Міністерство науки і освіти України
Житомирський державний технологічний університет

Кафедра програмного забезпечення систем

КУРСОВА РОБОТА

з дисципліни «Сучасні бази даних та аналіз даних»
на тему: Кластер ний аналіз покупців інтернет магазину

Студента 6 курсу ЗПІ-10м групи
спеціальності 7.05010301 «Програмне забезпечення систем»
._____ _____
(прізвище та ініціали)

Керівник: _____Сугоняк І. І._____

Національна шкала _____
Кількість балів: _____Оцінка: ECTS _____

Члени комісії: _____ _____
(підпис) (прізвище та ініціали)
_____ _____
(підпис) (прізвище та ініціали)
_____ _____
(підпис) (прізвище та ініціали)

м. Житомир – 2015 рік

ЗМІСТ

ВСТУП

Актуальність полягає в необхідністі оперативної аналітичної обробки інформації та ефективної організації великих обсягів даних для формування асортиментна товарів інтернет магазину. Проблеми узгодженості даних, оперативності виконання запитів та забезпечення доступу до інформації можуть бути вирішені з використанням технології сховищ даних.

Метою курсової роботи є дослідження особливостей проектування та реалізації сховищ даних інтернет магазину.

Завданням на курсову роботу є:

- аналіз теоретичних засад проектування та реалізації OLAP-систем;
- визначення інформаційних потреб з формування асортименту товарів;
- вибір фактів та вимірів для збереження;
- проектування сховища даних та перенесення даних;
- вибір математичних методів інтелектуального аналізу даних;
- реалізація звітності та інтерфейсних засобів інформаційної системи.

Предметом дослідження є можливості застосування концепції DATA MINING для забезпечення інформаційних потреб прийняття рішень з формування асортименту товарів.

Об‘єктом дослідження є методи та засоби проектування сховищ даних та застосування інструментарію DataMining і засобів багатовимірного аналізу для обробки даних.

1. ТЕОРЕТИЧНИЙ АНАЛІЗ МОДЕЛЕЙ ТА МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

1.1 Основні поняття Data Mining

Data Mining – це процес підтримки ухвалення рішень, заснований на пошуку в даних прихованих закономірностей (шаблонів інформації).

Технологію Data Mining достатньо точно визначає Григорій Піатецкий - Шапіро (Gregory Piatetsky-Shapiro) – один із засновників цього напряму: “Data Mining – це процес виявлення в сирих даних раніше невідомих, нетривіальних, практично корисних і доступних інтерпретації знань, необхідних для ухвалення рішень в різних сферах людської діяльності” .

Суть і мету технології Data Mining можна визначити так: це технологія, яка призначена для пошуку у великих об'ємах даних неочевидних, об'єктивних і корисних на практиці закономірностей.

Неочевидних – це значить, що знайдені закономірності не виявляються стандартними методами обробки інформації або експертним шляхом.

Об'єктивних – це значить, що знайдені закономірності повністю відповідатимуть дійсності, на відміну від експертної думки, яка завжди є суб'єктивною.

Практично корисних – це значить, що висновки мають конкретне значення, якому можна знайти практичне застосування.

Знання – сукупність відомостей, яка утворює цілісний опис, відповідний деякому рівню обізнаності про описуване питання, предмет, проблему і т.д.

Використовування знань (knowledge deployment) означає дійсне застосування знайдених знань для досягнення конкретних переваг (наприклад, в конкурентній боротьбі за ринок).

Приведемо ще декілька визначень поняття Data Mining.

Data Mining – це процес виділення з даних неявної і неструктурованої інформації і представлення її у вигляді, придатному для використовування.

Data Mining – це процес виділення, дослідження і моделювання великих об'ємів даних для виявлення невідомих до цього шаблонів (patterns) з метою досягнення переваг в бізнесі (визначення SAS Institute).

Data Mining – це процес, мета якого – знайти нові значущі кореляції, зразки і тенденції в результаті просівання великого об'єму бережених даних з використанням методик розпізнавання зразків плюс застосування статистичних і математичних методів (визначення Gartner Group).

«Mining» англійською означає «видобуток корисних копалин», а пошук закономірностей у величезній кількості даних дійсно схожий на цей процес.

Перш ніж використовувати технологію Data Mining, необхідно ретельно проаналізувати її проблеми :

- Data Mining не може замінити аналітика;
- не може складати розробки і експлуатації додатку Data Mining;
- потрібна підвищена кваліфікація користувача;
- витягання корисних відомостей неможливе без доброго розуміння суті даних;
- складність підготовки даних;
- висока вартість;
- вимога наявності достатньої кількості репрезентативних даних.

Data Mining тісно пов’язана з різними дисциплінами , що засновані на інформаційних технологіях та математичних методах обробки інформації (рис. 1.1).

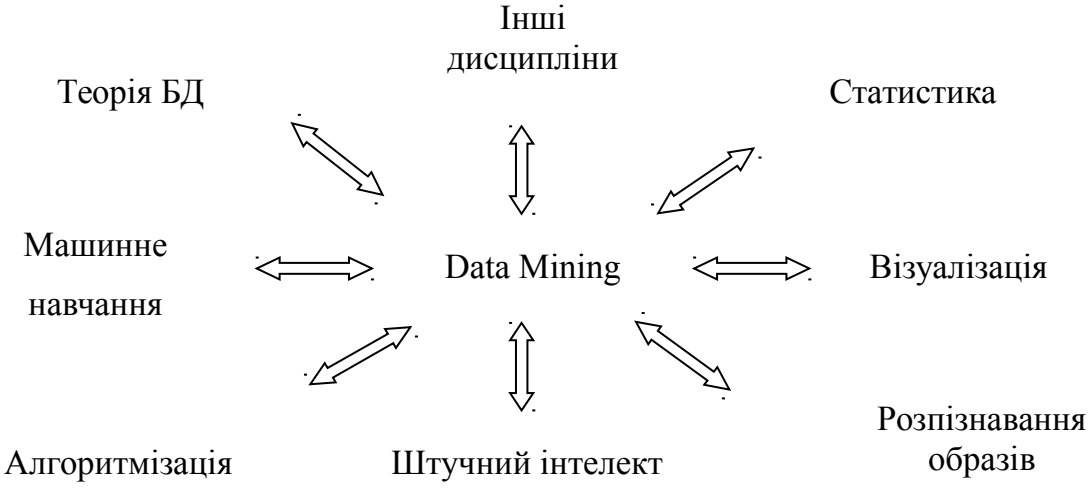


Рис. 1.1. Data Mining як мультідісциплінарна область

Кожний з напрямів, що сформували Data Mining, має свої особливості. Проведемо порівняння з деякими з них.

1.2 Порівняння статистики, машинного навчання і Data Mining

Статистика – це наука про методи збору даних, їх обробки і аналізу для виявлення закономірностей, властивих явищу, що вивчається.

Статистика є сукупністю методів планування експерименту, збору даних, їх уявлення і узагальнення, а також аналізу і отримання висновків на підставі цих даних.

Статистика оперує даними, що отримані в результаті спостережень або експериментів.

Перевагами є:

- більш ніж Data Mining, базується на теорії;
- більш зосереджується на перевірці гіпотез.

Єдиного визначення машинного навчання на сьогоднішній день немає.

Машинне навчання можна охарактеризувати як процес отримання програмою нових знань. Мітчелл в 1996 році дав таке визначення: «Машинне навчання – це наука, яка вивчає комп'ютерні алгоритми, автоматично що поліпшуються під час роботи».

Одним з найпопулярніших прикладів алгоритму машинного навчання є нейронні мережі.

Алгоритми машинного навчання є:

- більш евристичні;
- концентрується на поліпшенні роботи агентів навчання.

Переваги Data Mining:

- інтеграція теорії і евристик;
- сконцентрована на єдиному процесі аналізу даних, включає очищення даних, навчання, інтеграцію і візуалізацію результатів.

Методи Data Mining

Методи, що використовує технологія Data Mining можна розподілити на технологічні, статистичні та кібернетичні.

Таблиця 1.1

Методи Data Mining

Методи Data Mining	Характеристика
Технологічні методи	а) безпосереднє використання даних, або збереження даних. Методи цієї групи: кластерний аналіз, метод найближчого сусіда; б) виявлення і використання формалізованих закономірностей, або дистиляція шаблонів - логічні методи, методи візуалізації, методи крос-табуляції, методи, що засновані на рівняннях.
Статистичні методи	а) дескриптивний аналіз і опис вихідних даних; б) аналіз зв'язків (кореляційний і регресійний аналіз, факторний аналіз, дисперсійний аналіз); в) багатовимірний статистичний аналіз (компонентний аналіз, дискримінантний аналіз, багатовимірний регресійний аналіз, канонічні кореляції і ін.); г) аналіз тимчасових рядів (динамічні моделі і прогнозування).
Кібернетичні методи	а)штучні нейронні мережі (розпізнавання, кластеризація, прогноз); б) еволюційне програмування (в т.ч. алгоритми методу групового обліку аргументів); в) генетичні алгоритми (оптимізація); г) асоціативний алгоритм; г) нечітка логіка; д) дерева рішень; є) системи обробки експертних знань.

Відмінності Data Mining від інших методів аналізу даних

Традиційні методи аналізу даних в основному орієнтовані на перевірку наперед сформульованих гіпотез (статистичні методи) і на «грубий розвідувальний аналіз», що становить основу оперативної аналітичної обробки даних (Online Analytical Processing, OLAP), тоді як одне з основних положень Data Mining – пошук неочевидних закономірностей. Інструменти Data Mining можуть знаходити такі

закономірності самостійно і також самостійно будувати гіпотези про взаємозв'язки. Оскільки саме формулювання гіпотези щодо залежності є найскладнішою задачею, перевага Data Mining в порівнянні з іншими методами аналізу є очевидною.

Більшість статистичних методів для виявлення взаємозв'язків в даних використовує концепцію усереднювання по вибірці, що приводить до операцій над неіснуючими величинами, тоді як Data Mining оперує реальними значеннями.

OLAP більше підходить для розуміння ретроспективних даних, Data Mining спирається на ретроспективні дані для отримання відповідей на питання про майбутнє.

1.3 Алгоритм кластеризації

Кластеризація (або кластерний аналіз) - це завдання розбиття множини об'єктів на групи, які називаються кластерами. Усередині кожної групи повинні виявитися «схожі» об'єкти, а об'єкти різних групи повинні бути як можна більш відмінні. Головна відмінність кластеризації від класифікації полягає в тому, що перелік груп чітко не заданий і визначається в процесі роботи алгоритму.

Застосування кластерного аналізу в загальному вигляді зводиться до наступних етапів:

Відбір вибірки об'єктів для кластеризації.

Визначення безлічі змінних, за якими будуть оцінюватися об'єкти у вибірці. При необхідності - нормалізація значень змінних.

Обчислення значень міри схожості між об'єктами.

Застосування методу кластерного аналізу для створення груп схожих об'єктів (кластерів).

Представлення результатів аналізу.

Після отримання та аналізу результатів можливе корегування обраної метрики і методу кластеризації до отримання оптимального результату.

Заходи відстаней

Отже, як же визначати «схожість» об'єктів? Для початку потрібно скласти вектор характеристик для кожного об'єкта - як правило, це набір числових значень, наприклад, зростання-вага людини. Однак існують також алгоритми, що працюють з якісними (т.зв. категорійними) характеристиками.

Після того, як ми визначили вектор характеристик, можна провести нормалізацію, щоб всі компоненти давали однаковий внесок при розрахунку «відстані». У процесі нормалізації всі значення наводяться до деякого діапазону, наприклад, [-1, -1] або [0, 1].

