпець у вигляді вектора, але, оскільки він буде дуже розрідженим, пам'ять буде витрачатися вкрай неефективно. Щоб уникнути цього, ми будемо застосовувати структури даних, які забезпечать більшу ефективність при поданні розріджених одновимірних масивів (векторів). Найпростіший з них-звичайний список, одно-або двусвязний, але він неекономічний з точки зору доступу до елементів. Тому ми будемо використовувати дерево, яке забезпечить більш швидкий доступ до елементів. Наприклад, ви можете використовувати те ж дерево, що і для стовпців, але тоді вам доведеться створювати своє власне дерево для кожного стовпця, що призведе до значних витрат пам'яті і часу обробки. Але краще створити єдине дерево для зберігання всіх комбінацій вимірювань, використовуваних в рядках, яке буде ідентично описаному вище, але його елементи будуть не покажчиками на рядки (яких як таких не існує), а їх індексами, а самі значення індексів нас не цікавлять і використовуються тільки в якості унікальних ключів. Потім ми будемо використовувати ці ключі для пошуку потрібного елемента всередині стовпця. Самі стовпці найпростіше представити у вигляді звичайного двійкового дерева. Графічно отриману структуру можна представити наступним чином:

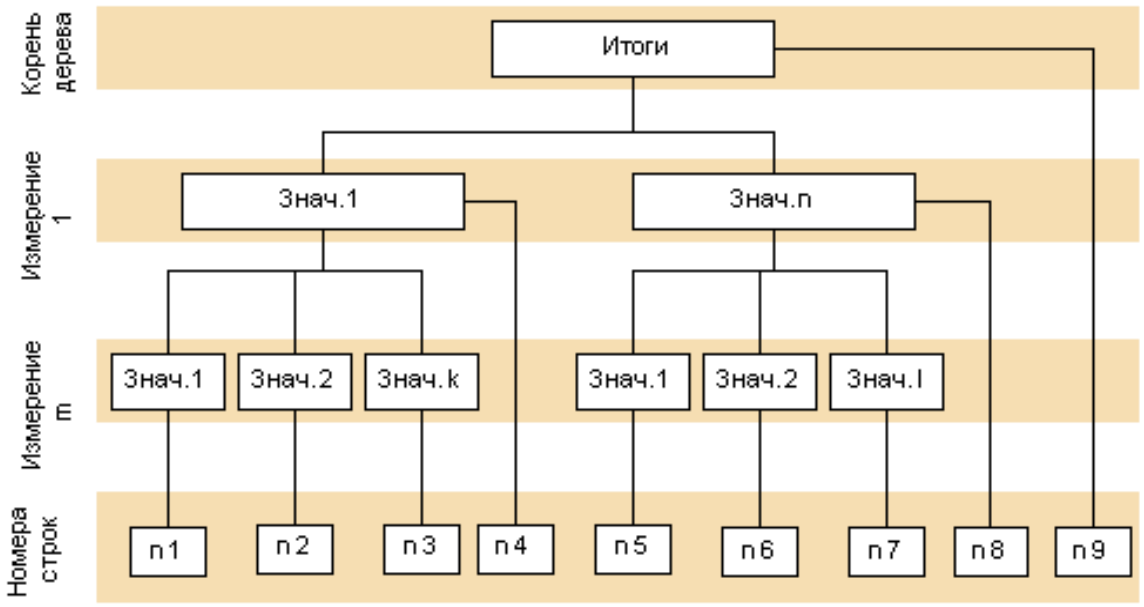


Рис. 3.11 Двійкове дерево

В узагальненому вигляді послідовність дій по додаванню елемента в матрицю можна описати наступним чином:

1. Визначте номери рядків, до яких додаються елементи;
2. Визначте набір стовпців, до яких додаються елементи;
3. Для всіх стовпців знайдіть елементи з потрібними номерами рядків і додайте до них поточний елемент (додавання включає підключення необхідної кількості значень фактів і обчислення агрегованих значень, які можуть бути визначені поступово).

Після виконання цього алгоритму ми отримуємо матрицю, яка являє собою зведену таблицю, яку нам потрібно було побудувати.

3.8. Реалізація OLAP механізму

Для зручності відображення даних, отриманих з гіперкуба, і спрощення їх використання механізм OLAP реалізований в додатковому модулі звітності, який реалізований в якості доповнення в програмному продукті MSExcel.

Критерії вибору даних визначаються за допомогою Користувацької функції, яка створюється за допомогою графічного інтерфейсу (рис. 3.12).

