UNIT-I INTRODUCTION

Introduction – Data warehouse delivery method – system process – typical process flow within a data ware house – query management process – process architecture – meta data data marting.

Introduction

Data warehouse

- collection of the key pieces of information used to manage and direct the business for profitability
- a data warehouse is the data (meta/fact/dimension/aggregation) and the process managers (load / warehouse / query) that make information available, enabling people to make informed decisions

Example uses:

- deciding the level of stock on a supermarket shelf,
- focused targeting of customers for loyalty schemes and direct mail promotions
- strategic decision making on major market segments and company profitability

Data Warehouse Delivery Method

Design Considerations

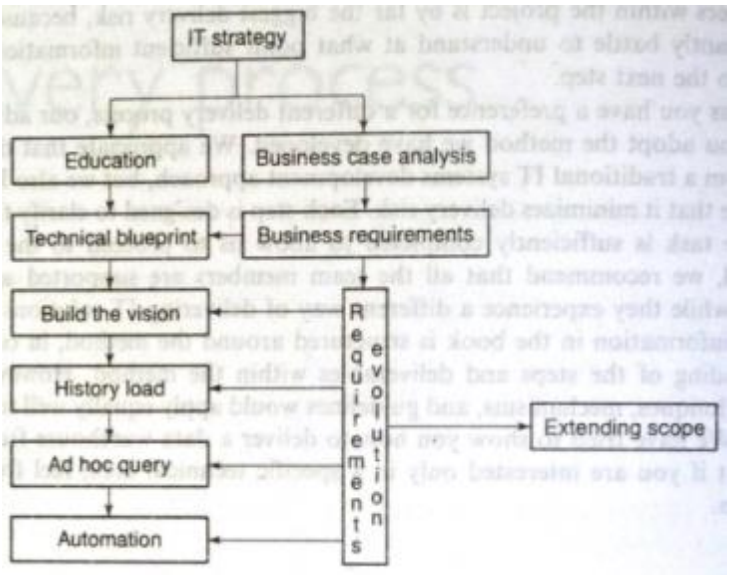
- have to **evolve and grow** as the business requirement for information changes over a period of time.
- never static, must be designed to change constantly, in a **performant** way
- Process has to be fundamentally different from a traditional waterfall method
- DW components cannot be completed until the requirements are completed, which can lead to constant requirements iteration without delivery, i.e, " **paralysis by analysis**."
- design and build around what we know today, and what we can guess about the future

Data Warehouse Delivery Method

- variant of the **joint application development** approach, adapted for delivery of data warehouses
- entire delivery process is **staged** in order to·minimize the risk, by producing production-quality deliverables
- development cycle of between 18 and 24 months from start to end
- the approach ensure that business benefits are delivered **incrementally** through the development process
- cost of investment can be offset and measured against the benefits being generated by the data warehouse
- designed to deliver an **enterprise data warehouse**, not a point solution.
- provide information about many aspects of information being used by the business.
- not small databases (that is, 50GB) or data marts being used to address a specific functional need
- provide an accurate and consistent view of enterprise information

The Data Warehouse Delivery Process (IE-TB-HAA-BBER)

- IT strategy
- Education And Prototyping
- Technical Blueprint
- Building The Vision
- History Load
- Ad Hoc Query
- Automation
- Business Case Analysis
- Business Requirements
- Extending Scope
- Requirements Evolution



IT strategy

- Data warehouses are strategic investments, which may require business process redesign in order to generate the projected benefits.
- If there is no overall IT strategy that includes a data warehouse, it can be difficult to procure and retain funding for the project

Business Case

- critical that we understand the business case for investment.
- necessary to understand the level of investment, regarding the scope and size of the solution
- purpose is to identify the projected business benefits that should be derived from using the data warehouse
  - may or may not be quantifiable (for example, a $5 000 000 saving per annum)
- Without a clear business or built as an act of faith tend to suffer from credibility problems

Education And Prototyping

- experiment with the concept of data analysis and educate themselves on the value of a data warehouse
- initial proto typing activity, used for understanding of the feasibility and benefits of a data warehouse

<u>Prototyping activity</u>

- addresses a clearly defined technical objective;-
- can be thrown away once the feasibility of the concept has been shown
  - does not become the first bit of the data warehouse
- addresses a small subset of the eventual data content of the data warehouse;
- timescale is non-critical for new technologies

Business Requirements

- By understanding the business requirements for both the short and medium term, we can design a solution that satisfies the short-term need, but is capable of growing to the full solution
- at least 20% of the time should be spent on understanding the likely longer-term requirement
- Within this stage, we must determine
  - the logical model for information within the data warehouse,
  - the source systems that provide this data (that is, mapping rules),
  - the business rules to be applied to data,
  - the query profiles for the immediate requirement.

