

Міністерство освіти та науки
Національний університет "Львівська політехніка"
кафедра "Інформаційні системи та мережі"

Моделювання процесів і технологій сховищ даних
Методичні вказівки

до виконання лабораторних робіт
з курсу "Технології сховищ даних"

для студентів, що навчаються за спеціальністю
8.080405 "Інтелектуальні системи прийняття рішень"

Уклала
Шаховська Н.Б.,
доц. каф. ІСМ, к.т.н.

Затверджені на засіданні
кафедри ІСМ
протокол № 1
від „ „ 200_ р.

Львів 2008

Лабораторна робота № 4. Моделювання репозиторію метаданих у сховищах даних

Мета роботи: Вивчення порядку, методів та засобів створення структури та складу метаданих для сховищ даних за схемою Захмана, розроблення засобів навігації метаданих.

Теоретичні відомості

Метадані (від грецьк. Meta і лат. Data), буквально перекладається як «дані про дані», інформація про інший набір даних.

Одне з корисних визначень наступне: «Метадані - це структуровані, кодовані дані, які описують характеристики об'єктів-носіїв інформації, що сприяють ідентифікації, виявленню, оцінюванню, керуванню цими об'єктами».

Майкл Бреккет (Michael Brackett) визначає метадані (які він називає «даними про ресурси даних») як «будь-які дані про інформаційні ресурси організації». Адрієн Танненбаум (Adrienne Tannenbaum) називає метадані «детальним описом сутності даних». Ці визначення розкривають формулювання «дані про дані».

Тема ця підіймається відтоді, як існують дані: метадані були необхідні для опису значення і властивостей інформації з метою кращого її розуміння, керування і використання. Класичним прикладом є бібліотеки. Книги (дані) можна класифікувати, керувати ними і знаходити тільки за допомогою відповідних метаданих (тобто заголовка, автора і ключових слів змісту).

Зазвичай під метаданими розуміється будь-яка інформація, необхідна для аналізу, проектування, побудови, впровадження і застосування в комп'ютерній системі. У разі інформаційних систем метадані особливо спрощують керування, створення запитів, повноцінне використання і розуміння даних. Багато недавніх проектів, як наукові, так і практичні, націлені на вивчення метаданих. Генерування, зберігання і керування метаданими допомагають в підтримці використання величезних обсягів інформації, доступних в наші дні в будь-якій електронній формі. Оскільки все, з чим працює комп'ютер, за суттю є даними, і свого роду метадані супроводжують будь-які дані, то це поняття дотичне до будь-якої сфери застосувань і набуває різних форм залежно від застосування.

Метадані систем сховищ даних іноді розділяють на два типи:

- 1) службові метадані, що використовуються для функцій витягання, перетворення і завантаження, для перенесення інформації з транзакційних систем у сховищі;
- 2) інтерфейсні метадані, що використовуються для опису екранів і створення звітів.

Є наступні типи метаданих у сховищі:

- *метадані* початкової системи:
 - специфікації джерел даних, таких як репозиторії;
 - описова інформація (наприклад, частота оновлення, юридичні обмеження і методи доступу);
 - інформація про процеси, такі як графік завдань і коди витягання;
- *метадані* перетворення даних:
 - інформація про отримання даних (наприклад, планування передавання даних і результатів, а також відомості про використання файлів);
 - керування таблицями вимірів, наприклад, визначення вимірів і присвоєння сурогатних ключів;
 - перетворення і агрегація, наприклад, розширення і відображення даних, програми (скрипти) завантаження СКБД, визначення агрегатів даних;
 - документування перевірок, робіт і журналів, наприклад, журналів перетворення даних і записів стеження за походженням даних;
- *метадані СКБД*, такі як:
 - зміст системних таблиць СКБД;
 - рекомендації з опрацювання.

У загальному випадку, для користувача сховища даних потрібні метадані, принаймні, наступних типів.

- Описи структур даних, їх взаємозв'язків.
- Інформація про дані, що зберігаються у сховищі, і підтримувані ним агрегати даних.

- Інформація про джерела даних і про міру їх достовірності. Одна і та ж інформація могла потрапити у сховище даних з різних джерел. Користувач повинен мати можливість знати, яке джерело було вибрано основним, і яким чином здійснюється узгодження і очищення даних.

- Інформація про періодичність оновлень даних. Бажано знати не тільки те, якому моменту часу відповідають необхідні для користувача дані, але і коли вони наступного разу будуть оновлені.

- Інформація про власників даних. Користувачу системи підтримки прийняття рішень може виявитися корисною інформація про наявність в системі даних, до яких він не має доступу, про власників цих даних і про дії, які він повинен зробити, щоб дістати доступ до даних.

- Статистичні оцінки часу виконання запитів. До виконання запиту корисно мати хоч би приблизну оцінку часу, який буде потрібний для отримання відповіді, і обсягу цієї відповіді.

Роль метаданих у сховищі

Найкраще можна пояснити сутність метаданих, описуючи їх роль і призначення в реалізації процесів сховища даних. Метадані можна використовувати трьома способами:

- *пасивно*, забезпечуючи чітку документацію про структуру, процес розроблення і використання системи СД; доступна документація необхідна всім учасникам (тобто кінцевим користувачам, системним адміністраторам, а також розробникам застосувань);

- *активно*, шляхом зберігання конкретних семантичних аспектів (наприклад, правил перетворення) у вигляді метаданих, які можна інтерпретувати і використовувати під час виконання; у цьому випадку процеси сховища даних керуються метаданими. А отже, код (тобто активні метадані) і додаткова документація погоджено і уніфіковано керуються в одному репозиторії, при цьому актуальність документації зростає;

- *напівактивно*, за рахунок зберігання статичної інформації (наприклад, визначень структур, специфікацій конфігурацій), яку прочитуватиме інший програмний компонент під час виконання; наприклад, опрацьовувачам запитів необхідні метадані для перевірки існування атрибутів; на відміну від активного використання, тут метадані тільки читаються, але не виконуються.

Створення і керування метаданими служить двом цілям:

- 1) мінімізації робіт з розроблення і адміністрування СД;
- 2) ефективнішому витяганню інформації з СД.

Перша мета, в основному, стосується:

- *підтримки інтеграції систем:*

- схеми й інтеграція даних залежать від метаданих, що описують структуру і сенс окремих джерел даних і цільових систем;

- правила перетворення можна застосувати до початкових даних і зберігати як мета дані;

- більше того, інтеграція різних інструментів можлива тільки тоді, коли вони розділяють «дані», які в такому випадку є метаданими системи сховища даних;

- *підтримки аналізу і проектування нових застосувань:*

- метадані підвищують контрольованість і надійність процесу розроблення застосувань, забезпечуючи інформацію про сенс даних, їх структуру і джерела;

- метадані, що стосуються рішень з проектування застосувань, можна використовувати повторно;

- *підвищення гнучкості системи і можливості повторного використання наявних програмних модулів:*

- це можливо тільки для активного і напівактивного використання метаданих; семантичні аспекти, що швидко змінюються, явним чином зберігаються у вигляді метаданих поза прикладними програмами;

- підтримка істотно простіша;

- систему можна розширити і адаптувати без жодних труднощів;

- цей підхід також дає можливість повторного використання «фрагментів коду»;

- *автоматизації адміністративних процесів.*

- метадані керуються запуском різних процесів СД (наприклад, завантаження і оновлення);
- інформація про їх виконання (журнали доступу, кількість доданих у сховищі записів і т.ін.) також міститься в репозиторії, легко доступному адміністратору;
- *посилення механізмів безпеки:*
 - метадані повинні забезпечити правила доступу і призначені для користувача права для всієї системи СД;
 - керування доступом в сховищі даних іноді вимагає застосування складних методів; наприклад, оперативне джерело може містити нешкідливу інформацію про окремі показники роботи компанії, проте сумарні значення у сховищі іноді виявляються найважливішим секретом; з другого боку, персональні доходи кожного співробітника є таємницею, але при цьому підсумкова сума зарплат в СД може зовсім не бути критичною інформацією.

Друга мета відноситься до ефективного витягання інформації, а точніше до:

- *підвищення якості даних.* Якість даних визначається наступними характеристиками:
 - 1) узгодженістю (чи є подання даних однорідним, чи немає дублікатів, даних з пересіченими або конфліктними визначеннями);
 - 2) повнотою (чи всі дані присутні);
 - 3) точністю (збігом значень, що зберігаються, і фактичних);
 - 4) своєчасністю (чи актуальне значення, що зберігається у сховищі);
- *правила перевірки якості даних;*
- *поліпшення взаємодії усередині системи сховища даних;*
- *поліпшення аналізу даних;*
- *застосування загальної термінології і мови взаємодії усередині корпорації.*

Правила перевірки якості даних необхідно задати, зберегти у вигляді метаданих і перевіряти при кожному оновленні сховища даних. Крім того, висока якість вимагає підтримки контролю даних. Метадані забезпечують інформацію про час створення і про автора даних, про джерело, значення даних у момент отримання (про спадковість даних), і про подальший шлях від джерела до поточного місцеперебування (data lineage — про походження даних). Отже, користувачі можуть відновити ланцюжок, яким рухаються дані за час перетворення, і перевірити точність поверненої інформації;

Поліпшення взаємодії усередині системи сховища даних. Взаємодія відбувається як за допомогою виконання простих запитів і звітних застосувань, так і з використанням складних аналітичних інструментів. Метадані забезпечують відомості про значення даних, термінологію і бізнес-концепції підприємства, а також їх зв'язок з даними. Тому метадані підвищують якість запитів за рахунок точнішого і строгого формулювання, а також скорочують витрати на користувачів, яким необхідний доступ, оцінка і застосування відповідної інформації;

Поліпшення аналізу даних. Методи аналізу даних подані широко — починаючи від простих застосувань звітності та систем підтримки прийняття рішень і закінчуючи складними застосуваннями видобування даних. У цьому напрямі метадані необхідні для розуміння предметної області і її подання у сховищі з тим, щоб адекватно застосувати і інтерпретувати результати;

Застосування загальної термінології і мови взаємодії усередині корпорації. Доступність метаданих як унікального джерела документації для користувачів має і інші переваги. Вона гарантує узгоджені засоби взаємодії й інтерпретації інформації зі сховища, а також усуває двозначність і забезпечує узгодженість відомостей усередині компанії, дозволяє розділяти знання і досвід.