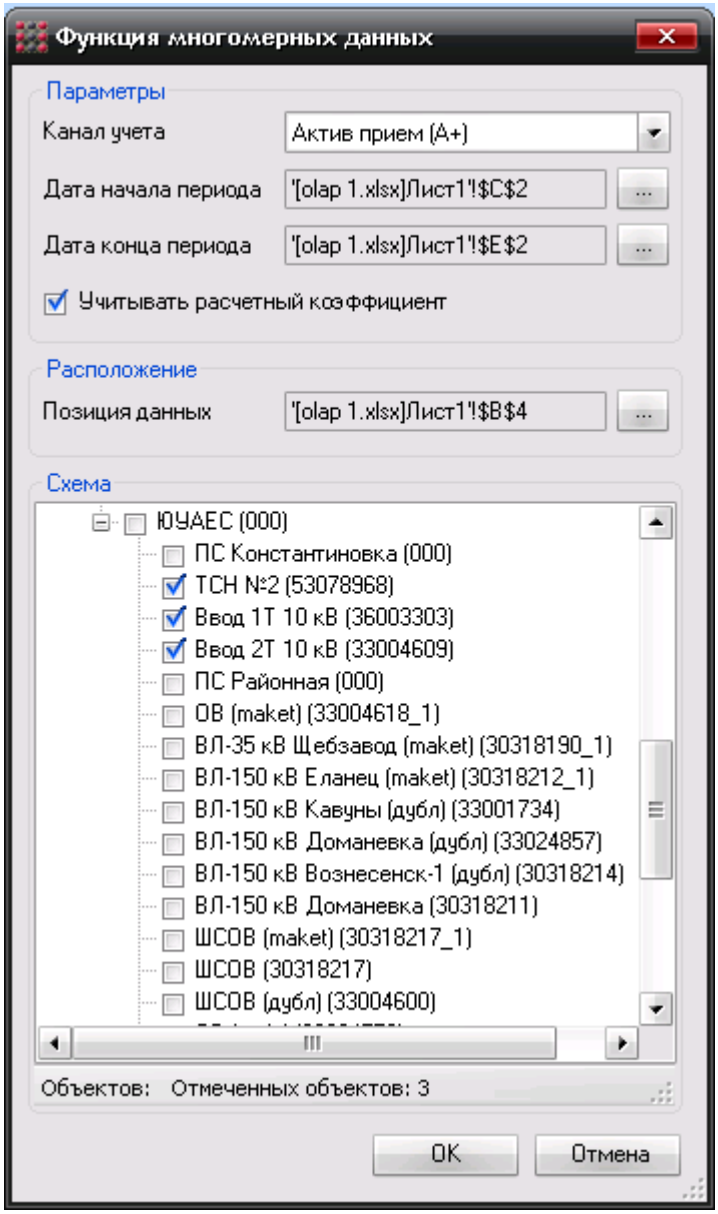


Рис. 3.12 Інтерфейс створення призначеної для користувача функції для отримання даних куба

Це задає часовий інтервал і об'єкти, для яких вибираються дані з СД, а також комірку, з якої почнеться побудова багатовимірної таблиці. Коли ви закриваєте вікно, створюється функція перегляду:

= SingleChannelFunction("630, 640,641";Лист1!$C $2;Лист1!$E $2;1;1;Лист1!$C $4)

- Дані отримуються натисканням кнопки "Відновити дані"на додатковій панелі інструментів Novasys. Оскільки період часу також встановлюється

спеціальними функціями, при оновленні їх значення визначаються за допомогою графічного інтерфейсу рис. 3.13.

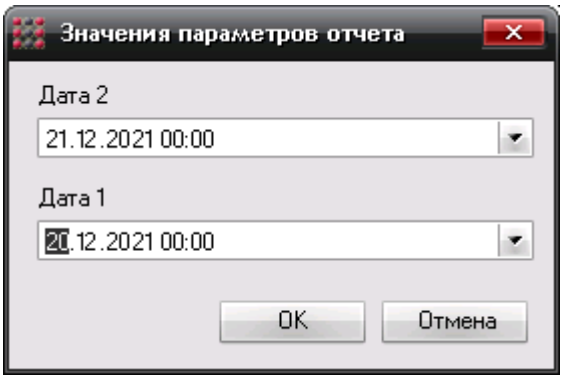


Рис. 3.13 Визначення початкової і кінцевої дати періоду даних

• Спосіб розміщення вимірювань в багатовимірній таблиці також задається у відповідному вікні, як відразу після отримання даних, так і в будь-який час при натисканні на кнопку «Шаблон OLAP» панелі інструментів «Новасис» (рис. 3.14.). Ви також можете вказати рівні агрегування даних вимірювань там.