Technical Blueprint

- must deliver an overall architecture that satisfies the longer-term requirements, and a definition of the components that must be implemented in the short term
- The blueprint must identify:
  - the overall system architecture,
  - the server and data mart architecture for both data and applications,
  - the essential components of the database design,
  - the data retention strategy,
  - the backup and recovery strategy,
  - the capacity plan for hardware and infrastructure (for example, LAN and WAN).

Building The Vision

- stage where the first production deliverable is produced
- smallest component of the data warehouse that adds business benefit
- the purpose of minimizing the scope of this phase is to reduce project risk, and to timebox the deliverable into a 4- 6-month exercise
- example
  - build the major infrastructure components for extracting and loading data, but limit them to one or two data sources, with minimal history

History Load

- remainder of the required history is loaded into the data warehouse
- new entities would not be added
- Example:

- After the building the vision phase, backload two years' worth of sales history from archive tape
- 50GB database could easily expand to become 400GB, may need to focus on resolving the very large database (VLDB) issues

Ad Hoc Query

- end-user access tools that are capable of automatically generating the database query that answers any question posed by the user
- Example
  - sales by store last week
  - this is converted into the database query by the access tool, which is aware of the structure of information within the data warehouse

Automation

- many of the following operational management processes are fully automated
  - backing up, restoring and archiving data;
  - generating aggregations from predefined definitions within the data warehouse;
  - monitoring query profiles, and determining the aggregations to maintain system ·performance

Extending Scope

- the scope of the data warehouse is extended to address·a new set of business requirements
  - this involves the loading of additional data sources into the data warehouse
  - introduction of new data marts using the existing information
- effort and complexity can be substantial, requiring this to be a separate phase

Requirements Evolution

- requirements.are never static
- Business requirements will constantly change
- process has to supports this, and allows these changes to be reflected within the.system
- architecture is designed to change and grow to match changing business needs
- new user requirements are continually fed into the development activities.
- Partial deliverables are produced, fed back to users, and then reworked, ensuring that the overall system is continually updated to meet the business needs

System Process

- Overview
- Typical Process flow within a data warehouse
- Extract and load process
- Clean and transform data
- Backup and archive process
- Query management process

Overview

- Data warehouses must be architected to support three major driving factors:
  - populating the warehouse,
  - day-to-day management of the warehouse,
  - the ability to cope with requirements evolution
- difficult challenge is to architect systems that automate requirements that are continuing to change
- The processes required to populate the warehouse focus on extracting the data, cleaning it up, and making it available for analysis.
- Requirements evolution tends to be the most complex aspect of a data warehouse

Example

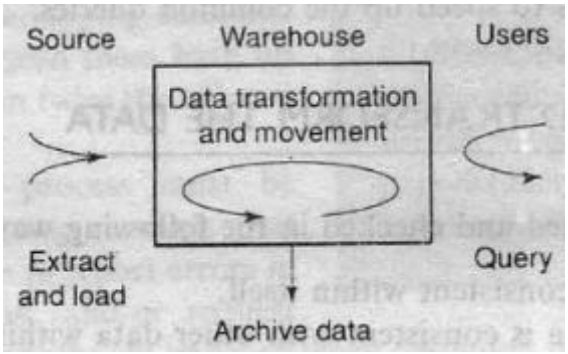
- sales analysis data warehouse
- first production phase requires an initial 50GB of data
- likely to grow to 1 TB of data over the following 2 years
- the underlying hardware must be capable of supporting the eventual database size

Typical Process flow within a data warehouse

The processes are:

1. Extract and load the data.
2. Clean and transform data into a form that can cope with large data volumes, and provide good query performance.
3. Back up and archive data.
4. Manage queries, and direct .them to the appropriate data sources.