Метадані системи сховища даних містяться в репозиторії — структурованій системі зберігання і витягання, реалізованій на основі СКБД. Для інтерпретації метаданих необхідно зберігати структуру репозиторію (тобто схему метаданих) і їх семантику.

Для інформації про дані сховища доцільно застосовувати шестивимірну класифікаційну схему Захмана (Zachman) (відповідно до відповідей на запитання що? хто? де? коли? чому? як?):

- об'єкти (що?);
- суб'єкти (хто?);
- місцезнаходження (де?);

- час (коли?);
- фактори впливу, чинники (чому?);
- способи (як?).

При цьому використовується наступна формалізація:

- сутність або вміст сховища даних;
- люди, які використовують сховище даних;
- місцезорешування даних, важливе з погляду керування сховищем даних;
- моменти завантаження даних і обчислення підсумкових таблиць;
- рушійні сили створення і розвитку сховища даних;
- дії, які виконуються з даними;
- повчальні метадані (як новий чинник, що використовуватиметься для підтримки розвитку моделі сховища даних).

Метадані зберігаються в окремій базі даних метаданих або репозиторії. Системне програмне забезпечення для створення репозиторію пропонується рядом компаній, в першу чергу, розробниками СКБД.

	Мотивація	Люди	Дані	Функції	Місце	Час
Контекст	Мета і стратегія побудови СД	Люди, що використовують СД	Об'єкти, які моделює СД	Основні бізнес-процеси	Географія СД	Періоди завантаження даних
Модель СД	Дії, які виконуються над даними	Моделі потоків робіт	Семантичні моделі	Моделі бізнес-процесів	Система логістики	Базовий графік робіт
Системна модель	Моделі бізнес-правил	Архітектура інтерфейсу користувача	Концептуальна модель даних	Архітектура застосувань	Архітектура розподіленої системи	Структура опрацювання подій
Технологічна модель	Моделі правил опрацювання подій	Архітектура подання	Фізична модель даних	Архітектура програмно-апаратної системи	Технологічна архітектура	Структура циклів керування
Детальне подання	Специфікації правил роботи системи	Специфікації ролей та прав доступу	Специфікації форматів даних	Код програмних компонент	Специфікації архітектури мережі	Специфікації опрацювання подій і переривань
Робоче СД	Стратегія і практика	Структура СД	Дані	Функції, що виконуються	Географічне положення і мережі	Плани

Рис. 1. Модель Захмана в контексті побудови метаданих сховища даних.

Важливість цього ключового аспекту сховища даних можна продемонструвати у різних аспектах:

- єдині правила найменування об'єктів,
- єдині одиниці вимірювання для однотипних об'єктів,
- єдине фізичне подання однотипних об'єктів,
- єдині атрибути подання однотипних об'єктів, тощо.

Порядок виконання роботи

1. Розроблення структури метаданих сховища даних

- *Зміст завдання:* Розробити шестивимірну модель метаданих за схемою Захмана.

Для кожної з категорій визначити множину значень метаданих та їх властивостей. Розробити структуру таблиць для їх збереження у базі даних. Виконання завдання виконується за таким прикладом

-) Визначення об'єктів (сутностей та їх властивостей) для моделі сховища даних, розробленого у попередніх роботах:

Таблиця 1.

Приклад метаданих про об'єкти

@table					
Код	Назва	Тип об'єкта	Сховище	Примітка	Користувач
28h/@form	@form	Форма	2		Unknown
28h/@report	@report	Звіт	2		Unknown
28h/@reportparam	@reportparam	Параметр звіту	2		Unknown
28h/@reportrun	@reportrun	Запуск звіту	2		Unknown
28h/@store	@store	Сховище	2		Unknown
28h/@table	@table	Таблиця	2		Unknown
28h/@tablecolumn	@tablecolumn	Колонка таблиці	2		Unknown
28h/dc_journal	dc_journal	Документ	2		Unknown
28h/dt_journal	dt_journal	Деталь документа	2		Unknown
28h/ob_admunit	ob_admunit	Адміністративно-територіальна одиниця	2		Unknown
28h/ob_client	ob_client	Клієнт	2		Unknown
28h/ob_dept	ob_dept	Підрозділ	2		Unknown

(код сховища – див таблиця 3)

- а) Визначення суб'єктів (персон, посадових осіб та їх властивостей) для моделі сховища даних, розробленого у попередніх роботах:

Таблиця 2.

Приклад метаданих про осіб, пов'язаних зі сховищем даних

А)

run					
Звіт	Користувач	Тип	Виконати	Примітка	Дата запису
oba_product_settl	Петренко А.І.	Сист. Адмін.	oba_product_settl_perc		15:17:12 22.10.2007
ohg_date	Василенко В.І.	Аналітик продажу	CreateCalendar({p_begdate},{p_enddate},{p_montype_id},{p_tuetype_id},{p_wedtype_id},{p_thutype_id},{p_fritype_id},{p_satttype_id},{p_sunttype_id},{p_period_id})		17:16:03 22.10.2007

Б)

Table_user	
Користувач	Дозволена таблиця
oba_product_settl	dc_order
oba_product_settl	dt_order
oba_product_settl	ob_partner
oba_product_settl	ob_person
oba_product_settl	ob_product
ohg_date	dc_order
ohg_date	dt_order
ohg_date	ob_product

- б) Визначення місцезнаходження важливих елементів сховища даних для моделі, розробленої у попередніх роботах:

Таблиця 3.

Приклад метаданих про місцезнаходження елементів сховища даних

metadata					
id	Тип джерела	Шлях до файлу	Ім'я файлу	Назва таблиці	Тип поповнення
1	Access	d:\	source1.mdb	documents	джерело даних
2	Access	d:\	База даних.mdb	documents	СД
3	Excel	d:\	avans.xls	documents	джерело даних

- с) Визначення метаданих, що описують місцезнаходження важливих елементів сховища даних для моделі, розробленої у попередніх роботах:

Таблиця 4.

Приклад метаданих про час та регламент

autoexe					
Код	Параметр	Опис	Запуск_процедури	Об'кт запуску	Періодичність
1	01.____	додавання даних з джерела	add_data	1	місяць
2	01.____	додавання даних з джерела	add_data	3	місяць
3	01.____	архівування даних	arch_data	2	місяць

У цій таблиці об'єктом запуску є код мета даних з таблиці 3.

- д) Визначення чинників, факторів впливу, які визначають розвиток та процеси функціонування сховища даних для моделі, розробленої у попередніх роботах:

Таблиця 5.

Приклад метаданих про чинники сховища даних

№	Назва фактору	Відповідальний	На що впливає	Примітки
0.	Угода на виконання досліджень для створення сховища даних	Моя фірма	проект сховища даних	Виконано у 2007 році.
1.	Технічний проект сховища даних	Моя фірма	реалізація сховища даних	Виконано у 2007 році.
2.	Регламент формування аналітичних звітів	керівник аналітичного департаменту	вітрини даних	вітрини оновлюються до виконання звітів
3.	Графік оновлення сховища даних	адміністратор сховища	бази даних, вітрини відділів, репозиторій метаданих	

- е) Визначення способів та засобів виконання певних дій для сховища даних для моделі, розробленої у попередніх роботах:

Таблиця 6.

Приклад метаданих способи виконання операцій

№	Операція	Способи виконання	Програма-виконувач	Шлях
0.	Завантаження оперативних даних	ETL	Oracle ETL Manager	D:\DataWarehouse\Progs\Oracle\EtlManager.exe
1.	Формування вітрин даних	Агрегування, аналіз, збереження	Aggregation Express, Oracle Express Server, Administratitot tools	
2.	Формування регламентованих звітів	Виклик прикладної програми	Prepared Report Manager	D:\DataWarehouse\Progs\Application\ReptManager.exe
3.	Архівування даних	Виклик прикладної програми	Сховище даних	Module1->archive_dw()

2. Розроблення структури таблиць для репозиторію метаданих.

Зміст завдання: Розробити структуру таблиць для зберігання кожного з шести вимірів мета даних.

Приклад:

metadata : таблиця		
	Имя поля	Тип данных
	id	Счетчик
	source_type	Текстовый
	filepath	Текстовый
	source_name	Текстовый
	tablename	Текстовый
	type	Текстовый

3. Заповнення таблиць значеннями метаданих

Зміст завдання: Розробити приклади значень метаданих кожного з шести вимірів та внести їх у розроблені таблиці.