• Доступні вимірювання перераховані в лівому списку на формі. Програма реалізована таким чином, що виконання додаткових вимірювань відбувається відносно швидко і не вимагає великих зусиль програміста. Передбачається, що робоча система включатиме близько 15 доступних вимірювань (як згадувалося вище, вимірюванням є вісь гіперкуба).

• Порядок, в якому вимірювання вводиться в список «рядки» / «стовпці», визначає позицію вимірювання. Наприклад: перший рядок-це крайнє ліве ім'я рядка, другий-справа, перший стовпець-це ім'я верхнього стовпця, а другий-нижній, і так далі

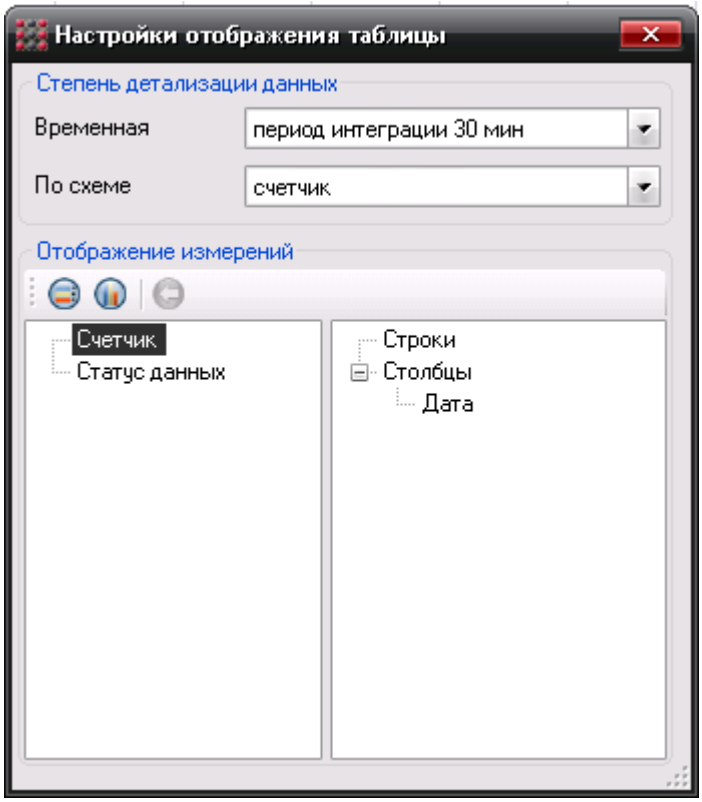


Рис. 3.14 Налаштування відображення даних OLAP куба

Протягом 2-3 секунд результат відображається в комірках аркуша, починаючи з комірки заголовка, зазначеної в налаштуваннях функції. Дані вимірювань виходять зліва направо і зверху вниз. Іншими словами, крайнє ліве вимірювання (в рядках) є першим критерієм для вибору значень з таблиці фактів, а найменше вимірювання (щодо стовпців) є останнім критерієм.

На рис. 3.15. показаний результат побудови куба даних у вигляді багатовимірної таблиці. Рядки-лічильники і коди якості даних, стовпці-значення даних.