Нарешті, для кожної пари об'єктів вимірюється «відстань» між ними - ступінь схожості. Існує безліч метрик, ось лише основні з них:

1. Евклідова відстань

Найбільш поширена функція відстані. Являє собою геометричним відстанню в багатовимірному просторі:

$$\rho(x, x') = \sqrt{\sum_{i}^{n} (x_i - x_i')^2}$$

2. Квадрат евклідова відстані

Застосовується для додання більшої ваги більш віддаленим один від одного об'єктам. Це відстань обчислюється таким чином:

$$\rho(x, x') = \sum_{i}^{n} (x_i - x_i')^2$$

3. Відстань міських кварталів (Манхеттенський відстань)

Це відстань є середнім різниць по координатах. У більшості випадків ця міра відстані призводить до таких же результатів, як і для звичайного відстані Евкліда. Однак для цього заходу вплив окремих великих різниць (викидів) зменшується (тому вони не зводяться в квадрат). Формула для розрахунку манхеттенського відстані:

$$\rho(x, x') = \sum_{i}^{n} | x_i - x_i' |$$

4. Відстань Чебишева

Ця відстань може виявитися корисним, коли потрібно визначити два об'єкти як «різні», якщо вони розрізняються за якої-небудь однієї координаті. Відстань Чебишева обчислюється за формулою:

$$\rho(x, x') = \max(|\ x_i - x_i^{'}\ |)$$

5. Степенна відстань

Застосовується у випадку, коли необхідно збільшити або зменшити вагу, що відноситься до розмірності, для якої відповідні об'єкти сильно відрізняються. Статечне відстань обчислюється за наступною формулою:

$$\rho(x, x') = \sqrt[r]{\sum_{i}^{n}(x_i - x_i^{'})^p},$$

де r і p - параметри, які визначаються користувачем. Параметр p відповідальний за поступове зважування різниць по окремих координатам, параметр r відповідальний за прогресивне зважування великих відстаней між об'єктами. Якщо обидва параметра - r і p - дорівнюють двом, то ця відстань збігається з відстанню Евкліда.

Вибір метрики повністю лежить на дослідника, оскільки результати кластеризації можуть істотно відрізнятися при використанні різних заходів.

Класифікація алгоритмів

Для себе я виділив дві основні класифікації алгоритмів кластеризації.

1. Ієрархічні і плоскі.

Ієрархічні алгоритми (також звані алгоритмами таксономії) будують не одне розбиття вибірки на непересічні кластери, а систему вкладених розбиття. Т.ч. на виході ми отримуємо дерево кластерів, коренем якого є вся вибірка, а листям - найбільш дрібні кластера.

Плоскі алгоритми будують одне розбиття об'єктів на кластери.

2. Чіткі і нечіткі.

Чіткі (або непересічні) алгоритми кожному об'єкту вибірки ставлять у відповідність номер кластера, тобто кожен об'єкт належить тільки одного кластеру. Нечіткі (або пересічні) алгоритми кожному об'єкту ставлять у відповідність набір

речових значень, що показують ступінь відносини об'єкта до кластерів. Тобто кожен об'єкт відноситься до кожного кластеру з деякою ймовірністю.

Об'єднання кластерів

У разі використання ієрархічних алгоритмів постає питання, як об'єднувати між собою кластера, як вираховувати «відстані» між ними. Існує кілька метрик:

1. Одиночна зв'язок (відстані найближчого сусіда)

У цьому методі відстань між двома кластерами визначається відстанню між двома найбільш близькими об'єктами (найближчими сусідами) в різних кластерах. Результуючі кластери мають тенденцію об'єднуватися в ланцюжка.

2. Повный зв'язок (відстань найбільш віддалених сусідів)

У цьому методі відстані між кластерами визначаються найбільшою відстанню між будь-якими двома об'єктами в різних кластерах (тобто найбільш віддаленими сусідами). Цей метод зазвичай працює дуже добре, коли об'єкти походять з окремих груп. Якщо ж кластери мають видовжену форму або їх природний тип є «цепочечном», то цей метод непридатний.

3. Незважене попарне середнє

У цьому методі відстань між двома різними кластерами обчислюється як середня відстань між усіма парами об'єктів в них. Метод ефективний, коли об'єкти формують різні групи, проте він працює однаково добре і у випадках протяжних («цепочечного» типу) кластерів.

4. Виважена попарне середнє

Метод ідентичний методу невиваженого попарного середнього, за винятком того, що при обчисленнях розмір відповідних кластерів (тобто число об'єктів, що містяться в них) використовується як вагового коефіцієнта. Тому даний метод повинен бути використаний, коли передбачаються нерівні розміри кластерів.

5. Незвішаний центроїдний метод

У цьому методі відстань між двома кластерами визначається як відстань між їх центрами тяжкості.

6. Зважений центроїдний метод (медіана)

Цей метод ідентичний попередньому, за винятком того, що при обчисленнях використовуються ваги для обліку різниці між розмірами кластерів. Тому, якщо є або підозрюються значні відмінності в розмірах кластерів, цей метод виявляється переважно попереднього.

Огляд алгоритмів

Алгоритми ієрархічної кластеризації

Серед алгоритмів ієрархічної кластеризації виділяються два основних типи: висхідні та низхідні алгоритми. Спадні алгоритми працюють за принципом «зверху-вниз»: на початку всі об'єкти поміщаються в один кластер, який потім розбивається на все більш дрібні кластери. Більш поширені висхідні алгоритми, які на початку роботи поміщають кожен об'єкт в окремий кластер, а потім об'єднують кластери у все більш крупні, поки всі об'єкти вибірки не будуть міститися в одному кластері. Таким чином будується система вкладених розбиття. Результати таких алгоритмів зазвичай представляють у вигляді дерева - дендрограмми. Класичний приклад такого дерева - класифікація тварин і рослин.

Для обчислення відстаней між кластерами частіше все користуються двома відстанями: одиночної зв'язком або повним зв'язком (див. Огляд заходів відстаней між кластерами).

До недоліку ієрархічних алгоритмів можна віднести систему повних разбиений, яка може бути зайвої в контексті розв'язуваної задачі.

Алгоритми квадратичної помилки

Задачу кластеризації можна розглядати як побудова оптимального розбиття об'єктів на групи. При цьому оптимальність може бути визначена як вимога мінімізації середньоквадратичної помилки розбиття:

$$e^2(X,L) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \| x_i^{(j)} - c_j \|^2$$

де cj - «центр мас» кластера j (точка з середніми значеннями характеристик для даного кластера).

Алгоритми квадратичної помилки відносяться до типу плоских алгоритмів. Найпоширенішим алгоритмом цієї категорії є метод k-середніх. Цей алгоритм будує задане число кластерів, розташованих якнайдалі один від одного. Робота алгоритму ділиться на кілька етапів:

1. Випадково вибрати k точок, які є початковими «центрами мас» кластерів.
2. Віднести кожен об'єкт до кластеру з найближчим «центром мас».
3. Перерахувати «центри мас» кластерів згідно їх поточним складом.
4. Якщо критерій зупинки алгоритму не задоволений, повернутися до п. 2.

В якості критерію зупинки роботи алгоритму зазвичай вибирають мінімальне зміна среднеквадратической помилки. Так само можливе зупиняти роботу алгоритму, якщо на кроці 2 не було об'єктів, що перемістилися з кластера в кластер.

До недоліків даного алгоритму можна віднести необхідність задавати кількість кластерів для розбиття.

Нечіткі алгоритми

Найбільш популярним алгоритмом нечіткої кластеризації є алгоритм c-середніх (c-means). Він являє собою модифікацію методу k-середніх. Кроки роботи алгоритму:

1. Вибрати початкове нечітке розбиття n об'єктів на k кластерів шляхом вибору матриці приналежності U розміру nx k.

2. Використовуючи матрицю U, знайти значення критерію нечіткої помилки:

$$E^2(X,U) = \sum_{i=1}^{N} \sum_{k=1}^{K} U_{ik} \| x_i^{(k)} - c_k \|^2$$

,

де ck - «центр мас» нечіткого кластера k:

$$c_k = \sum_{i=1}^{N} U_k x_i$$

.

3. Перегрупувати об'єкти з метою зменшення цього значення критерію нечіткої помилки.

4. Повертатися в п. 2 доти, поки зміни матриці U не стануть незначними.

Цей алгоритм може не підійти, якщо заздалегідь невідомо число кластерів, або необхідно однозначно віднести кожен об'єкт до одного кластеру.

Алгоритми, засновані на теорії графів

Суть таких алгоритмів полягає в тому, що вибірка об'єктів подається у вигляді графа G = (V, E), вершинам якого відповідають об'єкти, а ребра мають вагу, рівний «віддалі» між об'єктами. Перевагою графових алгоритмів кластеризації є наочність, відносна простота реалізації і можливість вносенія різних удосконалень, засновані на геометричних міркуваннях. Основними алгоритмам є алгоритм виділення зв'язкових компонент, алгоритм побудови мінімального покриває (остовного) дерева і алгоритм пошарової кластеризації.

Алгоритм виділення зв'язкових компонент

В алгоритмі виділення зв'язкових компонент задається вхідний параметр R і в графі видаляються всі ребра, для яких «відстані» більше R. Сполученими залишаються тільки найбільш близькі пари об'єктів. Сенс алгоритму полягає в тому, щоб підібрати таке значення R, що лежить в діапазон всіх «відстаней», при якому граф «розвалиться» на кілька зв'язкових компонент. Отримані компоненти і є кластери.

Для підбору параметра R зазвичай будується гістограма розподілів попарних відстаней. У завданнях з добре вираженою кластерної структурою даних на гістограмі буде два піки - один відповідає внутрікластерним відстаням, другий - межкластерним відстані. Параметр R підбирається із зони мінімуму між цими піками. При цьому управляти кількістю кластерів за допомогою порога відстані досить важко.

Алгоритм мінімального покриває дерева

Алгоритм мінімального покриває дерева спочатку будує на графі мінімальне покриває дерево, а потім послідовно видаляє ребра з найбільшою вагою. На малюнку зображено мінімальне покривюче дерево, отримане для дев'яти об'єктів.

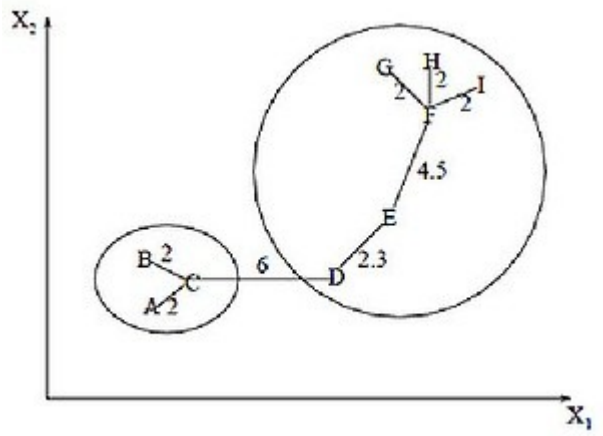


Рис. 1.2. Мінімальне покриваюче дерево

Шляхом видалення зв'язку, поміченої CD, з довжиною рівною 6 одиницям (ребро з максимальною відстанню), одержуємо два кластери: {A, B, C} і {D, E, F, G, H, I}. Другий кластер в подальшому може бути розділений ще на два кластери шляхом видалення ребра EF, яке має довжину, рівну 4,5 одиницям.

Пошарова кластеризація

Алгоритм пошарової кластеризації заснований на виділенні зв'язкових компонент графа на деякому рівні відстаней між об'єктами (вершинами). Рівень відстані задається порогом відстані c. Наприклад, якщо відстань між об'єктами $0 \le \rho(x, x') \le 1$, то $0 \le c \le 1$.