Приклад:

metadata : таблиця						
	id	Тип джерела	Шлях до файлу	Имя файла	Назва таблиці	Тип поповнення
	1	Access	d:\	source1.mdb	fact1	джерело даних
	2	Access	d:\	db1.mdb	fact	СД
	3	Excel	d:\	prodaz.xls	fact2	джерело даних
*	(Счетчик)					

4. Розроблення засобів навігації для репозиторію метаданих

Зміст завдання: Розробити набір пов'язаних екранних форм для доступу до всіх категорій метаданих.

5. Розроблення засобів запуску методів з репозиторію метаданих

Зміст завдання: Розробити процедури для запуску функцій з таблиці 6. Якщо це невиконуваний файл, то його можна запускати з середовища Access таким чином (через формування bat-файлу):


```
Public Sub create_bat_file(path As String, dat As Date)
Dim fs, a
Dim str As String
' дата виконання співпадає з сьогоднішньою
If dat = Date Then

str = CurDir
Set fs = CreateObject("Scripting.FileSystemObject")
Set a = fs.CreateTextFile(str & "\db.bat", True)
a.writeline("@echo off")
a.writeline("@path=c:\windows;")
a.writeline("@explorer " & Chr(34) & path & Chr(34))
a.Close
str = CurDir & "\db.bat"
Dim b
b = Shell(str, vbMaximizedFocus)
End If

End Sub
```

У цьому прикладі ми відкриваємо Провідник для певного каталогу, шлях якого вказаний в path. Якщо файл виконуваний, то запуск відбувається функцією Shell. Також передбачити процедуру формування дати запуску (в прикладі не вказана).

Зміст звіту по роботі:

1. Тема лабораторної роботи.
2. Завдання та постановка задачі лабораторної роботи.
3. Теоретична підготовка.
4. Опис виконаної роботи та отриманих результатів по кожному з пунктів завдання:
 - перелік, опис та обґрунтування метаданих всіх категорій за схемою Захмана;
 - структура таблиць для зберігання метаданих всіх вимірів (категорій);
 - приклади заповнення таблиць тестовими значеннями метаданих всіх категорій;
 - опис екранної форми управління навігацією в репозиторії метаданих;
 - опис екранних форм для відображення метаданих всіх вимірів (категорій);
 - процедура для запуску методів, вказаних у репозиторії метаданих, на виконання.
5. Висновки.

Лабораторна робота № 5. Моделювання процесів підготовки, інтеграції та завантаження даних

Мета роботи: Вивчення порядку, методів та засобів добування, узгодження, інтеграції даних, створення оперативних сховищ даних та завантаження даних у центральне сховище.

Теоретичні відомості

Інтеграція даних – це об'єднання даних, які спочатку вводяться в різні системи. Самі ці системи можуть розташовуватися в одній локальній мережі, але мати різні платформи і внутрішню архітектуру. Така ситуація практично неминуча у всіх підприємствах, що займаються складним бізнесом. Як правило, один єдиний постачальник не може створити систему, в якій однаково добре вирішені питання бухгалтерського обліку і автоматизації виробничого циклу, керування кадрами і документообігу і так далі. Приклад інтеграції даних з різних джерел подано на рис. 1.

Крім того, існують завдання, наприклад, маркетингового аналізу, залучення клієнтів, аналізу конкурентного середовища, які за своєю природою вимагають отримання (купівлі) інформації від різних постачальників. Ця інформація постачається у вигляді різноманітних баз даних або електронних таблиць і вимагає завантаження в загальну базу даних для сумісного аналізу.

Метою інтеграції даних є отримання єдиної і цілісної картини корпоративних даних предметної області. Інтеграція даних може бути описана за допомогою моделі, яка включає застосування, продукти, технології та методи:

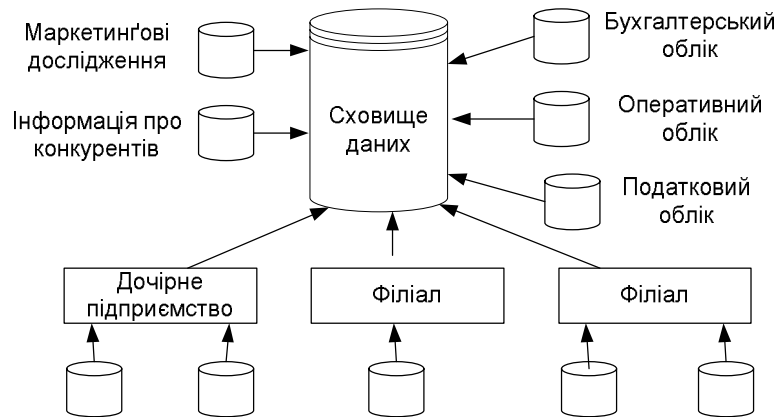


Рис. 1. Інтеграція даних з різних джерел.

- *застосування* – це рішення, створені постачальниками відповідно до вимог клієнтів, які використовують один або більше продуктів інтеграції даних;
- *продукти* – це готові комерційні рішення, що підтримують одну або більше технологій інтеграції даних;
- *технології* реалізують один або більше методів інтеграції даних;
- *методи* – це підходи до інтеграції даних, що не залежать від технологій.

Проблема інтеграції корпоративної інформації, даних і застосувань залишається актуальною для багатьох організацій. Із зростанням обсягу інформації завдання об'єднання розрізнених структур, таких як вітрини, бази або сховища даних, стає життєво важливою для багатьох компаній. Є три технології, які можуть допомогти в цьому – три «І» (або три «Е» в англійському варіанті). Це *інтеграція корпоративних застосувань* (*enterprise application integration, EAI*), *інтеграція корпоративної інформації* (*enterprise information integration, EII*) і *програмне забезпечення для витягання, перетворення і завантаження даних* (*extract, transform and load, ETL*).

Ці технології можуть використовуватись для широкого кола завдань: від інтеграції в режимі реального часу до пакетної інтеграції і від інтеграції даних до інтеграції застосувань. Для інтеграції даних в режимі реального часу найкраще підходить технологія ЕІІ. Для пакетної інтеграції даних – ETL. А для інтеграції застосувань – консолідація в режимі реального часу або пакетна, найкращим інструментом є технологія EAI.

- **EAI** – це технологія, за допомогою якої організація добивається централізації і оптимізації інтеграції корпоративних застосувань, зазвичай використовуючи ті або інші форми технології оперативної доставки інформації (*push technology*), яка керується зовнішніми подіями (*event-driven*);
- **ETL** – це технологія, яка перетворить дані (зазвичай за допомогою їх пакетного опрацювання) з операційного середовища, що включає гетерогенні технології, в інтегровані дані, що узгоджуються між собою, придатні для використання в процесі підтримки прийняття рішень; ETL-технологія орієнтована на бази даних, наприклад, сховище, вітрину або операційне сховище даних;
- **ЕІІ** – це технологія для інтеграції в режимі реального часу незіставних типів даних з численних джерел як всередині, так і за межами корпорації; інструменти ЕІІ забезпечують універсальний рівень доступу до даних і використовують технологію пошуку інформації (*pull technology*) або можливості роботи за запитами; технологія ЕІІ орієнтована на конкретних співробітників, які одержують інформацію через інструментальну панель або звіт.

Далі необхідно розглянути місце цих технологій у вже наявній архітектурі. Технологія EAI інтегрує транзакції двох або більше застосувань, технологія ETL інтегрує дані операційних систем і компонентів підтримки прийняття рішень, а технологія ЕІІ здійснює віртуальну інтеграцію даних з різних джерел.

Технологія ЕАІ найбільш функціональна тоді, коли необхідно зв'язати застосування у реальному часі для автоматизації процесів предметної області. Другий випадок застосування ЕАІ – це ситуація, коли необхідно, щоб зміни, внесені в одне застосування (зазвичай це невеликий набір записів), були відображені у всіх інших. Ця технологія дуже добре справляється із завданням фіксації змін і їх перенесення у відповідні застосування або системи.

Технологія ETL виявляється найкориснішою в тих випадках, коли необхідно створити сховище даних, що містить добре документовані і надійні дані для історичного аналізу, наприклад, для аналізу часових рядів або багатовимірних запитів. Ця технологія також використовується для інтеграції ключових довідкових даних. Технологія ETL незамінна для таких завдань, як видалення даних, що дублюються, здійснення процесів перевірки якості даних і т.ін. Ці інструменти також використовуються для створення окремих вітрин даних, що обслуговують конкретний відділ або бізнес-процес або призначених для яких-небудь довгострокових цілей. Інструменти ETL дають користувачу можливість запустити процеси, що повторюються, для більшої злагодженості дій і можливості їх багатократного використання. Такі процеси включають створення точних технічних метаданих, що підтримують загальну цілісність середовища business intelligence (BI).

Стандартизація послідовності операцій при завантаженні СД, враховуючи важливість і вартість багатьох рішень ETL, дозволить уникнути повторення помилок, зроблених в попередніх розробленнях. Крім того, досвід розроблення ETL виявив загальні частини ETL-процесів при завантаженні різнорідних джерел, що дозволяє говорити про одноманітність підходу до розроблення ETL для джерел даних довільного походження.