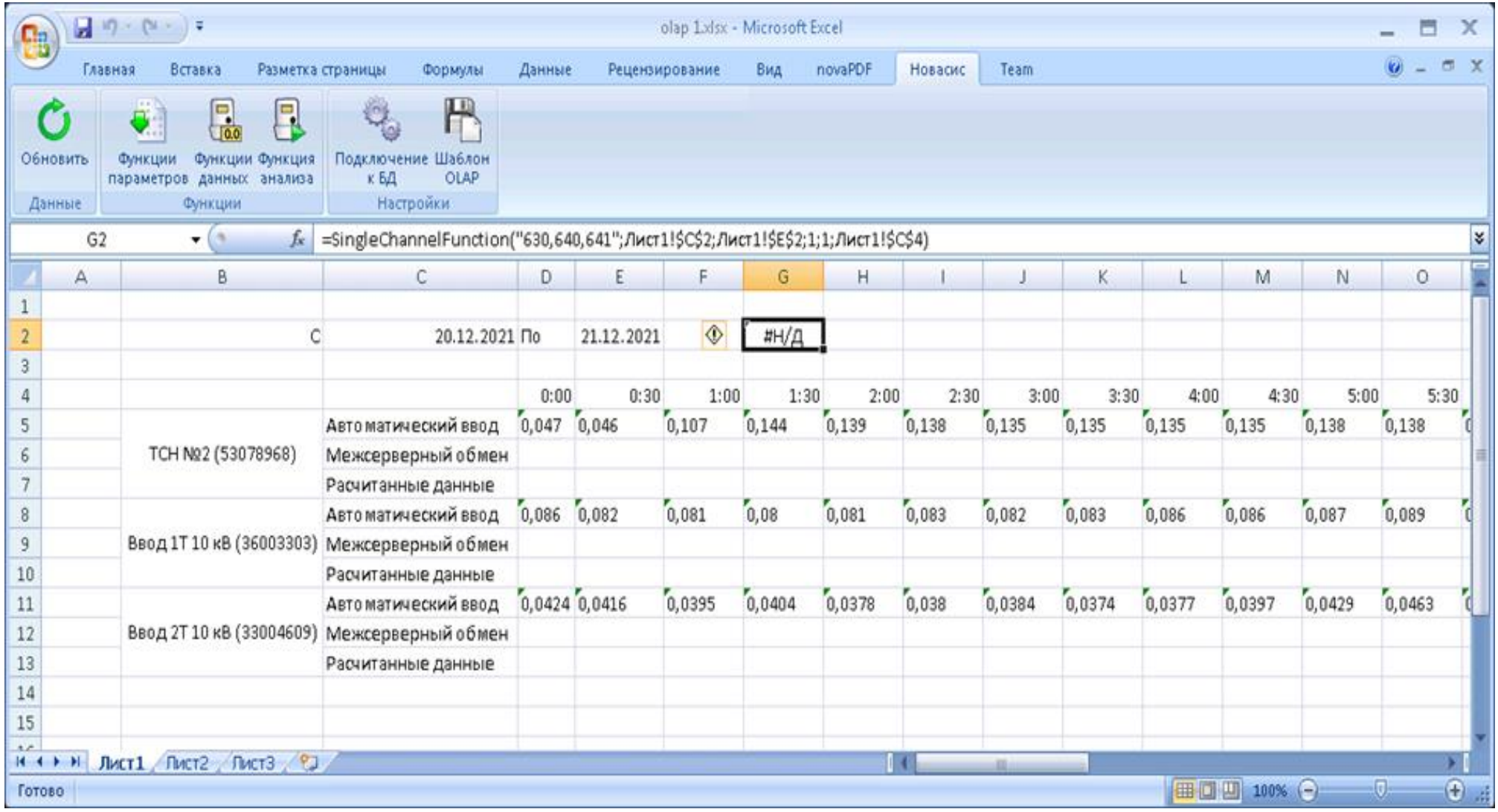


Рис. 3.15 Відображення багатовимірної таблиці

Висновки до третього розділу

У цьому розділі було розглянуто проект створення та впровадження інформаційно-аналітичної системи в роботу енергопідприємства. На першому рівні системи забезпечується автоматичне зчитування даних з приладів обліку і передача їх на сервер з операційною базою даних. Процедури очищення і перетворення дозволяють структурувати дані, забезпечити отримання якісної та актуальної інформації. Після виконання таких процедур дані в автоматичному режимі надходять в Сховище даних, де зберігаються в незмінному вигляді протягом заздалегідь затвердженого періоду зберігання.

Зі сховища, дані потрапляють у вітрини даних, які предметно орієнтовані і вузько спеціалізовані, що дозволяє діставати ефективний і швидкий доступ до даних. До вітрин даних користувачі отримують доступ через клієнтські програми, і мають можливість формувати звіти, аналізувати інформацію. Крім того, в проект закладається механізм аналітичної обробки даних, який знижує витрати часу на ручну аналіз даних і дозволяє формувати в автоматичному режимі складні багатовимірні звіти.

ВИСНОВКИ

Системи підтримки прийняття рішень досить складні і вимагають багато часу для впровадження. Вони вимагають ретельного визначення мети їх створення, аналізу роботи і структури підприємства, а також досить великих вкладень фінансових ресурсів в їх реалізацію. Однак, якщо компанії дійсно потрібна аналітична обробка даних і автоматизоване виявлення знань, впровадження такої системи окупиться максимум за рік.