Алгоритм пошарової кластеризації формує послідовність подграфов графа G, які відображають ієрархічні зв'язки між кластерами:

$G^0 \subseteq G^1 \subseteq \ldots \subseteq G^m$,

де $G^t = (V, E^t)$- граф на рівні c^t,

c^t - t-ий поріг відстані,

m - кількість рівнів ієрархії,

$G^0 = (V, o)$, o - порожня множина ребер графа, одержуване при t0 = 1,

$G^m = G$, тобто граф об'єктів без обмежень на відстань (довжину ребер графа), оскільки tm = 1.

За допомогою зміни порогів відстані $\{c^0, \ldots, c^m\}$, где $0 = c^0 < c^1 < \ldots < c^m = 1$, можливо контролювати глибину ієрархії одержуваних кластерів. Таким чином, алгоритм пошарової кластеризації здатний створювати як плоске розбивка даних, так і ієрархічне.

1.4 Data mining як частина системи аналітичної обробки інформації

Сховища даних

Інформаційні системи сучасних підприємств часто організовані так, щоб мінімізувати час введення і коректування даних, тобто організовані не оптимально з погляду проектування бази даних. Такий підхід ускладнює доступ до історичних (архівним) даних. Зміни структур в базах даних інформаційних систем дуже трудомісткі, а іноді просто неможливі.

В той же час, для успішного ведення сучасного бізнесу необхідна актуальна інформація, що надається в зручному для аналізу вигляді і в реальному масштабі часу. Доступність такої інформації дозволяє, як оцінювати поточне положення справ, так і робити прогнози на майбутнє, отже, ухвалювати більш зважені і обґрунтовані рішення. До того ж, основою для ухвалення рішень повинні бути реальні дані.

Якщо дані зберігаються в базах даних різних інформаційних систем підприємства, при їх аналізі виникає ряд складнощів, зокрема, значно зростає час, необхідний для обробки запитів; можуть виникати проблеми з підтримкою різних форматів даних, а також з їх кодуванням; неможливість аналізу тривалих рядів ретроспективних даних і т.д.

Ця проблема розв'язується шляхом створення сховища даних. Задачею такого сховища є інтеграція, актуалізація і узгодження оперативних даних з різнорідних джерел для формування єдиного несуперечливого погляду на об'єкт управління в

цілому. На основі сховищ даних можливо складання всілякої звітності, а також проведення оперативної аналітичної обробки і Data Mining.

Тоді загальна схема інформаційного сховища буде виглядати наступнім чином:

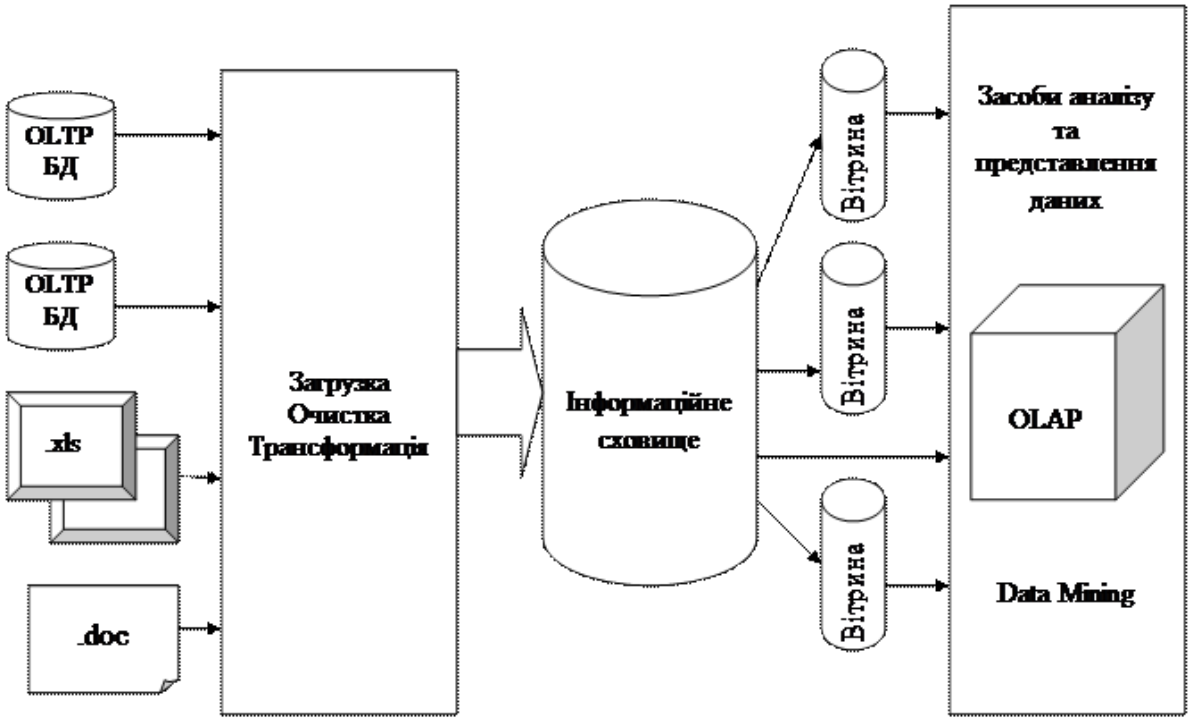


Рис. 1.4. Структура інформаційної системи

Біл Інмон (Bill Inmon) визначає сховища даних як "наочно орієнтовані, інтегровані, немінливі, підтримуючі хронологію набори даних, організовані з метою підтримки управління" і покликані виступати в ролі "єдиного і єдиного джерела істини", яке забезпечує менеджерів і аналітиків достовірною інформацією, необхідною для оперативного аналізу і ухвалення рішень[3].

Наочна орієнтація сховища даних означає, що дані з'єднані в категорії і зберігаються відповідно областям, які вони описують, а не застосуванням, що їх використовують.

Інтегрованість означає, що дані задовольняють вимогам всього підприємства, а не однієї функції бізнесу. Цим сховище даних гарантує, що однакові звіти, що згенерували для різних аналітиків, міститимуть однакові результати.

Прив'язка до часу означає, що сховище можна розглядати як сукупність "історичних даних": можливо відновлення даних на будь-який момент часу. Атрибут часу явно присутній в структурах сховища даних.

Незмінність означає, що, потрапивши один раз в сховищі, дані там зберігаються і не змінюються. Дані в сховищі можуть лише додаватися.

Річард Хакаторн, інший основоположник цієї концепції, писав, що мета Сховищ Даних – забезпечити для організації "єдиний образ існуючої реальності"[3].

Іншими словами, сховище даних є своєрідним накопичувачем інформації про діяльність підприємства.

Дані в сховищі представлені у вигляді багатовимірних структур під назвою "зірка" або "сніжинка".

Організація інформаційного сховища в реалізації бази даних. Схеми зірка та сніжинка

Схема типу зірки (Star Schema) – схема реляційної бази даних, що служить для підтримки багатовимірного представлення даних, які в ній зберігаються.

Особливості ROLAP-схеми типу "зірка":

а) одна таблиця фактів (fact table), яка сильно денормалізована. Є центральною в схемі, може складатися з мільйонів рядків і містить підсумовуванні або фактичні дані, за допомогою яких можна відповісти на різні питання;

б) декілька денормалізованих таблиць вимірювань (dimensional table). Мають меншу кількість рядків, ніж таблиці фактів, і містять описову інформацію. Ці таблиці дозволяють користувачу швидко переходити від таблиці фактів до додаткової інформації;

в) таблиця фактів і таблиці розмірності зв'язані ідентифікуючими зв'язками, при цьому первинні ключі таблиці розмірності мігрують в таблицю фактів як зовнішні ключі. Первинний ключ таблиці факту цілком складається з первинних ключів всіх таблиць розмірності;

ґ) агреговані дані зберігаються спільно з початковими.

Рис. 1.5. Схема «зірка»

Схема типу сніжинки (Snowflake Schema) – схема реляційної бази даних, яка служить для підтримки багатовимірного представлення даних, що в ній знаходяться, є різновидом схеми типу "зірка" (Star Schema).

Особливості ROLAP-схеми типу "сніжинка":

а) одна таблиця фактів (fact table), яка сильно денормалізована. Є центральною в схемі, може складатися з мільйонів рядків і містити підсумовуванні або фактичні дані, за допомогою яких можна відповісти на різні питання;

б) декілька таблиць вимірювань (dimensional table), які нормалізовані на відміну від схеми "зірка". Мають меншу кількість рядків, ніж таблиці фактів, і містять описову інформацію. Ці таблиці дозволяють користувачу швидко переходити від таблиці фактів до додаткової інформації. Первинні ключі в них складаються з єдиного атрибута (відповідають єдиному елементу вимірювання);

в) таблиця фактів і таблиці розмірності зв'язані ідентифікуючими зв'язками, при цьому первинні ключі таблиці розмірності мігрують в таблицю фактів як зовнішні

ключі. Первинний ключ таблиці факту цілком складається з первинних ключів всіх таблиць розмірності;

ґ) в схемі "сніжинка" агреговані дані можуть зберігатися окремо від початкових.

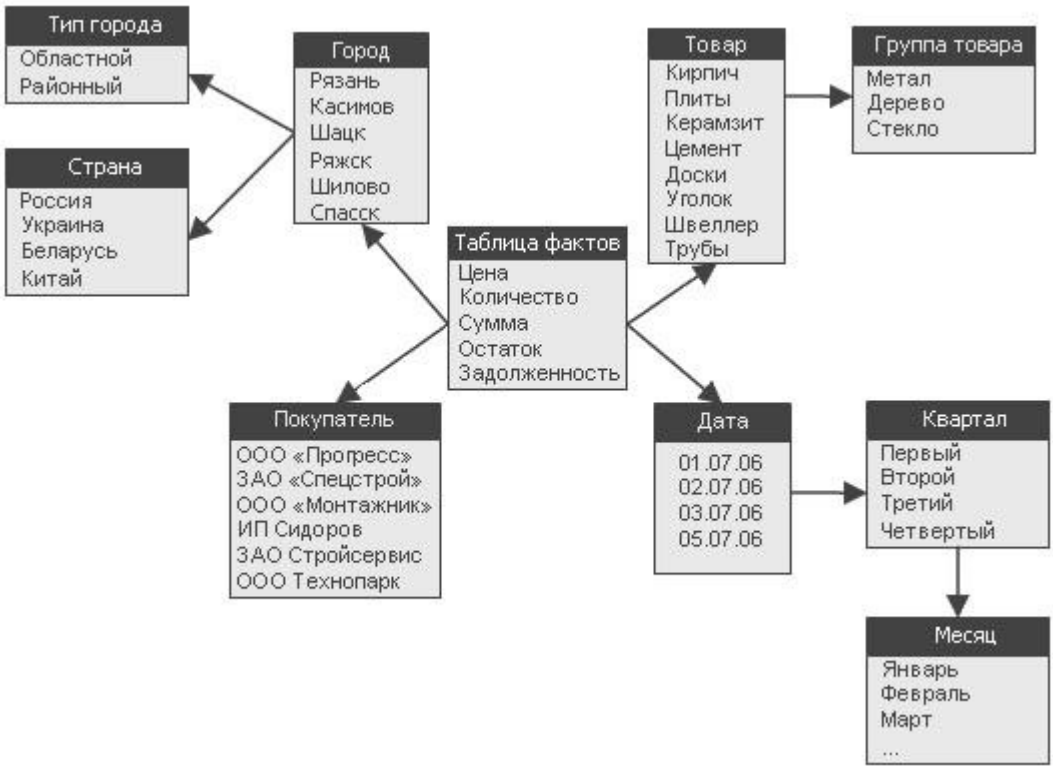


Рис. 1.6. Схема «сніжинка»

OLAP-системи

В основі концепції OLAP, або оперативної аналітичної обробки даних (On-Line Analytical Processing), лежить багатовимірне концептуальне представлення даних (Multidimensional conceptual view).