Загалом застосування ETL витягають інформацію з початкової бази даних, перетворюють її у формат, що підтримується базою даних призначення, а потім завантажують в неї перетворену інформацію. Щоб ініціювати процес ETL, застосовуються програми витягання даних для читання записів в початковій базі даних і для підготовки інформації, що зберігається в цих записах, до процесу перетворення. Щоб витягнути дані з початкової бази даних, можна вибрати один з трьох варіантів – створити власні програми, звернутися до готового спеціалізованого інструментарію ETL або використати поєднання і того, і іншого. У виробничому центрі Datagu створений власний оригінальний інструментарій, який дозволяє поєднувати технологічні прийоми, що склалися в Datagu, з програмами незалежних розробників, які виконують спеціалізовані функції, унікальні для конкретного оточення.

З активним розповсюдженням Internet процеси ETL стали все ширше застосовуватися і для підтримки Web-застосувань. Наприклад, постачальник може використовувати засоби ETL для «закачування» у Web-систему даних, необхідних для перевірки стану опрацювання замовлень, з внутрішніх програм. Програми, що реалізують підхід ETL, стають важливими компонентами багатьох ініціатив, пов'язаних з електронною комерцією, зокрема застосувань, що підтримують взаємодію між партнерами бізнесу, а також між виробником і споживачами.

У багатьох випадках наявний інструментарій ETL здатний задовольнити велику частину вимог до перенесення даних. Описавши в цілому завдання і переваги ETL-процесів, розглянемо їх місце і алгоритм роботи в процесі побудови сховищ даних.

Порядок виконання роботи

1. Підготовка і отримання вхідних даних з декількох джерел.

Зміст завдання: сформувати декілька (3-4) таблиць (документів) які містять однотипні дані отримані з різних джерел інформації, зображені в різних форматах та структурах (MS Access, Visual Foxpro, dBase, Ms Excel, структурований текст тощо), наприклад,

- Джерело 1 –таблиця MS Access у вигляді конструктора

documents : таблица		
	Имя поля	Тип данных
		Счетчик
	evdate	Дата/время
	employee_id	Числовой
	document_type	Числовой
	dept_id	Числовой
	tovar_id	Числовой
	count	Числовой
	price	Денежный
	suma	Денежный
	type	Числовой

documents									
Код	Дата	Працівник	Тип документа	Підрозділ	Товар	Кількість	Ціна	Сума	Тип операції
7	12.02.2007	4	4	3		1	1 000,00 грн.	1 000,00 грн.	розхід
8	13.02.2007	2	4	3		1	2 000,00 грн.	2 000,00 грн.	розхід
9	14.02.2007	3	4	3		1	3 000,00 грн.	3 000,00 грн.	розхід
10	15.02.2007	1	4	3		1	4 000,00 грн.	4 000,00 грн.	розхід

- Джерело2 –таблица MS Access у вигляді конструктора

source : таблица		
	Имя поля	Тип данных
		Счетчик
	evdate	Дата/время
	employee_id	Числовой
	document_type	Числовой
	dept_id	Числовой
	tovar_id	Числовой
	count	Числовой
	price	Денежный
	suma	Денежный
	type	Числовой

source									
Код	Дата	Працівник	Тип документа	Підрозділ	Товар	Кількість	Ціна	Сума	Тип операції
7	12.05.2007	4	2	3		1	1 000,00 грн.	1 000,00 грн.	прихід
8	13.05.2007	2	2	3		1	2 000,00 грн.	2 000,00 грн.	прихід
9	14.05.2007	3	2	3		1	3 000,00 грн.	3 000,00 грн.	прихід
10	15.05.2007	1	2	3		1	4 000,00 грн.	4 000,00 грн.	прихід

- Джерело 3 - Лист Excel по авансових звітах

Код	Дата	Працівник	Тип документа	Підрозділ	Товар	Кількість	Ціна	Сума	Тип операції
7	12.02.2007	4	3	5		1	1 000,00 грн.	1 000,00 грн.	розхід
8	13.02.2007	2	3	5		1	2 000,00 грн.	2 000,00 грн.	розхід
9	14.02.2007	3	3	5		1	3 000,00 грн.	3 000,00 грн.	розхід
10	15.02.2007	1	3	5		1	4 000,00 грн.	4 000,00 грн.	розхід

Связь с электронной таблицей

Файл электронной таблицы содержит несколько листов или диапазонов. Выберите нужный объект.

☒ листы
☐ именованные диапазоны

Лист1
 Лист2
 Лист3

Образцы данных для листа "Лист1".

	Код	Дата	Працівник	Тип документа	Підрозділ	Товар	Кіл
1	7	12.02.2007	4	3	5		1
2	8	13.02.2007	2	3	5		1
3	9	14.02.2007	3	3	5		1
4	10	15.02.2007	1	3	5		1

Связь с электронной таблицей

Установите этот флажок для использования данных из первой строки в качестве имен полей таблицы.

☒ Первая строка содержит заголовки столбцов

	Код	Дата	Працівник	Тип документа	Підрозділ	Товар	Кіл
1	7	12.02.2007	4	3	5		1
2	8	13.02.2007	2	3	5		1
3	9	14.02.2007	3	3	5		1
4	10	15.02.2007	1	3	5		1

- Джерело 4 - Лист Excel по касових виписках

Код	Дата	Праці вник	Тип докумен- та	Підрозд іл	Това р	Кількі сть	Ціна	Сума	Тип операції
30	12.05.2007	4	4	6		1	200,00 грн.	200,00 грн.	розхід
31	13.05.2007	2	4	6		1	400,00 грн.	400,00 грн.	розхід
32	14.05.2007	3	4	6		1	400,00 грн.	400,00 грн.	розхід
33	15.05.2007	1	4	6		1	200,00 грн.	200,00 грн.	розхід

- Джерело 5 – текстовий файл (файл з розділювачами)

60	12.06.07	3	1	4	10	3	3	9	-1
62	14.06.07	1	1	4	3	7	10	70	-1

Як видно з таблиці метаданих, джерела містять однотипну інформацію, яка має спільний характер, але відрізняється складом, способами подання і форматами. На основі цих метаданих може бути утворена інтегрована таблиця, яка буде виконувати функції оперативного сховища даних.

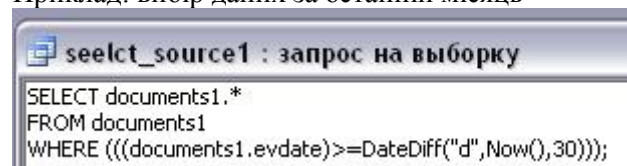
2. Визначення вимог до вхідних даних і критеріїв перетворення

Зміст завдання: Для попередньо отриманих і збережених у оперативному сховищі даних розробити систему норм і критеріїв перетворення їх до форми, придатної для збереження у сховищі даних.

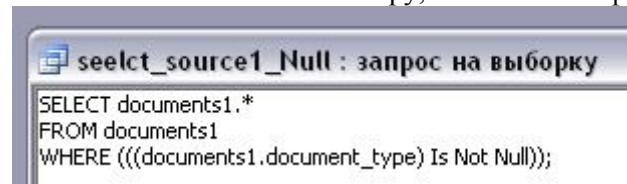
Приклади критеріїв та вимог:

– обмеження часового періоду фактів "від ... – до ...";

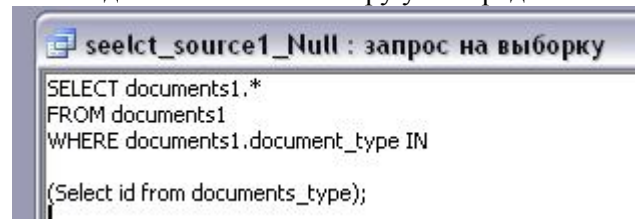
Приклад: вибір даних за останній місяць



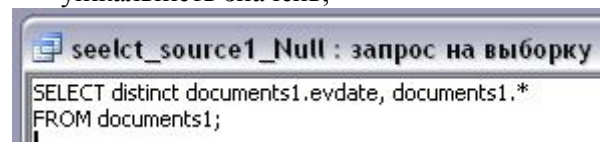
– обов'язкове значення виміру, чи показника факту;



– входження значень виміру у попередньо визначену множину чи область визначення;



– унікальність значень;



– дотримання правил синтаксису;

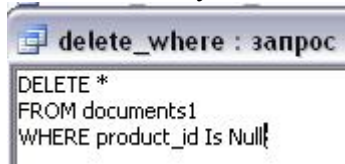
– семантичні обмеження.