Process flow within a data warehouse



Extract and load process

- Data extraction takes data from source systems and makes it available to the data warehouse;
- data load takes extracted data and loads it into the data warehouse
- This process must take data and add context and meaning in order to convert it into value-adding business information

This is achieved by

- extracting the data from the source systems,
- loading it into the database,
- stripping out any detail that is there to support the operational system
- adding more context (that is, more reference data),
- then reconciling the data with the other sources

Controlling The Process

to ensure that the various tools, logic modules, and programs are executed in the correct sequence and at the correct time, a controlling mechanism is required

Example:

it may be inappropriate to start the process that extracts EPOS transactions for a retail sales analysis data warehouse until all EPOS transactions have been received from all stores, and have been audited.

When To Initiate The Extract

- Data should be in a consistent state when it is extracted from the source system
- should be extracted only at a point where it represents the same instance of time as the extracts from the other data sources

Snapshot

- the information in a data warehouse represents a snapshot of corporate information
- so that the user is looking at a single, consistent, version of the truth

Example

- customer profiling data warehouse in the telecommunications sector – illogical to merge the list of customers at 7 pm on Friday customer database with the customer subscription events up to 7 pm on Thursday from a customer events database

Loading The Data

- Once the data is extracted from the source systems, it is then typically loaded into a temporary data store in order for it to be cleaned up and made consistent.
- The data warehouse probably is the first time that consistency issues between the two separate systems become apparent. Hence, checks are important
- the load process must be capable of fully automatic running
- should have the intelligence to report errors in the load and move on,
- ensure that the error recovery is an integral part of the design
- more complex in situations where the data warehouse has to be available 24 hours a day, 7 days per week

Copy Management Tools And Data Cleanup

- Most copy management tools implement these checks by allowing users to code the logic in either SQL, stored procedures, or their own programming language
  - substantial amount of development time
- If the source systems do not overlap much, and the consistency checks are simplistic, a copy management tool will cut down the coding effort required

Clean and transform data

This is the system process that takes the loaded data and structures it for query performance, and for minimizing operational costs.

Steps within the process:

1. Clean and transform the loaded data into a structure that speeds up queries.
2. Partition the data in order to speed up queries, optimize hardware performance and simplify the management of the data warehouse.
3. Create aggregations to speed up the common queries

Clean And Transform The Data

Data needs to be cleaned and checked in the following ways

Make sure data is consistent within itself

- examining a row of data and examine it, the contents of the row must make sense
- may be due to errors in the source systems
- Typical checks are for nonsensical phone numbers, addresses, counts, and so on.

Make sure that data is consistent with other data within the same source

- examining the data against other tables within the same source, the data must make sense
- check for customer/service in the transaction by comparing it with the list of valid customers/services

Make sure data is consistent with data in the other source systems

- examining a record and compare it with a similar record in a different source system
- reconciling a customer record with a copy in·a customer database & a copy in a customer events database.

Make sure data is consistent with the information already in the warehouse

- ensure that any data being loaded does not contradict the information already within the data warehouse

Transforming Into Effective Structures

- convert the source data in the temporary data store into a structure that is designed to balance query performance and operational cost
- Information in the data warehouse must be structured to support the performance requirement from the business

Backup and archive process

- the data within the data warehouse is backed up regularly in order to ensure that the data warehouse can always be recovered from data loss, software failure or hardware failure
- In archiving, older data is removed from the system in a format that allow.it to be quickly restored if required
- It is common to archive data as a flat file extract

Query management process

- manages the queries and speeds them up by directing queries to the most effective data source
- ensure that all the system resources are used in the most effective way
  - by scheduling the execution of queries
- monitor the actual query profiles
  - useful to determine which aggregations to generate

Directing Queries

- Data warehouses that contain summary data potentially provide a number of distinct data sources to respond to a specific query.

Consider the following example, in a retail sales analysis data warehouse

- The following user query
  - Report on sales of baked beans across Marin County, California, over the past 2 weeks

could be handled by any of the following tables

- all the detailed information over the past 2 weeks, filtering in all baked beans transactions for Marin County;
- 2 weeks worth of weekly summary tables of product by store across the week
- a bi-weekly summary table of product by region
- a bi-weekly summary table of product group by store

However, the scan performance will vary quite substantially between each table

- The ·query management process determines which table delivers the answer most effectively; by calculating which table would satisfy the query in the shortest space of time

Maximizing System Resources

queries from hell

- scan the entirety of detailed information
- constructed inappropriately
- perform repetitive scanning of a large table

The query management process must ensure that no single query can affect the overall system performance