Термін OLAP введений Коддом (E. F. Codd) в 1993 році. Головна ідея даної системи полягає в побудові багатовимірних таблиць, які можуть бути доступний для запитів користувачів. Ці багатовимірні таблиці або так звані багатовимірні куби будуються на основі початкових і агрегованих даних. І початкові, і агреговані дані для багатовимірних таблиць можуть зберігатися як в реляційних, так і в багатовимірних базах даних. Взаємодіючи з OLAP-системою, користувач може здійснювати гнучкий перегляд інформації, одержувати різні зрізи даних, виконувати аналітичні операції

деталізації, згортки, крізного розподілу, порівняння в часі. Вся робота з OLAP-системою відбувається в термінах наочної області[3].

Існує три способи зберігання даних в OLAP-системах або три архітектура OLAP - серверів:

- MOLAP (Multidimensional OLAP);

- ROLAP (Relational OLAP);

- HOLAP (Hybrid OLAP).

Таким чином, згідно цієї класифікації OLAP-продукти можуть бути представлений трьома класами систем:

- у разі MOLAP, початкові і багатовимірні дані зберігаються в багатовимірній БД або в багатовимірному локальному кубі;

- в ROLAP-продуктах початкові дані зберігаються в реляційних БД або в плоских локальних таблицях на файл-сервері. Агрегатні дані можуть поміщатися в службові таблиці в тій же БД;

- у разі використовування гібридної архітектури, тобто в HOLAP-продуктах, початкові дані залишаються в реляційній базі, а агрегати розміщуються в багатовимірній.

Логічним уявленням є багатовимірний куб — це набір зв'язаних заходів і вимірювань, які використовуються для аналізу даних[3].

OLAP-куб підтримує всі багатовимірні операції: довільне розміщення вимірювань і фактів, фільтрація, сортування, угрупування, різні способи агрегації і деталізації. Дані відображаються у вигляді крос-таблиць і крос-діаграм, всі операції відбуваються "на льоту", методи маніпулювання інтуїтивно зрозумілі.

OLAP-куб – могутній інструмент дослідження, що дозволяє проводити розвідувальний і порівняльний аналіз, виявляти тенденції, знаходити сезонність і тренд, визначати кращі і гірші товарні позиції, розраховувати їх частки у продажі.

Інтеграція OLAP і Data Mining

Обидві технології можна розглядати як складові частини процесу підтримки ухвалення рішень. Проте ці технології як би рухаються у різних напрямах: OLAP зосереджує увагу виключно на забезпеченні доступу до багатовимірних даних, а методи Data Mining в більшості випадків працюють з плоскими одновимірними таблицями і реляційними даними.

Інтеграція технологій OLAP і Data Mining "збагатила" функціональність і однієї, і іншої технології. Ці два види аналізу повинні бути тісно з'єднано, щоб інтегрована технологія могла забезпечувати одночасно багатовимірний доступ і пошук закономірностей.

Засіб багатовимірного інтелектуального аналізу даних повинен знаходити закономірності як в тих, що деталізуються, так і в агрегованих з різним ступенем узагальнення даних. Аналіз багатовимірних даних повинен будуватися над гіперкубом спеціального вигляду, вічка якого містять не довільні чисельні значення (кількість подій, об'єм продажів, сума зібраних податків), а числа, що визначають вірогідність відповідного поєднання значень атрибутів. Проекції такого гіперкуба (що виключають з розгляду окремі вимірювання) також повинні досліджуватися на предмет пошуку закономірностей. J. Han пропонує ще більш просту назву - "OLAP Mining" і висуває декілька варіантів інтеграції двох технологій[3]:

а) "Cubing then mining". Можливість виконання інтелектуального аналізу повинна забезпечуватися над будь-яким результатом запиту до багатовимірного концептуального уявлення, тобто над будь - яким фрагментом будь - якої проекції гіперкуба показників;

б) "Mining then cubing". Подібно даним, витягнутим з сховища, результати інтелектуального аналізу повинні представлятися в гіперкубічній формі для подальшого багатовимірного аналізу;

в) "Cubing while mining". Цей гнучкий спосіб інтеграції дозволяє автоматично активізувати однотипні механізми інтелектуальної обробки над результатом кожного кроку багатовимірного аналізу (переходу між рівнями узагальнення, витягання нового фрагмента гіперкуба і т.д.).

На сьогоднішній день небагато виробників реалізують Data Mining для багатовимірних даних. Крім того, деякі методи Data Mining, наприклад, метод найближчих сусідів або байєсівськая класифікація, через їх нездатність працювати з агрегованими даними незастосовні до багатовимірних даних.

2. СТРУКТУРА ІНФОРМАЦІЙНОГО СХОВИЩА ДЛЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ

2.1 Характеристика джерела даних для інформаційного сховища

У даній роботі за основу була узята БД-зразок Microsoft – Adventure Works[18]. Проект Adventure Works описує роботу виробника велосипедів - компанії "Adventure Works Cycles". Компанія займається виробництвом і реалізацією велосипедів з металевих і композиційних матеріалів на території Північної Америки, Європи і Азії. Головне виробництво, яке має в своєму розпорядженні 500 співробітників, знаходиться в місті Bothell, штат Вашингтон. Декілька регіональних офісів знаходяться безпосередньо на території ринків збуту.

Компанія реалізує продукцію оптом для спеціалізованих магазинів і на роздріб через Інтернет. Для вирішення демонстраційних завдань ми використовуватимемо в базі AdventureWorks дані об інтернет продажах, оскільки вони містять дані, які добре підходять для аналізу.

На рисунку 4.1 представлена транзакційна бази даних AdventureWorks, відділу продаж, яка містить наступні таблиці:

- таблиця SalesTaxRate – в якій містяться податкові ставки, вживані в областях або країнах і регіонах, в яких компанія Adventure Works Cycles здійснює ділову активність;

- таблиця ShoppingCartItem – містить замовлення клієнтів через інтернет до моменту виконання або відміни;

- таблиця SpecialOfferProduct – в якій приведені знижки на різні види (найменування) продукції;

- таблиця SpecialOffer – в якій містяться знижки на продаж;

- таблиця CountryRegionCurrency – зіставляє коди валют по стандартах Міжнародної організації по стандартизації (ISO) і коди країн або регіонів;

- таблиця Currency – містить описи валют по стандартах Міжнародної організації стандартизації (ISO);

- таблиця SalesTerritoryHistory – у таблиці відстежуються переміщення комерційних представників в інші комерційні території;

- таблиця SalesTerritory – в якій містяться території продажів, які обслуговуються групами продажів Adventure Works Cycles;

- таблиця SalesPersonQuotaHistory – містить зведення по історії продажів для комерційних представників;

- таблиця Store – містить список замовників, торгівельних посередників, що купують продукти в Adventure Works;

- таблиця CurrencyRate – містить курси обміну валюти;

- таблиця SalesPerson – містить поточні відомості про продажі для комерційних представників;

- таблиця SalesOrderDetail – містить окремі продукти, пов'язані з певним замовленням на продаж. Замовлення на продаж може містити замовлення на декілька продуктів;

- таблиця SalesOrderHeader – містить відомості про загальне або батьківське замовлення на продаж;

- таблиця Customer – містить поточні відомості про замовника. Клієнти розбиті на категорії по типах — приватний споживач або магазин роздрібної торгівлі;

- таблиця StoreContact – в якій зіставляються магазини і їх службовці, з якими безпосередньо співробітничають торгівельні представники компанії Adventure Works Cycles;

- таблиця SalesReason – в якій містяться можливі причини придбання клієнтом певного продукту;

- таблиця SalesOrderHeaderSalesReason – в якій замовлення на продаж зіставляються з кодами причин продажів;

- таблиця CustomerAddress – зіставляє замовників з їх адресами. Наприклад, замовник може мати різні адреси для виставляння рахунків і доставки.

2.2. Проектування сховищ даних

Методи інтелектуального аналізу інформації часто розглядають як природний розвиток концепції сховищ даних. Головна відмінність сховища від бази даних полягає в тому, що їх створення і експлуатація переслідують різну мету. База даних відіграє роль помічника в оперативному управлінні організацією. Це щоденні задачі отримання актуальної інформації: бухгалтерські звітності, облік договорів, тощо. Сховище даних накопичує всі необхідні дані для здійснення задач стратегічного управління в середньостроковому і довгостроковому періоді. Наприклад, продаж товару і генерація рахунку проводяться з використанням бази даних, а аналіз динаміки продажів за декілька років, що дозволяє спланувати роботу з постачальниками - за допомогою сховища даних.

Сховище даних (Data Warehouse) - це систематизована інформація з різнорідних джерел, яка є необхідною для обробки з метою ухвалення стратегічно важливих рішень

Сховище будується на основі клієнт-серверної архітектури, СУБД і утиліт підтримки прийняття рішень. Дані, що надходять у сховище, стають доступні тільки для читання.

Властивості сховища даних;

предметна орієнтація (інформацію організовано відповідно до основних аспектів діяльності);

інтегрованість даних (дані в сховище надходять з різних джерел і відповідно агрегуються);

стабільність, інваріантність у часі (записи в DW ніколи не змінюються, являючи собою відбитки даних, зроблені у певний час);

мінімізація збитковості інформації (перед завантаженням у сховища дані фільтруються, зберігаються у певній послідовності, а також формується деяка підсумкова інформація).

В сховищах даних надмірність даних є мінімальною (приблизно 1%), оскільки:

при завантаженні у сховище дані сортуються і фільтруються;

інформація у сховищах зберігається в хронологічному порядку, що майже повністю виключає перекриття даних;

при завантаженні у сховище дані зводяться до єдиного формату, включаючи обчислення підсумкових (агрегованих) показників.

Сервери багатовимірних баз даних можуть зберігати дані по-різному, крім агрегованих показників формується ще й додаткова інформація: поля часу, дати; адресні посилання, таблиці метаданих тощо. Це приводить до значного збільшення інформації. Вхідний масив розміром 200 Mb може розростись до об'єму 5 Gb. Сховище даних повинне бути оптимально організованою базою даних, яка забезпечує максимально швидкий і оперативний пошук інформації.

Вітрина даних - це спрощений варіант сховища даних, що містить лише тематично орієнтовані, агреговані дані

Глобальне сховище даних складається з трьох рівнів:

1) сховище агрегованих даних;

2) вітрини даних, які базуються на інформації зі сховища даних;

3) клієнтські робочі місця, на яких встановлено засоби оперативного аналізу даних.

У розпорядженні виробників прикладних програмних засобів є три різні технології роботи з базами даних:

DAO (Data Access Objects) - доступ до локальних баз даних;

RDO (Remote Data Objects) - доступ до віддалених баз даних;

AD(ActiveX Data Objects) - доступ до Widows-додатків через Інтернет. В основному використовується з міркувань безпеки.

Одним з перспективних напрямів удосконалення доступу до даних є гнучке конфігурування системи, коли розподіл між клієнтською і серверною частинами можливий за допомогою використання механізму віддалених

процедур.

Поряд з потоками даних існують і потоки метаданих, які розміщуються в депозитарії. Він дає змогу визначити семантичну структуру додатка у вигляді опису термінів предметної галузі, їхні взаємозв'язки й атрибути.