3. Аналіз, фільтрація та перетворення вхідних даних

Зміст завдання: Сформувати та виконати стосовно таблиці оперативного сховища даних запити на вилучення або зміну даних, які не відповідають критеріям, побудованим у п. 2

Наприклад,

- вилучення даних, які не містять обов'язкових значень і, внаслідок цього не можуть бути застосовані у сховищі даних:



- виправлення некоректних чи помилкових значень (наприклад, коли у джерела даних внесено не коди підрозділів, а назви

```
Public Sub update_fact()
Dim str As String
Dim rs As Recordset

Dim str1 As String
Dim rs1 As Recordset
'вибираємо всі значення назв підрозділів
Set rs = CurrentDb.Execute("select distinct dept_name from documents1")
While Not rs.EOF
    ' визначаємо код підрозділу за назвою
    str1 = "select id from dept where name=" & rs!dept_name
    Set rs1 = CurrentDb.execute(str)
    ' замінюємо назву на знайдений код
    str1 = "update documents1 set dept_name=" & rs1!id & " where dept_name=" & rs!dept_name
    rs.MoveNext
Wend

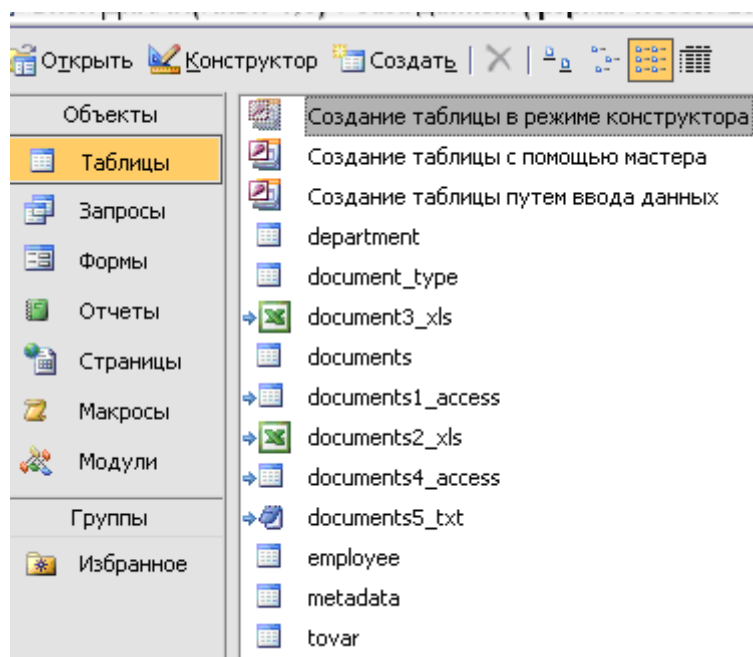
End Sub
```

4. Завантаження таблиці фактів

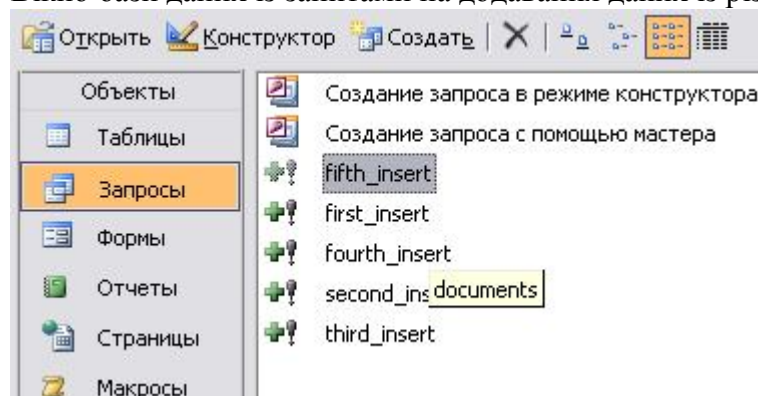
Зміст завдання: Створити та виконати запит (процедуру) переміщення значень з підготовленої таблиці вхідних даних сховища оперативних даних у таблицю фактів та провести аналіз виконаних дій.

- підготовка вхідних даних: внести зміни в склад, структуру і зміст джерел даних (по мірі потреби доповнити новими стовпчиками, вилучити зайві чи змінити параметри стовпчика, внести відсутні значення), узгодивши їх відповідності з таблицею метаданих та структурою ОСД;
- створити таблицю ОСД;
- приєднати визначені джерела даних в якості зовнішніх таблиць;
- сформулювати та реалізувати запити (процедури) для перенесення даних зовнішніх джерел в ОСД

Приклад:



Вікно бази даних із запитами на додавання даних із різних джерел



Запит на додавання даних із текстового файлу:

```
INSERT INTO documents ( evdate, employee_id, document_type, dept_id, tovar_id, [count], price,
suma, type )
SELECT Поле2, Поле3, Поле4, Поле5, Поле6, Поле7, Поле8, Поле9, Поле10
FROM documents5_txt;
```

Запит на додавання даних із файлу MS Access – джерело 1

```
INSERT INTO documents ( evdate, employee_id, document_type, dept_id, tovar_id, [count], price,
suma, type )
SELECT documents1_access.evdate AS Выражение1, documents1_access.employee_id AS
Выражение2, documents1_access.document_type AS Выражение3, documents1_access.dept_id
AS Выражение4, documents1_access.tovar_id AS Выражение5, documents1_access.count AS
Выражение6, documents1_access.price AS Выражение7, documents1_access.suma AS
Выражение8, documents1_access.type AS Выражение9
FROM documents1_access;
```

Запит на додавання даних із електронної таблиці – джерело 2

```
INSERT INTO documents ( evdate, employee_id, document_type, dept_id, tovar_id, [count], price,
suma, type )
SELECT documents1_access.evdate AS Выражение1, documents1_access.employee_id AS
Выражение2, documents1_access.document_type AS Выражение3, documents1_access.dept_id
AS Выражение4, documents1_access.tovar_id AS Выражение5, documents1_access.count AS
Выражение6, documents1_access.price AS Выражение7, documents1_access.suma AS
Выражение8, documents1_access.type AS Выражение9
FROM documents1_access;
```

Запит на додавання даних із бази даних – джерело 4

```
INSERT INTO documents ( evdate, employee_id, document_type, dept_id, tovar_id, [count], price,
suma, type )
SELECT documents4_access.evdate AS Выражение1, documents4_access.employee_id AS
Выражение2, documents4_access.document_type AS Выражение3, documents4_access.dept_id
AS Выражение4, documents4_access.tovar_id AS Выражение5, documents4_access.count AS
Выражение6, documents4_access.price AS Выражение7, documents4_access.suma AS
Выражение8, documents4_access.type AS Выражение9
FROM documents4_access;
```

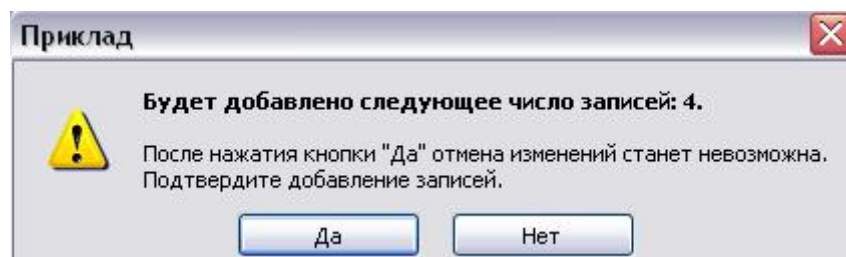
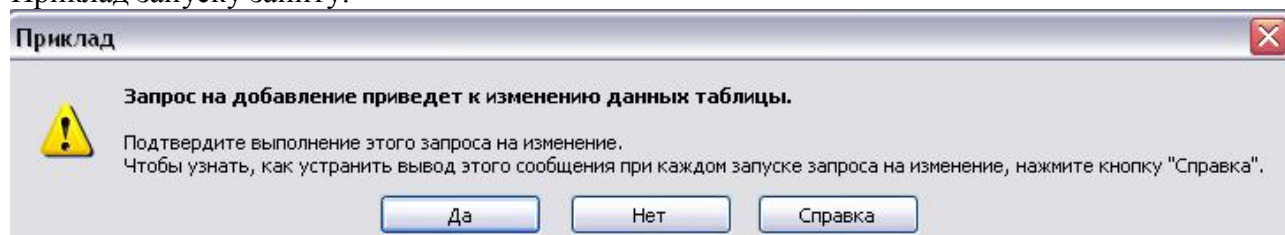
Запит на додавання даних із електронної таблиці – джерело 3

```
INSERT INTO documents ( evdate, employee_id, document_type, dept_id, tovar_id, [count], price,
suma, type )
```



```
SELECT Дата, Працівник, [Тип документа], Підрозділ, Товар, Кількість, Ціна, Сума, iif([Тип операції]="розхід",-1,1) AS type1
FROM document3_xls;
```

Приклад запуску запиту:



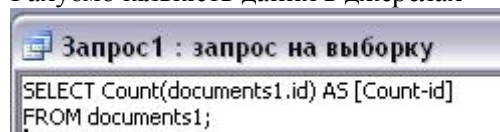
5. Аналіз та верифікація сховища даних

Зміст завдання: перевірити та обґрунтувати працездатність сховища даних, для цього

– перевірити наявність всіх необхідних значень у стовпчиках таблиці фактів;

Приклад:

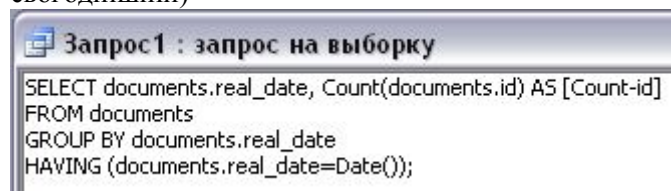
Рахуємо кількість даних в джерелах



І т.д.

Додаємо отримані кількості,

Рахуємо кількість вставлених у таблицю сховища записів (дата внесення цих записів рівна сьогоднішній)



Якщо суми рівні, то ETL пройшов без перешкод

- перевірити таблицю фактів на наявність "підвішених" рядків, що виникають при наявності в ній значень кодів, відсутніх у таблицях вимірів;
- перевірити таблицю фактів на наявність повторюваних рядків.