Query Capture

- query profiles change on a regular basis over the life of a data warehouse
- if the profile changes, the summary tables need to change as well
- In order to accurately monitor and understand what the new query profiles are, it can be very effective to. capture the physical queries that are being executed
- At various points in time, such as the end of the week, these queries can be analyzed to determine the new query profiles, and the resulting impact on summary tables.
- This analysis can be manual or automatic
- Query capture is typically part of the query management process

Process Architecture

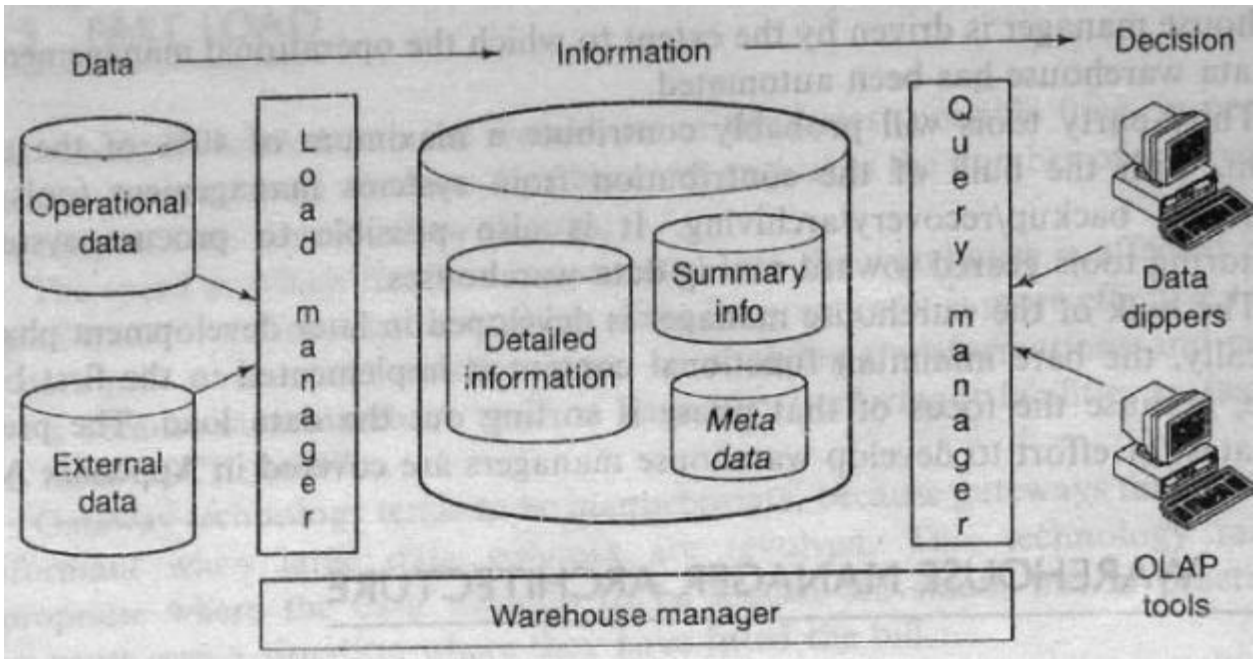
- Introduction
- Load Manager
  - Load Manager Architecture
  - Extract Data From Source
  - Fast Load
  - Simple Transformation
- Warehouse Manager
  - Warehouse Manager Architecture
  - Complex Transformations
  - Transform Into A Starflake Schema
  - Create Indexes And Views
  - Generate The Summaries
- Query Manager
  - Query Manager Architecture
- Detailed Information
  - Data Warehouse Schemas
  - Fact Data
  - Dimension Data
  - Partitioning Data
- Summary Information
- Meta Data
- Data Marting

Introduction

Mapping processes to systems

Process	Function	System manager
Extract and load	Extracts and loads the data, performing simple transformations before and during load	Load manager
Clean and transform data	Transforms and manages the data	Warehouse manager
Backup and archive	Backs up and archives the data warehouse	Warehouse manager
Query management	Directs and manages queries	Query manager