Метадані - це дані про дані, які визначають джерело, приймач та алгоритм трансформації даних під час перенесення їх від джерела до приймача

Метадані містять:

описи структур даних та їхніх взаємозв'язків;

інформацію про джерела даних і про ступінь їх вірогідності;

інформацію про власників даних, права доступу;

схему перетворення стовпців вхідних таблиць у стовпці кінцевих таблиць;

правила підсумовування, консолідації та агрегування даних;

інформацію про періодичність оновлення даних;

каталог використаних таблиць, стовпців та ключів;

фізичні атрибути стовпців;

кількість табличних рядків та обсяг даних;

часові ярлики (дата та час створення/модифікації записів);

статистичні оцінки часу виконання запитів.

Контроль модифікації (versioning) полягає у властивості метаданих відслідковувати зміни в структурі даних та їх значення в часі.

Функціональна архітектура сховища даних містить наступні компоненти:

сховище даних;

клієнтська частина системи (дизайнери сховища, засоби розробки додатків, засоби адміністрування, інструменти аналізу даних, завантаження словника метаданих з XML-файлу у сховище і експорт його зі сховища в XML-файл;

сервер обміну даними (Data Exchange Server) - набір програм імпорту/експорту даних зі сховища й каталогів для організації обміну даними із зовнішніми OLTP-системами;

бібліотеки прикладних класів: ACL (Application Class Library), VCL (Visual Component Library), Win Lite.

Наповнення інформаційних сховищ відбувається в декілька етапів:

екстракція (витяг) - імпорт даних у сховище з інформаційних підсистем, виробничих відділів та інших джерел;

трансформація - консолідування, агрегування даних, розбиття їх на фракції, коригування та трансформування у відповідні формати;

завантаження - у сховище, синхронізація з датою або зовнішніми подіями.

Обслуговування інформаційних сховищ полягає в: копіюванні баз даних, налаштуванні, тиражуванні, надсиланні застарілих баз даних до архіву, управлінні правами користувачів, створенні та редагуванні графічних діаграм баз даних, тощо.

Типи архівації у сховищах поділяють на:

звичайна;

копіювальна;

додаткова;

диференціальна;

щоденна.

Архівні магнітні носії зберігають у вогнетривких сейфах або за межами обчислювального центру. Крім того, розробляється план архівації компонентів сервера баз даних. Сучасні сервери автоматично підтримують копію свого каталогу на кожному сервері вузла. Цей процес називається реплікацією каталогів (directory replication).

Звичайна архівація каталогів на всіх серверах здійснюється раз на тиждень у вихідні дні, а диференціальна - щодня в робочі .дні. У річному архіві, як правило, зберігаються дані останнього тижня місяця. Усі зміни в каталозі сервера, а також в

особистих і загальних сховищах записуються у файли, які називаються журналами трансакцій (transaction log files).

Під час виконання додаткової архівації каталогу або інформаційного сховища архівуванню підлягають лише журнали трансакцій.

Для ефективної роботи зі сховищем даних, необхідно зібрати максимум інформації про процес. Наприклад, для прогнозування обсягів продажів можуть бути використані бази даних облікових систем компанії, маркетингові дані, відгуки клієнтів, дослідження конкурентів і т.п.

Необхідною для прогнозу є наступна інформація:

хронологія продажів;

стан складу на кожний день - якщо спад продажів буде пов'язаний із відсутністю товару на складі, а не через відсутність попиту;

відомості про ціни конкурентів;

зміни у законодавстві;

загальний стан ринку;

курс долара, інфляція;

відомості про рекламу;

відомості про відношення до продукції клієнтів;

різного роду специфічну інформацію. Наприклад, для продавців морозива - температуру, а для фармакологічних складів -санітарно-епідеміологічний стан, тощо.

Проблема полягає в тому, що зазвичай в системах оперативного обліку більша частина цієї інформації відсутня, а наявна - неповна або спотворена. Кращим варіантом в цьому випадку буде створення сховища даних, куди б з певною заданою періодичністю надходила вся необхідна інформація, заздалегідь систематизована і очищена (рис.8).

Рис. 2.1. Приклад сховища даних

Ефективна архітектура сховища даних організовується таким чином, щоб бути складовою частиною інформаційної системи управління підприємством.

Найбільш поширений випадок, коли сховище організовано за типом "зірка", де в центрі розміщуються факти і агрегатні дані, а "променями" є виміри. Кожна "зірка" описує певну дію, наприклад, продаж товару, його відвантаження, надходження коштів й інше:

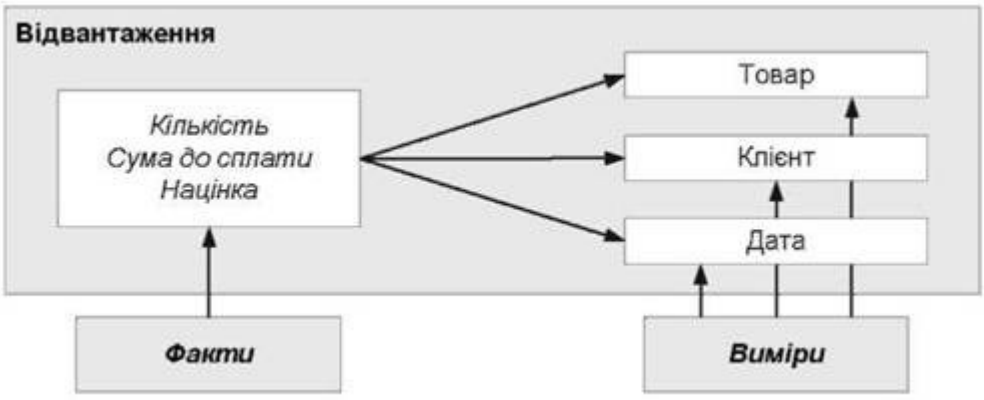


Рис.2.2. Схема організації сховища даних за типом "зірка"

Як правило, дані копіюються в сховище з оперативних баз даних і інших джерел відповідно до певних правил.

2.3 Структура інформаційного сховища

Для подальшого інтелектуального аналізу було розроблено структуру інформаційного сховища на базі схеми «сніжинка». На рисунку приведена логічна схема інформаційного сховища.

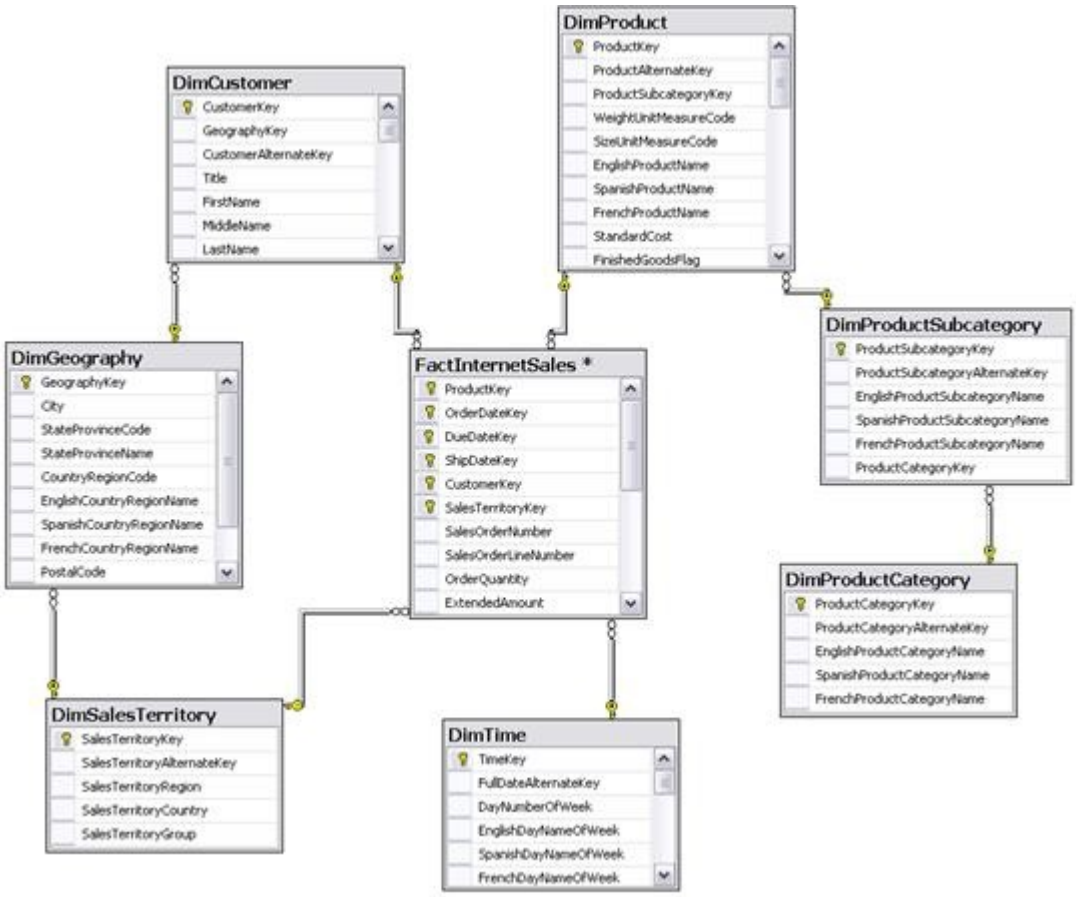


Рис. 2.3. Сховище даних

На цій схемі таблиці вимірювань містять інформацію про покупців (DimCustomer), про товари (DimProduct), про місце продаж (DimSalesTerritory), про час продаж (DimTime); консольні таблиці: під категорія товарів (DimProductSubcatecory), категорія товарів (DimProductCategory), узагальнене місце продажів (DimGeography) і таблиця фактів FactInternetSales містіть ключі для зв’язків

с таблицямі вимірювань (ProductKey, OrderDateKey, DueDateKey, ShipDateKey, CustomerKey, SalesTerritoryKey), а також самі дані для подальшого аналізу (SalesOrderNumber, SalesOrderLineNumber, OrderQuantity, ExtendedAmount).

В`ювері для структури інтелектуального аналізу по алгоритму асоціативних правил

Для полегшення аналізу створюються 2 в`ювера vAssocSeqLineItems і vAssocSeqOrders.

```
CREATE VIEW [dbo].[vAssocSeqLineItems]
AS
  SELECT
    OrderNumber
    ,LineNumber
    ,Model
  FROM
    [dbo].[vDMPrep]
  WHERE
            FiscalYear = '2004'
```

Рис. 2.4. SQL-інструкція на створення vAssocSeqLineItems

```
CREATE VIEW [dbo].[vAssocSeqOrders]
AS
  SELECT DISTINCT
    [OrderNumber]
    ,[CustomerKey]
    ,[Region]
    ,[IncomeGroup]
  FROM
    [dbo].[vDMPrep]
  WHERE
            [FiscalYear] = '2004'
```