Приклад:

Нехай є така таблиця фактів:

documents										
id	evdate	employee_id	documents_type	delp_id	товар_id	count	price	suma	type	real_date
1	12.02.2007	вахненко	1	3	1	1	1 000,00 грн.	1 000,00 грн.	-1	
2	13.02.2007	василенко	1	3	2	1	2 000,00 грн.	2 000,00 грн.	-1	
3	15.02.2007	іваненко	1	3	3	1	3 000,00 грн.	3 000,00 грн.	-1	

documents										
id	evdate	employee_id	documents_type	delp_id	tovar_id	count	price	suma	type	real_date
4	15.02.2007	петренко	4	3		1	4 000,00 грн.	4 000,00 грн.	-1	
9	15.02.2007	іваненко	1	3	3	1	3 000,00 грн.	3 000,00 грн.	-1	
11	15.02.2007	іваненко	1	3	3	1	3 000,00 грн.	3 000,00 грн.	-1	

Бачимо, що рядки повторюються за винятком значення ключів. Необхідно вилучити такі повтори.

Спершу визначаємо записи, що повторюються.

```

повтор : запрос на выборку
SELECT documents.evdate, documents.employee_id, documents.documents_type, documents.delp_id,
documents.tovar_id, documents.count, documents.price, documents.suma, documents.type, documents.id,
documents.real_date
FROM documents
WHERE (((documents.evdate) In (SELECT [evdate] FROM [documents] As Tmp GROUP BY
[evdate],[employee_id],[documents_type],[delp_id],[tovar_id],[count],[price],[suma],[type] HAVING Count(*)>1 And
[employee_id] = [documents].[employee_id] And [documents_type] = [documents].[documents_type] And [delp_id] =
[documents].[delp_id] And [tovar_id] = [documents].[tovar_id] And [count] = [documents].[count] And [price] =
[documents].[price] And [suma] = [documents].[suma] And [type] = [documents].[type])))
ORDER BY documents.evdate, documents.employee_id, documents.documents_type, documents.delp_id,
documents.tovar_id, documents.count, documents.price, documents.suma, documents.type;

```

Далі видаляємо ті записи, що повторюються, залишивши один з них (наприклад перший)

```

Запрос 2 : запрос на выборку
SELECT Min(povtor.id) AS [Min-id], povtor.evdate, povtor.employee_id, povtor.documents_type,
povtor.delp_id, povtor.tovar_id, povtor.count, povtor.price, povtor.suma, povtor.type,
povtor.real_date
FROM povtor
GROUP BY povtor.evdate, povtor.employee_id, povtor.documents_type, povtor.delp_id,
povtor.tovar_id, povtor.count, povtor.price, povtor.suma, povtor.type, povtor.real_date;

'povtor' без подчиненных в 'Запрос2' : запрос на выборку
SELECT povtor.evdate, povtor.employee_id, povtor.documents_type, povtor.delp_id, povtor.tovar_id,
povtor.count, povtor.price, povtor.suma, povtor.type, povtor.id, povtor.real_date
FROM povtor LEFT JOIN Запрос2 ON povtor.id = Запрос2.[Min-id]
WHERE (((Запрос2.[Min-id]) Is Null));

```

'povtor' без подчиненных в 'Запрос2'										
evdate	employee_id	documents_type	delp_id	tovar_id	count	price	suma	type	id	real_date
15.02.2007	іваненко	1	3	3	1	3 000,00 грн.	3 000,00 грн.	-1	9	
15.02.2007	іваненко	1	3	3	1	3 000,00 грн.	3 000,00 грн.	-1	11	

Ці записи знищуємо.

```

Запрос 3 : запрос на удаление
delete from documents where id in
(select id from ['povtor' без подчиненных в 'Запрос2'])

```

- виконати тестові запити на отримання контрольних підсумкових значень, таких як підрахунок кількості фактів по певному виміру, сумарного значення по певному виміру, тощо і звірити отримані результати з фактичними
- на підставі отриманих результатів зробити висновки про придатність чи непридатність побудованого сховища даних до застосування.

Зміст звіту по роботі:

1. Тема лабораторної роботи.
2. Завдання та постановка задачі лабораторної роботи.
3. Теоретична підготовка.
4. Опис виконаної роботи та отриманих результатів по кожному з пунктів завдання:
 - перелік, опис та обґрунтування критеріїв і вимог до вхідних даних;
 - запити(процедури) на аналіз та фільтрацію вхідних даних;
 - приклади виконання дій з виявлення та фільтрації невідповідних значень;
 - склад, структура та зміст таблиці вхідних даних до її перетворення та після;
 - запити (процедури) завантаження вхідних даних у таблицю фактів сховища даних
 - результати завантаження сховища даних;
5. Висновки.

Лабораторна робота № 6. Моделювання процесів оперативного аналізу даних

Мета роботи: Вивчення порядку, методів та засобів створення аналітичних та підсумкових даних і документів на основі технології OLAP.

Теоретичні відомості

OLAP (On-Line Analytical Processing) – системи аналітичного опрацювання, також відомі як системи підтримки прийняття рішення СППР (Decision Support System, DSS), орієнтовані на надання користувачам могутніх механізмів для швидкого і багатобічного аналізу даних. Причина використання OLAP для опрацювання запитів — це швидкість. Реляційні БД зберігають суть в окремих таблицях, які зазвичай добре нормалізовані. Ця структура зручна для операційних БД (системи OLTP), але складні багатотабличні запити в ній виконуються відносно поволі. OLAP робить миттєвий знімок реляційної БД і структурує її в просторову модель для запитів. Заявлений час опрацювання запитів в OLAP складає близько 0.1 % від аналогічних запитів до реляційної БД.

OLAP-структура, створена з робочих даних, називається OLAP-куб (OLAP-сховище). Куб створюється із з'єднання таблиць із застосуванням схеми «зірка».

Хоча OLAP і не є необхідним атрибутом сховища даних, він все частіше застосовується для аналізу накопичених у цьому сховищі відомостей.

Компоненти, що входять у типове сховище, подані на рис. 1.

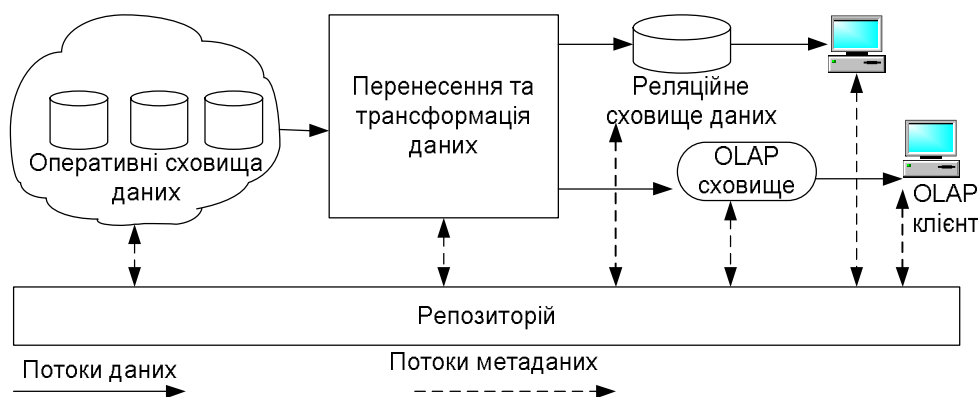


Рис. 1. Структура сховища даних з OLAP.

Оперативні дані збираються з різних джерел, очищаються, інтегруються і переміщуються у реляційне сховище. При цьому вони вже доступні для аналізу за допомогою різних засобів побудови звітів. Потім дані (повністю або частково) готуються для OLAP-аналізу. Вони можуть бути завантажені до спеціальної БД OLAP або залишені у реляційному сховищі. Найважливішим його елементом є метадані, тобто інформація про структуру, розміщення і трансформацію даних. Завдяки ним забезпечується ефективна взаємодія різних компонентів сховища.

Підсумовуючи, можна визначити *OLAP* як сукупність засобів багатовимірного аналізу даних, накопичених в сховищі. Теоретично засоби OLAP можна застосовувати і безпосередньо до

оперативних даних або їх точних копій (щоб не заважати оперативним користувачам). Але тоді виникає ризик певної непослідовності, а саме, аналізу оперативних даних, які безпосередньо для аналізу непридатні.

Види OLAP-рішень подані в таблиці 1.

Таблиця 1.

Вплив OLAP на керівні рішення

Тип аналізу	Вплив на керівні рішення
Аналіз росту продажів (по підприємствах)	Планування і оптимізація сезонних продажів різноманітних продуктів
Аналіз росту продажів (по клієнтах підприємств)	Поточне планування доставки продуктів підприємствам
Аналіз регіональних продажів через підприємства	Ефективніше керування ресурсами на підприємстві (час доставки, кількість, внутрішні переміщення між підприємствами).
Аналіз прибутковості продажів (по підприємствах)	
Аналіз росту цінних паперів	
Аналіз продажів і торгових структур	
Аналіз «продаж -запас-постачання»	
Аналіз відхилень у продажах і постачанні, заплановано/у наявності (по підприємствах)	Необхідна попередня оцінка результатів діяльності підприємства; з метою своєчасного виправлення відхилень і коригування продажів з виробничим планом
Аналіз росту цін (по підприємствах)	Визначення оптимального відношення між попитом і пропозицією
Аналіз структури клієнтів підприємства	Цей аналіз проводиться з метою оптимізації умов для найбільших клієнтів і визначення оптимального рівня цін
Аналіз структури і росту одержання рахунків дебіторів (по підприємствам)	Цей аналіз проводиться для рішення питання з рахунками дебіторів, що безпосередньо пов'язано з керуванням рухом грошових коштів; установлення різних кредитних ліній для різних клієнтів; визначення взаємозв'язку між знижками й умовами платежу
Аналіз руху вхідних і вихідних коштів(по підприємствам)	Необхідно керувати рухом грошових коштів компанії

Порядок виконання роботи

1. Детальний аналіз інформації

Зміст завдання: вибір даних за вимогами структурних одиниць та конкретних посад з метою розмежування даних.