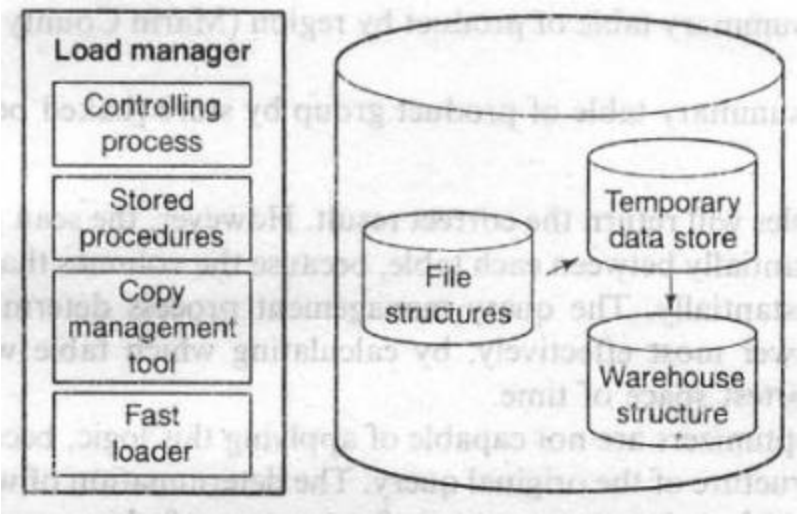
Architecture for a data warehouse



Load Manager

- system component that performs all the operations necessary to support the extract and load process
- constructed using a combination of off-the-•shelf tools, bespoke coding, C programs and shell scripts
- it performs the following operations:
  1. Extract the data from the source systems.
  2. Fast-load the extracted data into a temporary data store.
  3. Perform simple transformations into a structure similar to the one in the data warehouse
- Each of these functions has to operate automatically, and recover from any errors it encounters, to a very large extent with no human intervention
- this process tends to run overnight at the close of the business day

Load Manager Architecture



Extract Data From Source

- source data it has to be transferred from source systems, and made available to the data warehouse.
- ASCII files are FTPd across the LAN
- sufficient capacity in the LAN to allow this to happen efficiently
- Current gateway technology operates too slowly

Tools that implement load manager tasks

Task	Tool
Fast load	Database fast-load facility
Simple transformation	Copy management tool
Complex checking	Stored procedures or C /C ++
Job Control	UNIX shell scripts and source systems

Fast Load

- Data should be loaded in the fastest possible time, in order to minimize the total load window
- speed is affected by the kind of transformations
- more effective to load the data into a relational database prior to applying transformations and checks
- most database vendors have very performant facilities to fast-load data from ASCII flat files
- usually bypass the SQL layer of the database software

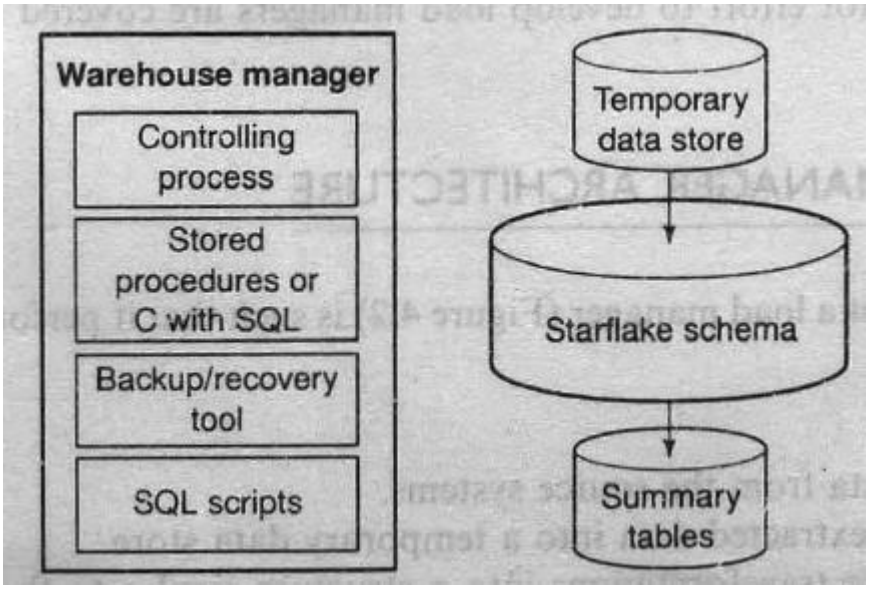
Simple Transformation

- Before or during the load there will be an opportunity to perform simple transformations on the data
- Example: loading EPOS sales transactions into a retail sales analysis data warehouse
  - Strip out all columns that are not required within the warehouse
  - Convert all values to the required data types