Рис.2.5. SQL-інструкція на створення vAssocSeqOrders

Ці вьювери создаються на підставі вьювера vDMPrep, який у свою чергу був створений з таблиць сховища AdventureWorks.

```
CREATE VIEW [dbo].[vDMPrep]
AS
SELECT pc.EnglishProductCategoryName, COALESCE (p.ModelName,
p.EnglishProductName) AS Model, c.CustomerKey, s.SalesTerritoryGroup AS Region,
CASE WHEN Month(GetDate()) < Month(c.[BirthDate])
THEN DateDiff(yy, c.[BirthDate], GetDate()) - 1 WHEN Month(GetDate()) =
Month(c.[BirthDate]) AND Day(GetDate()) < Day(c.[BirthDate]) THEN DateDiff(yy,
c.[BirthDate], GetDate())-1 ELSE DateDiff(yy, c.[BirthDate], GetDate())END AS Age,
CASE WHEN c.[YearlyIncome] < 40000 THEN 'Low' WHEN c.[YearlyIncome] >
60000
THEN 'High' ELSE 'Moderate' END AS IncomeGroup,
      t.CalendarYear, t.FiscalYear, t.MonthNumberOfYear AS Month,
      f.SalesOrderNumber AS OrderNumber, f.SalesOrderLineNumber AS
LineNumber,
    f.OrderQuantity AS Quantity, f.ExtendedAmount AS Amount
FROM dbo.FactInternetSales AS f INNER JOIN
    dbo.DimTime AS t ON f.OrderDateKey = t.TimeKey INNER JOIN
    dbo.DimProduct AS p ON f.ProductKey = p.ProductKey INNER JOIN
    dbo.DimProductSubcategory AS psc ON p.ProductSubcategoryKey =
psc.ProductSubcategoryKey INNER JOIN
    dbo.DimProductCategory AS pc ON psc.ProductCategoryKey =
pc.ProductCategoryKey INNER JOIN
    dbo.DimCustomer AS c ON f.CustomerKey = c.CustomerKey INNER JOIN
    dbo.DimGeography AS g ON c.GeographyKey = g.GeographyKey INNER JOIN
     dbo.DimSalesTerritory AS s ON g.SalesTerritoryKey = s.SalesTerritoryKey
```