Приклади

Запитання	Текст	Результат				
Визначити підрозідли та документи у них, сума яких лежить у межах [1000;3000]	SELECT department.name, documents.evdate, documents.employee_id, documents.documents_type, documents.suma FROM department INNER JOIN documents ON department.id = documents.delp_id WHERE documents.suma	select_between				
		name	evdate	employee_id	Documents_type	suma
		бухгалтерія	12.02.2007	4	4	1 000,00 грн.
		бухгалтерія	13.02.2007	2	4	2 000,00 грн.
		бухгалтерія	14.02.2007	3	4	3 000,00 грн.

	Between 1000 And 3000;						
Визначити документи по підрозділа 2, 3 (каса, бухгалтерія)	SELECT documents.evdate, documents.employee_id, documents.documents_type, documents.delp_id FROM documents WHERE documents.delp_id In (2,3);	select_in					
		evdate	employee_id	documents_type	delp_id		
		12.02.2007	4	4	3		
		13.02.2007	2	4	3		
		14.02.2007	3	4	3		
		15.02.2007	1	4	3		
		15.04.2007		2	2		
Визначити підрозділи, назви яких починаються на «К»	SELECT department.id, department.name, department.prior_id FROM department WHERE department.name Like "к*";	select_like					
		id	name	prior_id			
		2	каса	1			
Вивести всі документи підлеглих підрозділів	SELECT documents.evdate, documents.employee_id, documents.documents_type, documents.delp_id FROM documents WHERE documents.delp_id In (select id from department where prior_id is Not Null);	select_in_null					
		evdate	employee_id	documents_type	delp_id		
		12.02.2007	4	4	3		
		13.02.2007	2	4	3		
		14.02.2007	3	4	3		
		15.02.2007	1	4	3		
		15.04.2007		2	2		
Вибрати всі документи по розходу (-1), у яких вказані працівники – inner join,	SELECT employee.surname, employee.name, documents.documents_type, documents.evdate, documents.suma, documents.type FROM employee INNER JOIN documents ON employee.id = documents.employee_id WHERE (((documents.type)=-1));	seelct_inner_join					
		surname	name	Documents_type	evdate	suma	type
		вахненко	марія	4	12.02.2007	1 000,00 грн.	-1
		василенко	василь	4	13.02.2007	2 000,00 грн.	-1
		іваненко	петро	4	14.02.2007	3 000,00 грн.	-1
		петренко	іван	4	15.02.2007	4 000,00 грн.	-1
Вибрати всі документи по розходу з вказанням працівника	SELECT employee.surname, employee.name, documents.documents_type, documents.evdate, documents.suma, documents.type FROM employee RIGHT JOIN documents ON employee.id = documents.employee_id WHERE (((documents.type)=-1));	select_left_join					
		surname	name	documents_type	evdate	suma	type
		вахненко	марія	4	12.02.2007	1 000,00 грн.	-1
		василенко	василь	4	13.02.2007	2 000,00 грн.	-1
		іваненко	петро	4	14.02.2007	3 000,00 грн.	-1
		петренко	іван	4	15.02.2007	4 000,00	-1

				грн.	
		2	04.04.2007	300,00	-1
				грн.	

2. Агрегування таблиці фактів

Зміст завдання: Створити набори підсумкових даних з низьким ступенем агрегування. Описати за допомогою запитів процедури створення груп фактів за значеннями одного, двох, трьох і більше вимірів. Розробити запити з високим рівнем агрегації – кількість агрегатів набагато більша, ніж кількість детальних даних.

Запитання	Текст	Результат																																																						
Порахувати кількість документів по днях	SELECT documents.evdate, Count(documents.id) AS [Count documents] FROM documents GROUP BY documents.evdate;	<table><tr><th colspan="2">group_count</th></tr><tr><th>evdate</th><th>Count documents</th></tr><tr><td>12.02.2007</td><td>1</td></tr><tr><td>13.02.2007</td><td>1</td></tr><tr><td>15.02.2007</td><td>2</td></tr><tr><td>04.04.2007</td><td>2</td></tr></table>	group_count		evdate	Count documents	12.02.2007	1	13.02.2007	1	15.02.2007	2	04.04.2007	2																																										
group_count																																																								
evdate	Count documents																																																							
12.02.2007	1																																																							
13.02.2007	1																																																							
15.02.2007	2																																																							
04.04.2007	2																																																							
На кожен день визначити максимальну суму, що зустрічається в документі	SELECT documents.evdate, Max(documents.suma) AS [Max-suma] FROM documents GROUP BY documents.evdate;	<table><tr><th colspan="2">group_max</th></tr><tr><th>evdate</th><th>Max-suma</th></tr><tr><td>12.02.2007</td><td>1 000,00 грн.</td></tr><tr><td>13.02.2007</td><td>2 000,00 грн.</td></tr><tr><td>15.02.2007</td><td>4 000,00 грн.</td></tr><tr><td>04.04.2007</td><td>300,00 грн.</td></tr></table>	group_max		evdate	Max-suma	12.02.2007	1 000,00 грн.	13.02.2007	2 000,00 грн.	15.02.2007	4 000,00 грн.	04.04.2007	300,00 грн.																																										
group_max																																																								
evdate	Max-suma																																																							
12.02.2007	1 000,00 грн.																																																							
13.02.2007	2 000,00 грн.																																																							
15.02.2007	4 000,00 грн.																																																							
04.04.2007	300,00 грн.																																																							
Порахувати кількість документів по днях та вибрати ті дні, у яких кількість документів >=2	SELECT documents.evdate, Count(documents.id) AS [Count documents] FROM documents GROUP BY documents.evdate having count(id)>=2	<table><tr><th colspan="2">group_count_having</th></tr><tr><th>evdate</th><th>Count documents</th></tr><tr><td>15.02.2007</td><td>2</td></tr><tr><td>04.04.2007</td><td>2</td></tr></table>	group_count_having		evdate	Count documents	15.02.2007	2	04.04.2007	2																																														
group_count_having																																																								
evdate	Count documents																																																							
15.02.2007	2																																																							
04.04.2007	2																																																							
На кожен день визначити максимальну суму, що зустрічається в документі про розхід	SELECT documents.evdate, Max(documents.suma) AS [Max-suma] FROM documents GROUP BY documents.evdate, type having type=-1;	<table><tr><th colspan="2">group_max</th></tr><tr><th>evdate</th><th>Max-suma</th></tr><tr><td>12.02.2007</td><td>1 000,00 грн.</td></tr><tr><td>13.02.2007</td><td>2 000,00 грн.</td></tr><tr><td>15.02.2007</td><td>4 000,00 грн.</td></tr><tr><td>04.04.2007</td><td>200,00 грн.</td></tr></table>	group_max		evdate	Max-suma	12.02.2007	1 000,00 грн.	13.02.2007	2 000,00 грн.	15.02.2007	4 000,00 грн.	04.04.2007	200,00 грн.																																										
group_max																																																								
evdate	Max-suma																																																							
12.02.2007	1 000,00 грн.																																																							
13.02.2007	2 000,00 грн.																																																							
15.02.2007	4 000,00 грн.																																																							
04.04.2007	200,00 грн.																																																							
Вибрати всі документи по торгівлі (Накладні) та банківські виписки	SELECT documents.id, documents.evdate, documents.delp_id, documents.tovar_id, documents.count, documents.price, documents.suma*documents.type as [suma], IIF(documents.type=1,»Прихід»,»Розхід») as [Тип операції], «Накладна» as [type] FROM documents WHERE documents.documents_type=1	<table><tr><th colspan="9">union</th></tr><tr><th>id</th><th>evdate</th><th>delp_id</th><th>tovar_id</th><th>count</th><th>price</th><th>suma</th><th>Тип операції</th><th>type</th></tr><tr><td>1</td><td>12.02.2007</td><td>3</td><td>1</td><td>1</td><td>1 000,00 грн.</td><td>-1 000,00 грн.</td><td>Розхід</td><td>Накладна</td></tr><tr><td>2</td><td>13.02.2007</td><td>3</td><td>2</td><td>1</td><td>2 000,00 грн.</td><td>-2 000,00 грн.</td><td>Розхід</td><td>Накладна</td></tr><tr><td>3</td><td>15.02.2007</td><td>3</td><td>3</td><td>1</td><td>3 000,00 грн.</td><td>-3 000,00 грн.</td><td>Розхід</td><td>Накладна</td></tr><tr><td>5</td><td>04.04.2007</td><td>2</td><td></td><td></td><td></td><td>-</td><td>Розхід</td><td>Банків</td></tr></table>	union									id	evdate	delp_id	tovar_id	count	price	suma	Тип операції	type	1	12.02.2007	3	1	1	1 000,00 грн.	-1 000,00 грн.	Розхід	Накладна	2	13.02.2007	3	2	1	2 000,00 грн.	-2 000,00 грн.	Розхід	Накладна	3	15.02.2007	3	3	1	3 000,00 грн.	-3 000,00 грн.	Розхід	Накладна	5	04.04.2007	2				-	Розхід	Банків
union																																																								
id	evdate	delp_id	tovar_id	count	price	suma	Тип операції	type																																																
1	12.02.2007	3	1	1	1 000,00 грн.	-1 000,00 грн.	Розхід	Накладна																																																
2	13.02.2007	3	2	1	2 000,00 грн.	-2 000,00 грн.	Розхід	Накладна																																																
3	15.02.2007	3	3	1	3 000,00 грн.	-3 000,00 грн.	Розхід	Накладна																																																
5	04.04.2007	2				-	Розхід	Банків																																																