Українські енергетичні підприємства зараз потребують глобальної автоматизації своєї діяльності: починаючи від вичитування даних з приладів обліку, закінчуючи аналітичною обробкою даних. Автоматизація дозволить вам забезпечити максимальну якість даних, усунути помилки в розрахунках і підвищити ефективність роботи людей. Процес такої автоматизації почався всього кілька років тому, і сьогодні навіть не вся інформація була переведена з паперових записів в електронну структуровану форму. Таким чином, помилки звітності між відповідними ліцензіатами та енергетичним ринком все ще величезні.

Аналітична обробка даних в енергетичних компаніях поки взагалі не використовується. Але за допомогою його впровадження можна буде прогнозувати споживання енергії, знижувати втрати в електричних ланцюгах і виявляти помилки в звітах, тим самим знижуючи витрати часу і фінансів.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Барсегян А. А., Купріянов М. С., Степаненко В. В., Холод І. І. Методи і моделі аналізу даних : OLAP і Data Mining. - СПб.: БХВ-Петербург, 2014.
2. Erik Thomsen. OLAP Solutions. Second Edition. Building multidimensional information systems. 2020
3. Бергер, А. б. Microsoft SQL Server 2005 Analysis Services. OLAP і багатовимірний аналіз даних - СПб.: БХВ-Петербург, 2017.
4. Бєлов В. С. Інформаційно-аналітичні системи. Основи проектування і застосування : навчальний посібник, керівництво, практикум / Московський державний університет економіки, статистики і інформатики. - М., 2015.
5. Сергій Кузнєцов. Основи сховищ даних і BI по Ральфу Кимбаллу. http://citcity.ru/ 16022/
6. Олексій Федоров, Наталія Елманова. Введення в OLAP -Диалог-МИФИ. 2020
7. Олексій Арустамов. Попередня обробка і очищення даних перед завантаженням в сховищі. http://www.basegroup.ru/library/dw olap/dataclearing/
8. Олексій Стариков. Ядро OLAP системи. http://www.basegroup.ru/library/dw olap/olap core part 1/
9. Oracle. Сховища даних і аналітичні системи. 2017
10. В.Е. ТумановВ. Е. Туманів. Проектування сховищ даних для додатків систем ділової обізнаності (Business Intelligence Systems). 2020 р. http://www.intuit.ru/department/database/bispowerd/
11. SAS Visual Text Analytics, URL:https://www.SAS.com/ru_ru/software/visual-text-analytics.html
12. Бідюк П.І., Романенко В.Д., Тимощук О.Л. Аналіз часових рядів (навчальний посібник) — Київ: Політехніка, 2010. — 317 с.
13. А.В. Антонов. Методы классификации и технология Галактика - Зум. Научно-техническая информация. Сер. 1. Вып. 6. 2004.

14. Документація SAS, URL: https://support.sas.com/en/documentation.html

15. Introduction to Information Retrieval, URL: https://nlp.stanford.edu/IR-book/

16. International Journal of Computer Applications (0975 – 8887) Volume 181 – No.1, липень 2018, Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents

17. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: исследование зависимостей. — М.: Финансы и статистика, 1985.

18. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.

19. Вапник В. Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.

20. Журавлев Ю. И., Рязанов В. В., Сенько О. В. «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006. ISBN 5-7036-0108-8.

21. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999. ISBN 5-86134-060-9.

22. Роман Шамин. Курс «Машинное обучение и искусственный интеллект в математике и приложениях». НОЦ Математического института им. В.А.Стеклова РАН Флах П. Машинное обучение. — М.: ДМК Пресс, 2015. — 400 с. — ISBN 978-5-97060-273-7.

23. A. L. Samuel Some Studies in Machine Learning Using the Game of Checkers. II-Recent Progress.

24. Mitchell T. Machine Learning. — McGraw-Hill Science/Engineering/Math, 1997. ISBN 0-07-042807-7.URL: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e382?gi=9553d6c5f3e8

25. Machine Learning Tom M. Mitchell, 432 pages

26. Nils J. Nilsson, Introduction to Machine Learning, 2018. URL: https://medium.com/datadriveninvestor/bias-and-variance-in-machine-learning-51fdd38d1f86

27. Pedro Domingos (September 2015), The Master Algorithm, Basic Books

28. Ray Solomonoff, An Inductive Inference Machine, IRE Convention Record, Section on Information Theory, Part 2, pp., 56–62, 1957.