Рис. 2.6. SQL-інструкція на створення vDMPrep

3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ АНАЛІТИЧНОЇ ОБРОБКИ ДИНИХ

3.1 Створення джерела даних

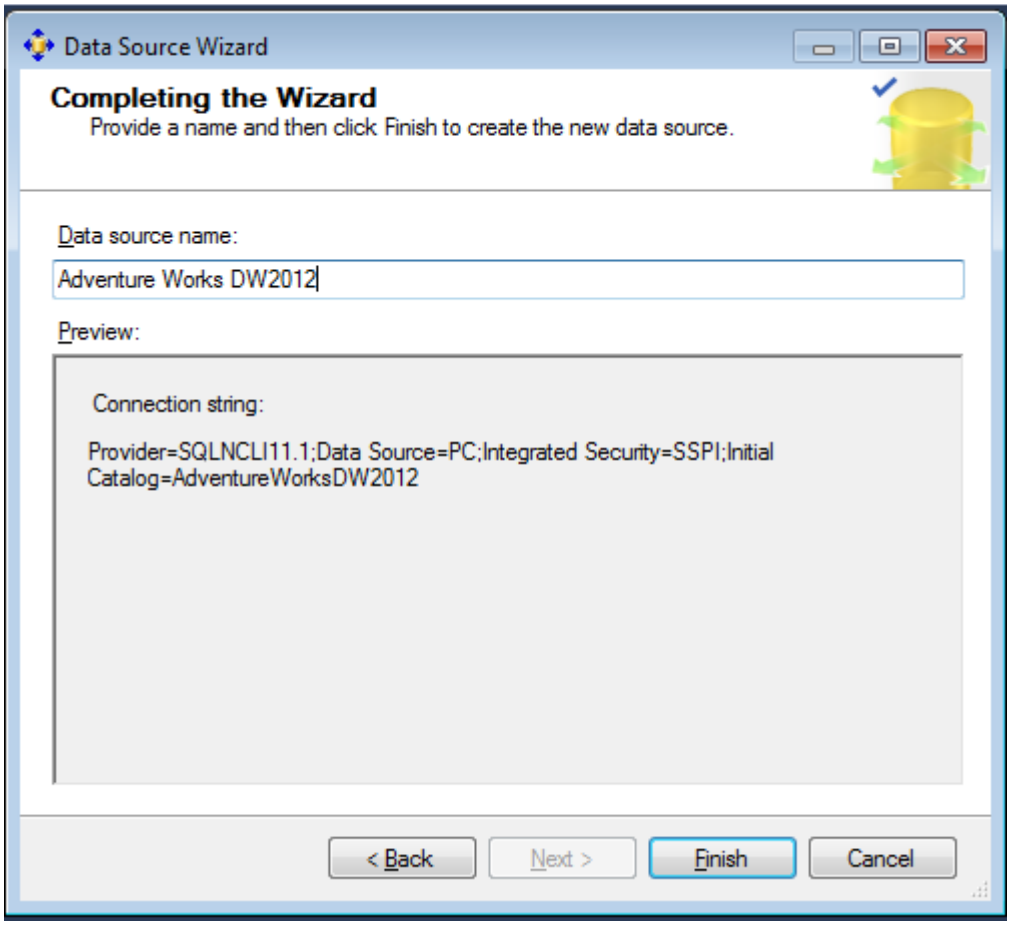


Рис. 3.1. Створення джерела даних

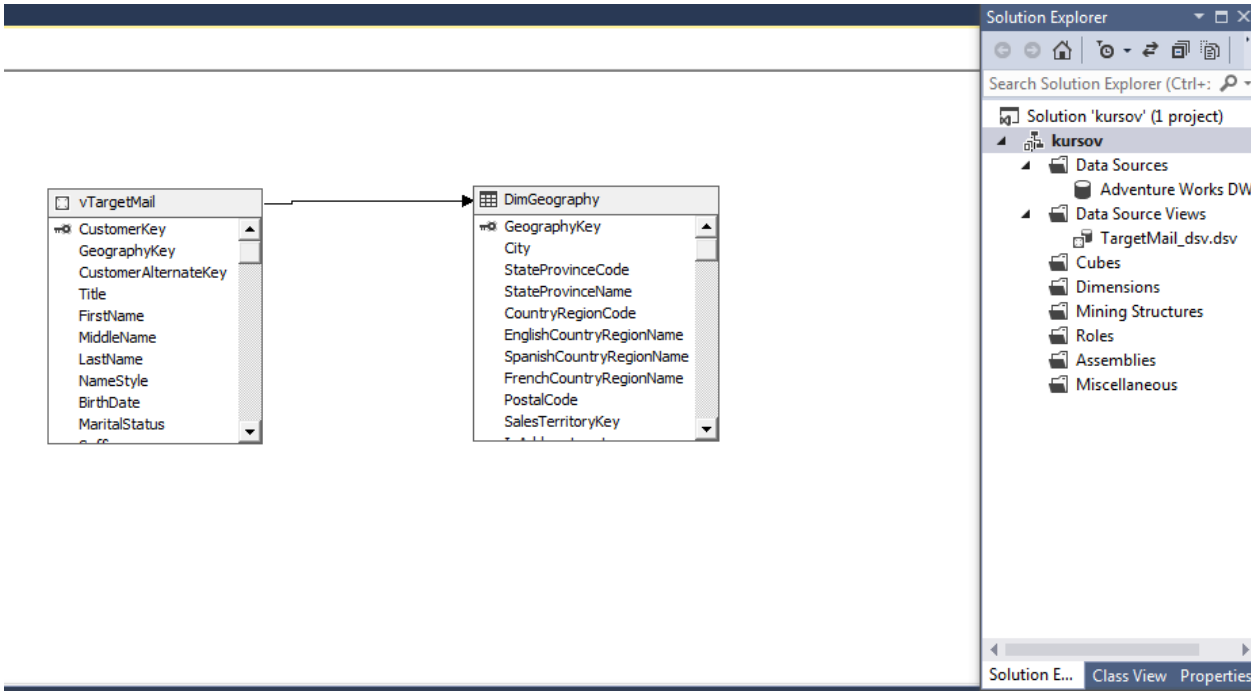


Рис. 3.1.2 Вказані зв'язки між таблицями

3.2 Створення представлення джерела даних

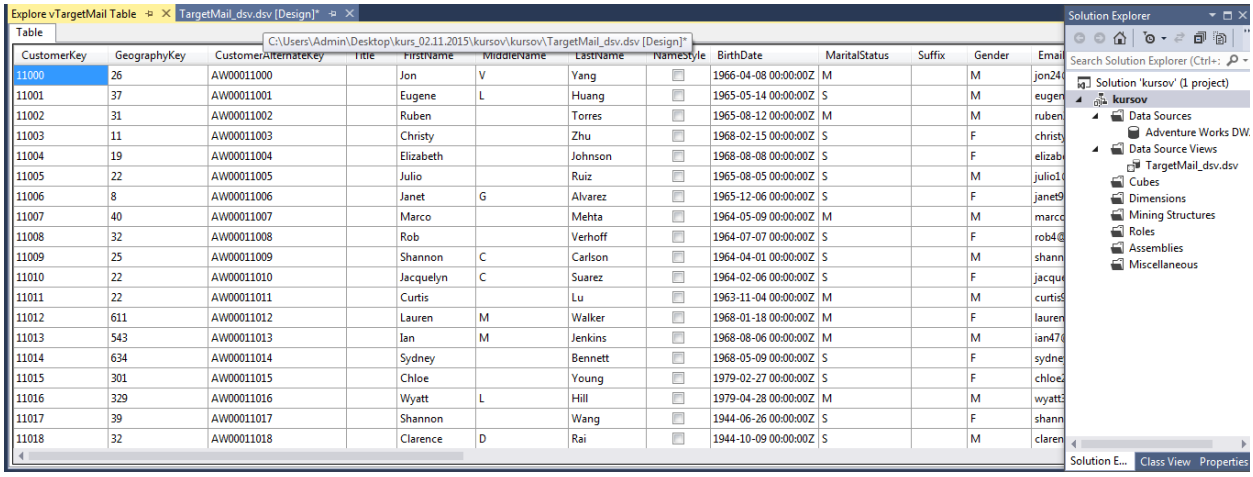


Рис. 3.2.1.Ознайомлення с даними

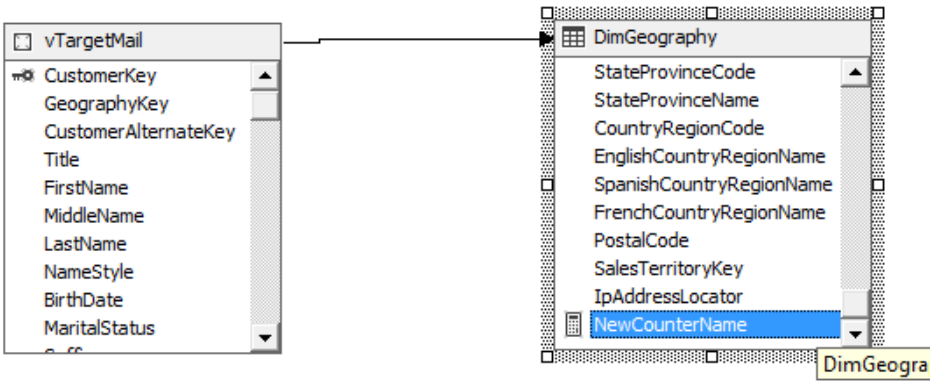


Рис. 3.3. Створення іменованого обчислення

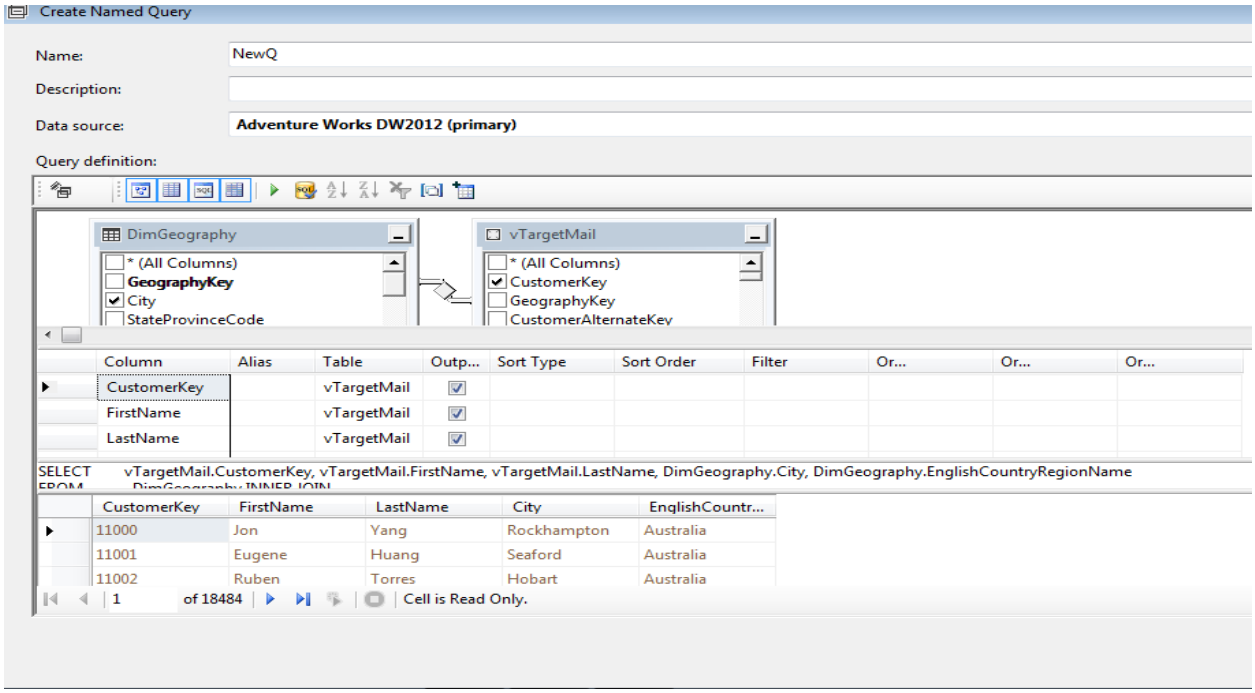


Рис. 3.4. Створення іменованого запиту

3.3 ЗАВДАННЯ КЛАСТЕРИЗАЦІЇ

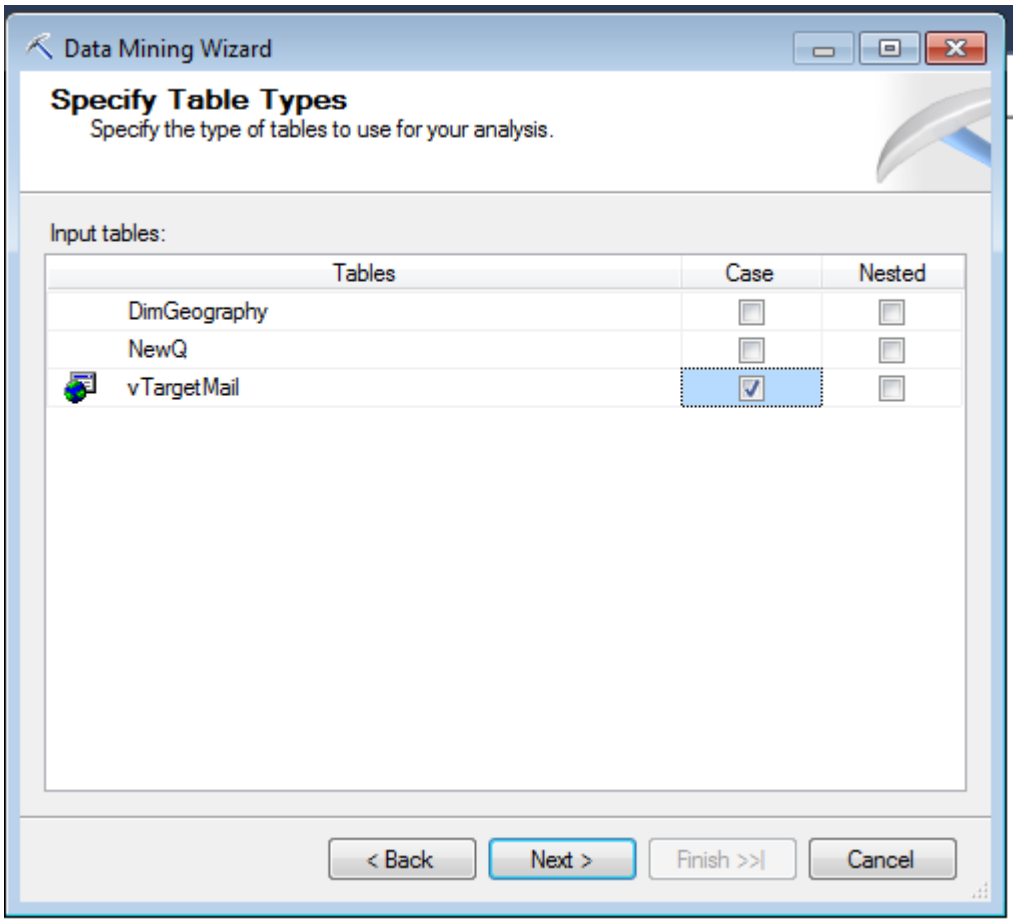


Рис. 3.5. Створення нової структури і моделі інтелектуального аналізу. Вибір таблиці варіантів.

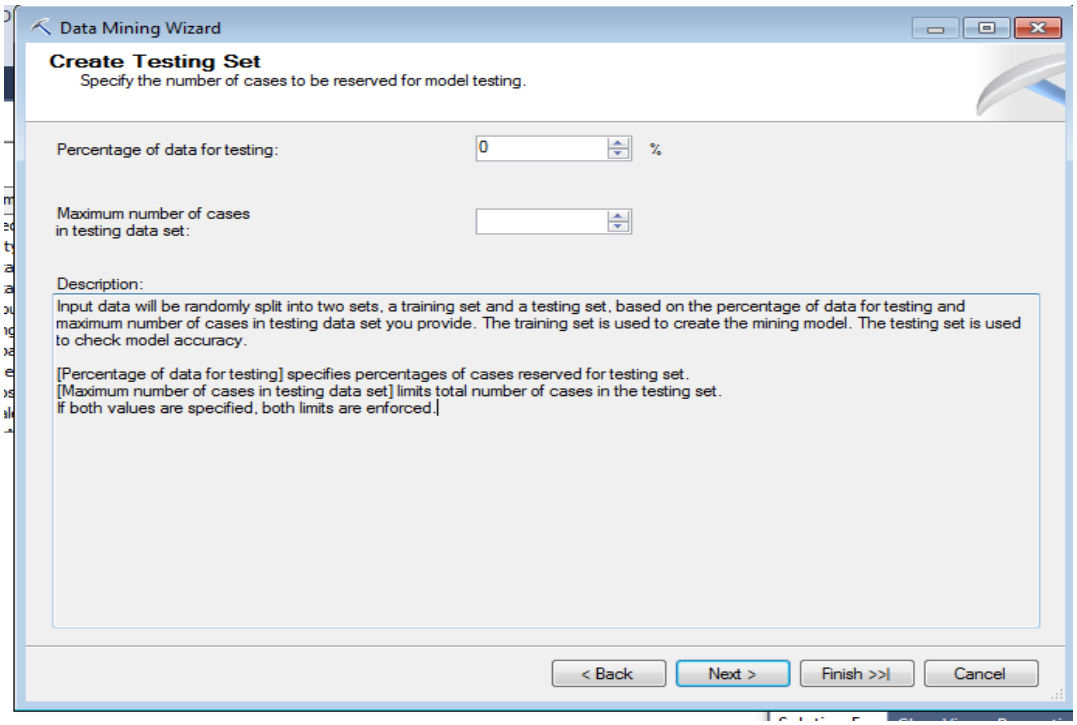


Рис. 3.6. Резервування даних для цілі тестування.