	UNION ALL SELECT documents.id, documents.evdate, documents.delp_id, NULL as tovar_id, NULL as [count],NULL as [price], documents.suma*documents.ty pe as [suma], IIF(documents.type=1,»Прихі д»,»Розхід») as [Тип операції], «Банківська виписка» as [type] FROM documents WHERE documents.documents_type=2	<table><tr><td></td><td>007</td><td></td><td></td><td></td><td></td><td>200, д 00 грн.</td><td></td><td>ська виписк а</td></tr><tr><td>6</td><td>04.04.2 007</td><td></td><td></td><td></td><td></td><td>300, Прихі 00 д грн.</td><td></td><td>Банків ська виписк а</td></tr></table>		007					200, д 00 грн.		ська виписк а	6	04.04.2 007					300, Прихі 00 д грн.		Банків ська виписк а																																													
	007					200, д 00 грн.		ська виписк а																																																									
6	04.04.2 007					300, Прихі 00 д грн.		Банків ська виписк а																																																									
Вибрати всі документи по торгівлі (Накладні) та банківські виписки та впорядкувати за зростанням сум у межах одного типу документа	SELECT documents.id, documents.evdate, documents.delp_id, documents.tovar_id, documents.count, documents.price, documents.suma*documents.ty pe as [suma], IIF(documents.type=1,»Прихі д»,»Розхід») as [Тип операції], «Накладна» as [type] FROM documents WHERE documents.documents_type=1 order by suma UNION ALL SELECT documents.id, documents.evdate, documents.delp_id, NULL as tovar_id, NULL as [count],NULL as [price], documents.suma*documents.ty pe as [suma], IIF(documents.type=1,»Прихі д»,»Розхід») as [Тип операції], «Банківська виписка» as [type] FROM documents WHERE documents.documents_type=2 order by suma	<table><tr><th colspan="9">union_order</th></tr><tr><th>id</th><th>evdate</th><th>delp_id</th><th>tovar_id</th><th>count</th><th>price</th><th>suma</th><th>Тип операції</th><th>type</th></tr><tr><td>3</td><td>15.02.2 007</td><td>3</td><td>3</td><td>1</td><td>3 000, 00 грн.</td><td>-3 000, 00 грн.</td><td>Розхід</td><td>Накладна</td></tr><tr><td>2</td><td>13.02.2 007</td><td>3</td><td>2</td><td>1</td><td>2 000, 00 грн.</td><td>-2 000, 00 грн.</td><td>Розхід</td><td>Накладна</td></tr><tr><td>1</td><td>12.02.2 007</td><td>3</td><td>1</td><td>1</td><td>1 000, 00 грн.</td><td>-1 000, 00 грн.</td><td>Розхід</td><td>Накладна</td></tr><tr><td>5</td><td>04.04.2 007</td><td>2</td><td></td><td></td><td></td><td>- 200, 00 грн.</td><td>Розхід</td><td>Банківська виписка</td></tr><tr><td>6</td><td>04.04.2 007</td><td></td><td></td><td></td><td></td><td>300, Прихі 00 д грн.</td><td></td><td>Банківська виписка</td></tr></table>	union_order									id	evdate	delp_id	tovar_id	count	price	suma	Тип операції	type	3	15.02.2 007	3	3	1	3 000, 00 грн.	-3 000, 00 грн.	Розхід	Накладна	2	13.02.2 007	3	2	1	2 000, 00 грн.	-2 000, 00 грн.	Розхід	Накладна	1	12.02.2 007	3	1	1	1 000, 00 грн.	-1 000, 00 грн.	Розхід	Накладна	5	04.04.2 007	2				- 200, 00 грн.	Розхід	Банківська виписка	6	04.04.2 007					300, Прихі 00 д грн.		Банківська виписка
union_order																																																																	
id	evdate	delp_id	tovar_id	count	price	suma	Тип операції	type																																																									
3	15.02.2 007	3	3	1	3 000, 00 грн.	-3 000, 00 грн.	Розхід	Накладна																																																									
2	13.02.2 007	3	2	1	2 000, 00 грн.	-2 000, 00 грн.	Розхід	Накладна																																																									
1	12.02.2 007	3	1	1	1 000, 00 грн.	-1 000, 00 грн.	Розхід	Накладна																																																									
5	04.04.2 007	2				- 200, 00 грн.	Розхід	Банківська виписка																																																									
6	04.04.2 007					300, Прихі 00 д грн.		Банківська виписка																																																									
Порахувати к-сть документів, працівників, що продали певні	SELECT tovar.group, Count(documents.tovar_id) AS [Count-tovar_id], Count(documents.employee_id	<table><tr><th colspan="4">Запрос4</th></tr><tr><th>group</th><th>Count-tovar_id</th><th>Count-employee_id</th><th>Count-documents_type</th></tr></table>	Запрос4				group	Count-tovar_id	Count-employee_id	Count-documents_type																																																							
Запрос4																																																																	
group	Count-tovar_id	Count-employee_id	Count-documents_type																																																														

групи товарів) AS [Count-employee_id], Count(documents.documents_t ype) AS [Count- documents_type] FROM tovar INNER JOIN (department INNER JOIN documents ON department.id = documents.delp_id) ON tovar.id = documents.tovar_id GROUP BY tovar.group;	1	1	1	1
		2	2	2	2

3. Формування аналітичних звітів за допомогою технології Pivot table

Зміст завдання: Створити і виконати запити для формування підсумкових звітів за кожним з вимірів сховища даних – загальної кількості фактів, пов'язаних з певним значенням виміру, сумарного значення характеристики фактів за вимірами, середнього значення одного з атрибутів фактів за значеннями виміру. Для цього застосувати операції групування та агрегатні функції мови SQL – Count(*), Sum(*), Avg(*), можливо, інші. Результати запитів відобразити у вигляді текстових і графічних документів. Завдання виконується за наступними зразками:

Новая форма

Создание формы со сводной таблицей Excel.

Выберите в качестве источника данных таблицу или запрос:

\$dcr_cash

Конструктор
Мастер форм
Автоформа: в столбец
Автоформа: ленточная
Автоформа: табличная
Автоформа: сводная таблица
Автоформа: сводная диаграмма
Диаграмма
Сводная таблица

OK Отмена

\$dcr_cash

Перетащите сюда поля фильтра

evdate ▼

Общие итоги

+ -

reason_id ▼ partner_id ▼

Сумма "suma"

Общие итоги

+ -

Операції по без готівці:

	Підстава ▼	Партнер ▼		
	банк		Всього по	Grand Total
Дата ▼	партнер1			
31.12.2000		53,3	53,3	53,3
04.01.2001		0	0	0
08.03.2001	200		200	200
07.07.2001		-300	-300	-300
14.03.2002	300		300	300
Grand Total	500	-246,7	253,3	253,3

Microsoft Excel - Лист в Рух коштів: параметри

Файл Правка Вид Вставка Формат Сервис

Закрити и вернуться в Рух коштів: параметри

Arial 10 Ж К Ч

B5 fx 553,3

	A	B	C
1		Період	Операція
2		На поч.	Всього На кін
3	Підрозділ	Залишок	
4	склад	5187	5187
5	фірма	553,3	553,3
6	Grand Total	5740,3	5740,3
7			

Microsoft Excel - Лист в Готівкові та безготівкові операції

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Закрити и вернуться в Готівкові та безготівкові операції

Arial 10 Ж К Ч

B2 fx Техніка для бізнесу

	A	B	C	D	E	F
1		Клієнт	Працівник	Підстава	Тип	Тип оп.
2		Техніка для бізнесу			Всього по Техніка для бізнесу	Всього
3		Петренко І.А.		Всього по Петренко І.А.		
4		заміна CD				
5		гот	гот Итог			
6	Дата	1				
7	25.10.2007 18:16	30	30	30	30	30
8	Всього	30	30	30	30	30

Зміст звіту по роботі:

- Тема лабораторної роботи.
- Завдання та постановка задачі лабораторної роботи.
- Теоретична підготовка.
- Опис виконаної роботи та отриманих результатів по кожному з пунктів завдання:
 - перелік, опис та обґрунтування способів і порядку виконання аналізу за різною кількістю факторів;
 - текст SQL - запитів(процедур) для аналізу фактів за одним фактором для кожного з вимірів;
 - текст SQL - запитів(процедур) для аналізу фактів за двома факторами для 2-3 пар вимірів;
 - текст SQL - запитів(процедур) на створення багатомірних аналітичних кубів для 3 і більше вимірів;
 - текст SQL - запитів(процедур) на виконання зрізів багатомірного куба за значенням одного з вимірів;
 - приклади результатів виконання запитів розроблених у кожному з пунктів завдання.
 - вихідні документи з відображенням результатів аналізу в текстовій та графічній формах
- Висновки.