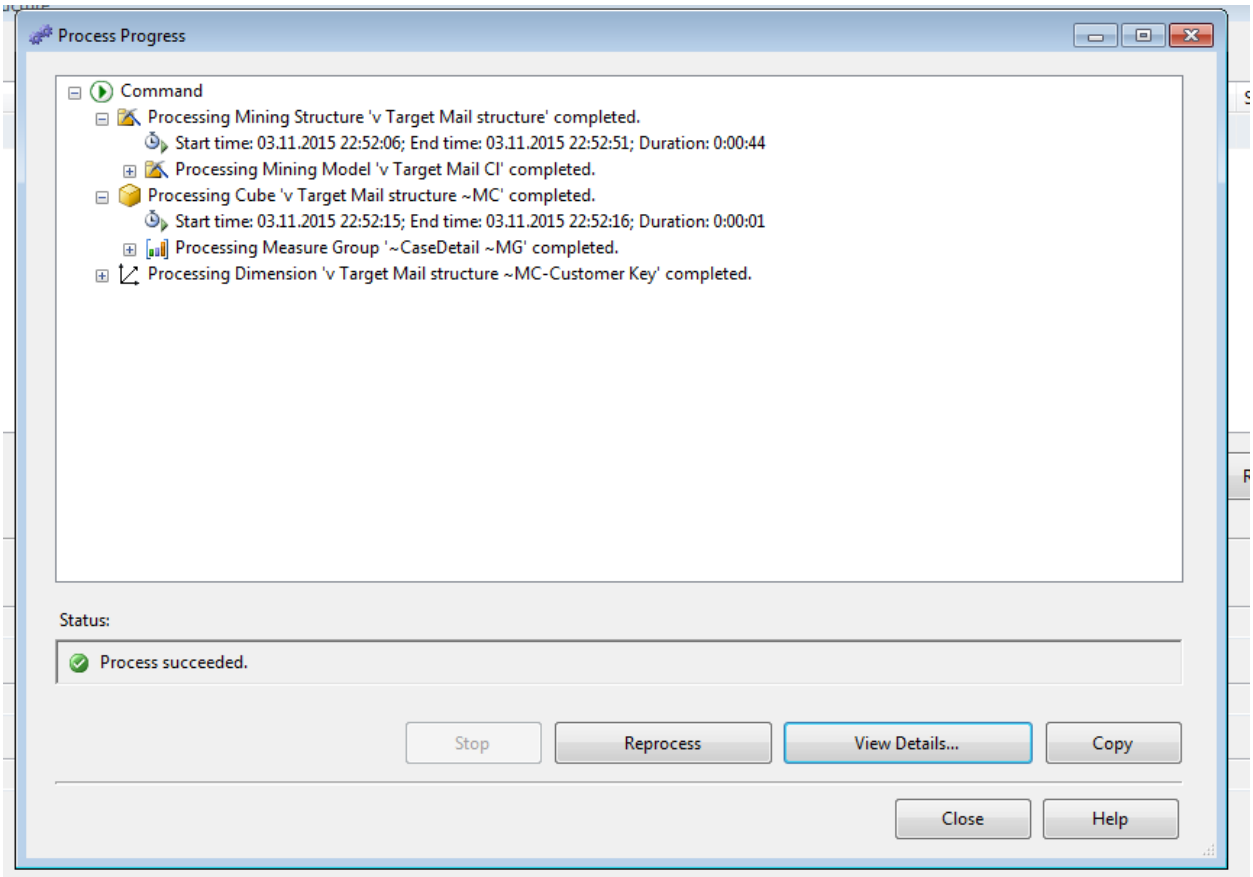


Рис. 3.7. Повна обробка для створення структури

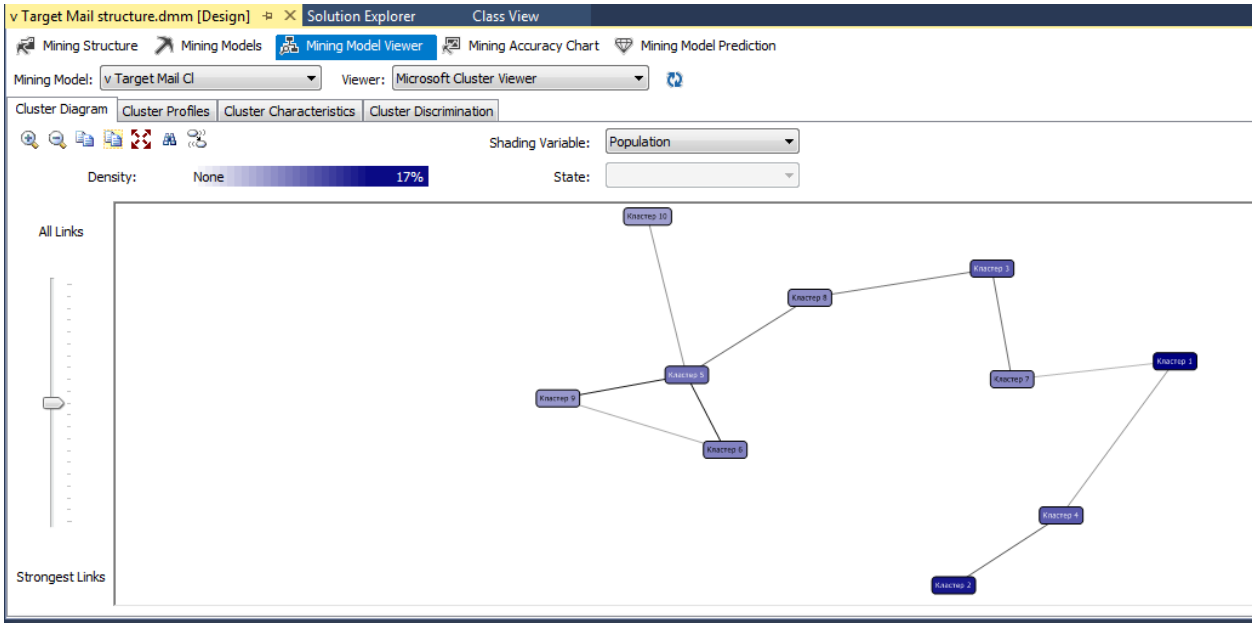


Рис. 3.8. Діаграма кластерів

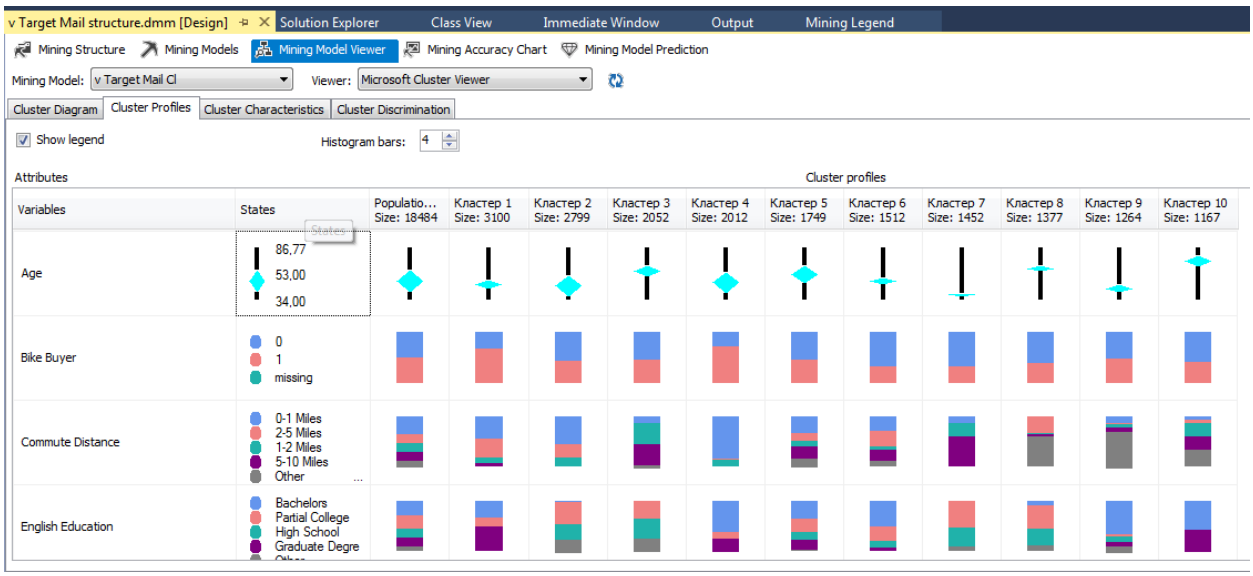
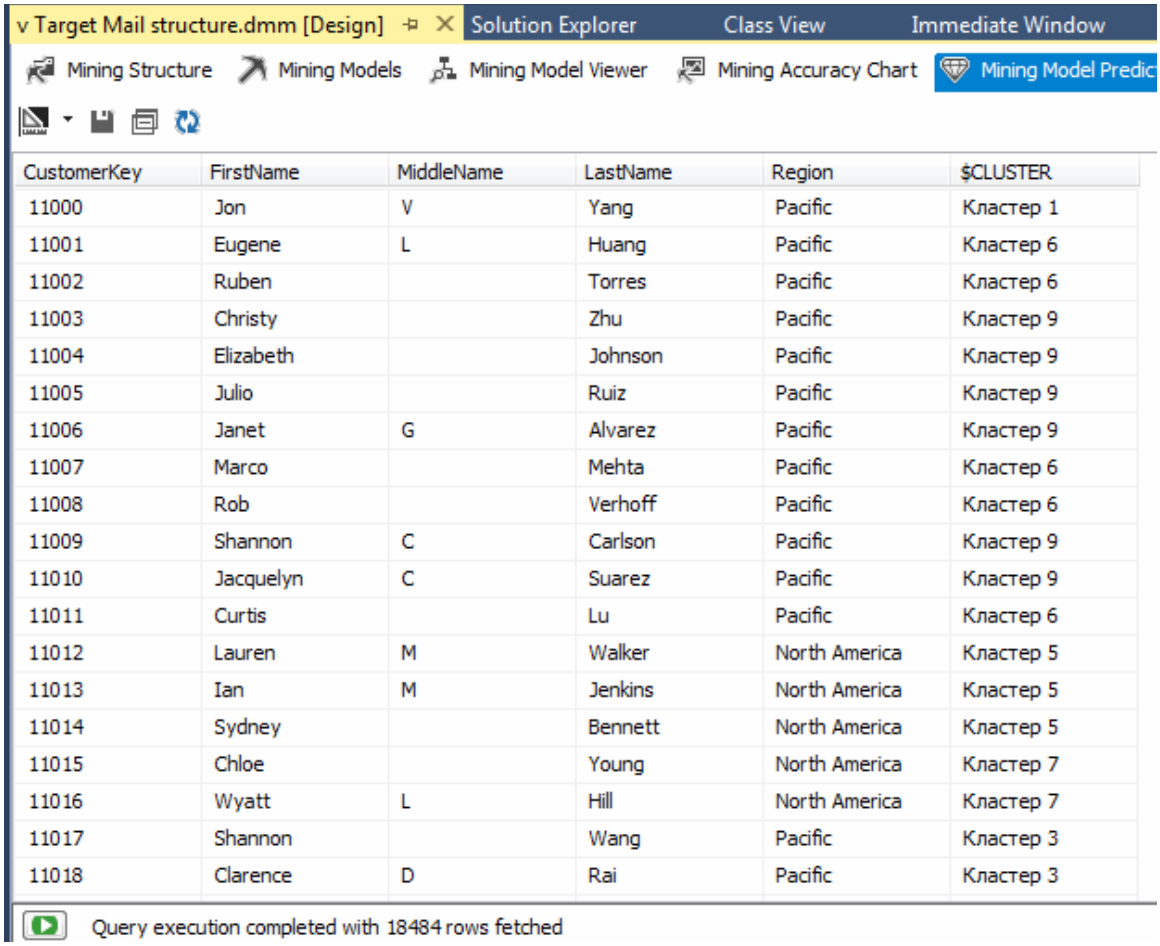


Рис. 3.9. Характеристики виявлених кластерів



CustomerKey	FirstName	MiddleName	LastName	Region	$CLUSTER
11000	Jon	V	Yang	Pacific	Кластер 1
11001	Eugene	L	Huang	Pacific	Кластер 6
11002	Ruben		Torres	Pacific	Кластер 6
11003	Christy		Zhu	Pacific	Кластер 9
11004	Elizabeth		Johnson	Pacific	Кластер 9
11005	Julio		Ruiz	Pacific	Кластер 9
11006	Janet	G	Alvarez	Pacific	Кластер 9
11007	Marco		Mehta	Pacific	Кластер 6
11008	Rob		Verhoff	Pacific	Кластер 6
11009	Shannon	C	Carlson	Pacific	Кластер 9
11010	Jacquelyn	C	Suarez	Pacific	Кластер 9
11011	Curtis		Lu	Pacific	Кластер 6
11012	Lauren	M	Walker	North America	Кластер 5
11013	Ian	M	Jenkins	North America	Кластер 5
11014	Sydney		Bennett	North America	Кластер 5
11015	Chloe		Young	North America	Кластер 7
11016	Wyatt	L	Hill	North America	Кластер 7
11017	Shannon		Wang	Pacific	Кластер 3
11018	Clarence	D	Rai	Pacific	Кластер 3

Рис. 3.10. Результат виконаного запиту

ВИСНОВКИ

Data Mining включає величезний набір різних аналітичних процедур, що робить його недоступним для звичайних користувачів, які слабо розбираються в методах аналізу даних. Компанія StatSoft знайшла вихід і з цієї ситуації, даний пакет Statistica можуть використовувати як професіонали, так і звичайні користувачі, що володіють невеликими досвідом і знаннями в аналізі даних і математичній статистиці. Для цього крім загальних методів аналізу були вбудовані готові закінчені (сконструйовані) модулі аналізу даних, призначені для вирішення найбільш важливих і популярних завдань: прогнозування, класифікації, створення правил асоціації і т.д.

ЛІТЕРАТУРА

1. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки данных. Вводный курс. - М.: Гелиос АРВ, 2002. - 368 с.

2. Гайна Г.А. Організація баз даних і знань. Мови баз даних: Конспект лекцій.- К .:КНУБА, 2002. - 64 с.

3. Гайна Г.А., Попович Н.Л. Організація баз даних і знань. Організація реляційних баз даних: Конспект лекцій. - К.:КНУБА, 2000. - 76 с.

4. Гарсиа-Молина Г., Ульман Д., Уидом Д. Системы баз данных.-М.: Издательский дом "Вильямс", 2003. - 1088 с.

5. Григорьев Ю.А., Ревунков Г.И. Банки данных.-М.: Изд-во МГТУ им. Н.Э.Баумана, 2002. - 320 с.

6. Грофф Дж., Вайнберг П. Энциклопедия SQL. - СПб.: Питер, 2003. - 896 с.

7. Дейт К.Дж. Введение в системы баз данных. - К.: Диалектика, 1998. - 784 с.

8. Диго С.М. Проектирование и использование баз данных.-М.: Финансы и статистика, 1995. - 208 с.

9. Карпова Т.С. Базы данных: модели, разработка, реализация. - СПб.: Питер, 2001. - 304 с.

10. Когаловский М.Р. Энциклопедия технологий баз данных.- М.: Финансы и статистика, 2002. - 800 с.

11. Конноли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. - М.: Издательский дом "Вильямс", 2003. - 1440 с.

12. Кренке Д. Теория и практика построения баз данных. - СПб.: Питер, 2003. - 800 с.

13. Малыхина М.П. Базы данных: основы, проектирование, использование. - СПб.: БХВ-Петербург, 2004. - 512 с.

14. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. - СПб.: БХВ-Петербург, 2004. - 1040 с.