Warehouse Manager

- system component that performs all the operations necessary to support the warehouse mgmnt. process
- This system is typically constructed .using a combination of third--party systems management software, bespoke coding, C programs and shell scripts
- complexity of the warehouse manager is driven by the extent to which the operational management of the data warehouse has been automated

Warehouse Manager Architecture



Operations performed by warehouse manager

1. Analyze the data to perform consistency and **referential integrity** checks.
2. Transform and merge the source data in the temporary data store into the published data warehouse.
3. Create indexes, business views, partition views, business synonyms against the base data.
4. Generate denormalizations if appropriate.
5. Generate any new aggregations that may be required.
6. Update all existing aggregations.
7. Back up incrementally or totally the data within the data warehouse.
8. Archive data that has reached the end of its capture life

Complex Transformations

- ensure that the basis of comparison is the same: ie the reference information is identical
- we start by reconciling the key reference items, and then work our way outwards to related reference items; before addressing the base transaction

Transform Into A Starflake Schema

- transform it into a form suitable for decision support queries
- the bulk of the factual data lies in the center, surrounded by the reference (dimension) data.
- Three variations on this theme are commonly used:
  - star schemas,
  - snowflake schemas
  - starflake schemas

Create Indexes And Views

- most relational technologies have facilities to create indexes in parallel, distributing the load across the hardware and significantly reducing the elapsed time
- overhead of inserting a row into a table & indexes can be far higher than that of re-creating indexes
- more effective to drop all indexes against tables
- Fact data - a prime candidate for dropping indexes before the data load
- retain existing indexes on the dimension tables

- the warehouse manager creates views that combine a number of partitions into a single fact table
- user access tools made to think that the fact table partitions are one large table

Example:if we have adopted a monthly partitioning strategy for a Sales Fact table, we could consider creating:

- a view across the last quarter;
- a view across the last six months;
- views corresponding to .financial quarters for the business

Generate The Summaries

- the warehouse manager has to create a set of aggregations to speed up query performance
- operate on the basis that most queries select a subset of data from a particular dimension
- Summaries are generated automatically
- achieved through the use of embedded SQL in either stored procedures or C programs
  - create table { ... } as select {.. . . } from { . .. } where { . .. }
- can create the table in parallel - making full use of system resources - reducing the total elapsed time

Using metadata

- Metadata can be used to address the issue of changing summaries as per query profiles
  - by data-driving the generation of summaries

Using query statistics

- maintaining the responsiveness of the system even when the query profiles change over time
- can be achieved by gathering statistics on the queries being physically executed

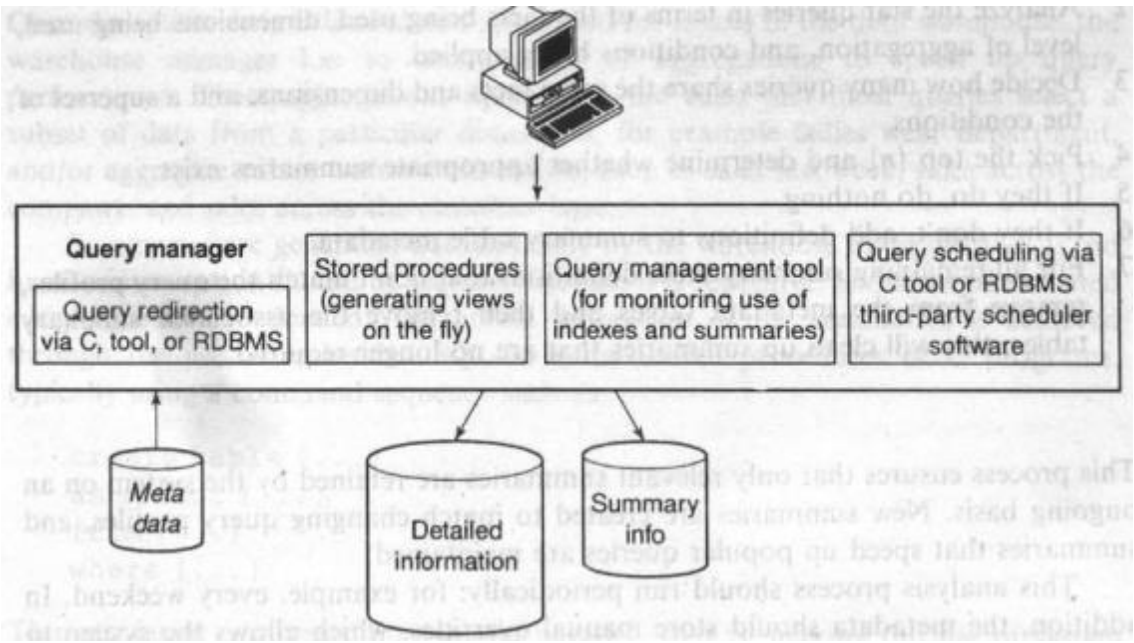
With the query statistics, the warehouse manager can be extended to:

1. Convert the SQL queries into star queries
2. Analyze the star queries
3. Decide how many queries share the same facts and dimensions
4. Pick .the top {n} and determine whether appropriate summaries exist
5. If they do, do nothing
6. If they don' t, add definitions to summary table metadata
7. For all remaining summary table definitions, remove from the metadata tables

Query Manager

- system component that performs all the operations necessary to support the query management process
- constructed using a combination of user access tools, specialist data warehousing monitoring tools, native database facilities, bespoke coding, C programs, and Shell scripts

Query Manager Architecture



It performs the following operations:

1. Direct queries to the appropriate table(s).
2. Schedule the execution of user queries.

Tools that implement query manager tasks

Task	Tool
Direct queries to the appropriate table	User access tools, stored procedures or C/C++
Schedule. the execution of user queries	Stored procedures or C/C++, native database facilities, user access tools, or third-party scheduler software

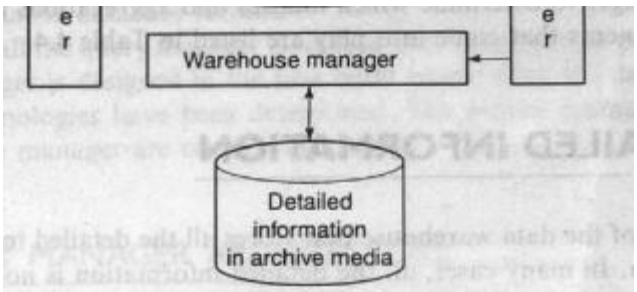
Detailed Information

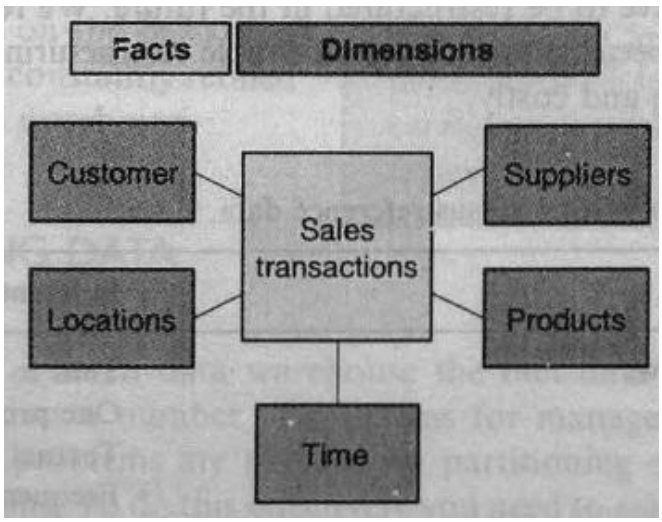
- This is the area of the data warehouse that stores all the detailed information in the starflake schema
- In many cases, all the detailed information is not held online the whole time
- aggregated to the next level of detail, and the detailed information is then offloaded into tape archive.
- On a rolling basis, detailed information is loaded into the warehouse to supplement the aggregated data
- to implement a rolling strategy, historical data will have to be loaded at regular intervals

Example

- very common to store transactions at the level of product by store by day for retail sales analysis DWH
- appropriate for analyzing actual sales against projected sales
- this degree of detail does not provide **basket analysis**
- it is not possible to examine a specific customer basket transaction., or analyze components within it

A data warehouse with detailed data in archived storage





Data Warehouse Schemas

- Star schemas are database schemas that structure the data to exploit a typical decision support query
- most queries tend to share similarities
  - queries examine a set of factual transactions
  - Queries analyze facts in a variety of ways
- the information to support these queries can be described in a way that places the large volume of factual transactions in the center, and the way in which it is analyzed surrounds it

A retail sales analysis star schema

Refer picture

Fact Table

- The central factual transaction table is called the fact table

Dimension Tables

- The surrounding reference tables are called the dimension tables

Star Schema

- Putting them together representing a star: hence the name

Fact Data

Within a data warehouse, information is split into two distinct classes:

- the **basic factual information event**
  - contains the physical information that describe.a factual event that occurred within the business
- the **reference information** that is used to analyze the factual event
- Fact data is the major database component of a typical data warehouse, constituting around 70% of the total database volume
- Detailed aggregations are treated as fact data

Examples:

- EPOS transactions, phone calls, and banking transactions against accounts

Characteristics of fact versus reference data

Fact	Reference
Millions or billions of rows	Tens to a few million rows
Multiple foreign keys	One primary key
Numeric	Textual descriptions
Don't change	Frequently modified

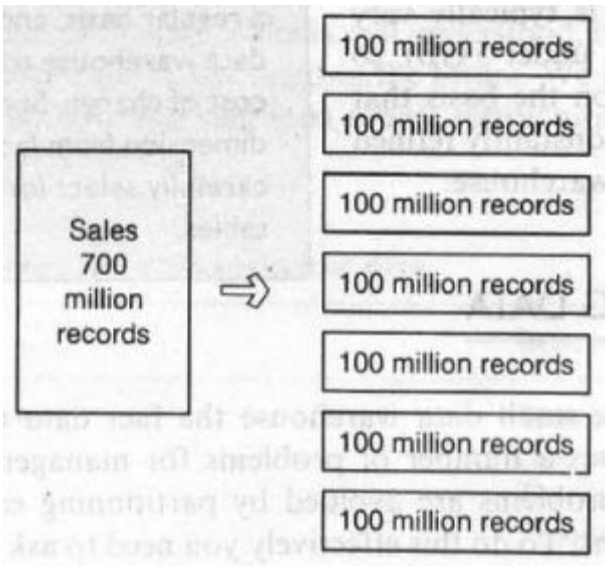
Dimension Data

- Dimension data is the information that is used to analyze the elemental transaction
- The structuring of information in this way makes it possible to optimize query performance
- dimension data will change over time
- structure the dimensions with a view allowing rapid change

Examples:a product hierarchy, customers

Partitioning Data

- fact data tables can become extremely large
- number of problems for management, backup etc
- These problems are avoided by partitioning each fact table into multiple separate partitions
- partitioning the large-volume data in order to minimize the set of data that needs to be scanned
- a partitioning strategy is created to favor queries that scan only recent history, such as week to date and month to date



An example of a partitioned retail sales fact table

Reasons for partitioning

- Partitioning for ease of management
  - can grow to many hundreds of gigabytes or even beyond a terabyte in size
  - add a column to a single table of a terabyte in size
  - impossible to drop and re-create the index for a data load
- Partitioning to assist backup/recovery
  - Backing up or restoring a table containing all online fact data
  - cut down on the amount that has to be backed up on a regular basis
  - all partitions other than the current partition can be marked read-only
- Partitioning for performance
  - removing large parts of the fact table from the possible set of data that needs to be scanned.
  - achieved by breaking up the fact table into partitions
- Hardware partitioning
  - Data warehouses delivered on shared-nothing architectures work best if the data is partitioned to optimize the accessibility of data by each processing node

Summary Information

- area of the DW that stores all the predefined aggregations generated by the warehouse manager
- This area of the data warehouse should be treated as transient.
- It will change on an ongoing basis in order to respond to changing query profiles
- Summary information is essentially a replication of detailed information already in the data warehouse.

Summary data implications

- exists to speed up the performance of common queries;
- increases operational costs;
- may have to be updated every time new data is loaded into the data warehouse;
- may not have to be backed up, because it can be generated fresh from the detailed information

Advantage

- Optimal query performance is typically achieved through a high level of aggregations of the base data
  - bulk of the processing for a query is completed in advance

Disadvantage

- increase in the operational management cost, for creating and updating the summary table on a daily basis

Users own aggregations

- users created their own aggregations for very specific reasons
- example, a retail merchant may have created a sales aggregation for a set of stores that he or she uses
- These types of aggregation should not be prevented
- controlled by providing a reasonable limit to the amount of personal database space given to :each user

Meta Data

- area within the data warehouse that stores all the metadata definitions used by all processes within the data warehouse.
- Metadata is data about data:
- it is like a card index describing how information is structured within the data warehouse
- The structure of metadata will differ between each process
- copy management tools
  - use metadata to understand the mapping rules that they need to apply in order to convert source data into a common form
- User access tools
  - use metadata to understand how to build up a query

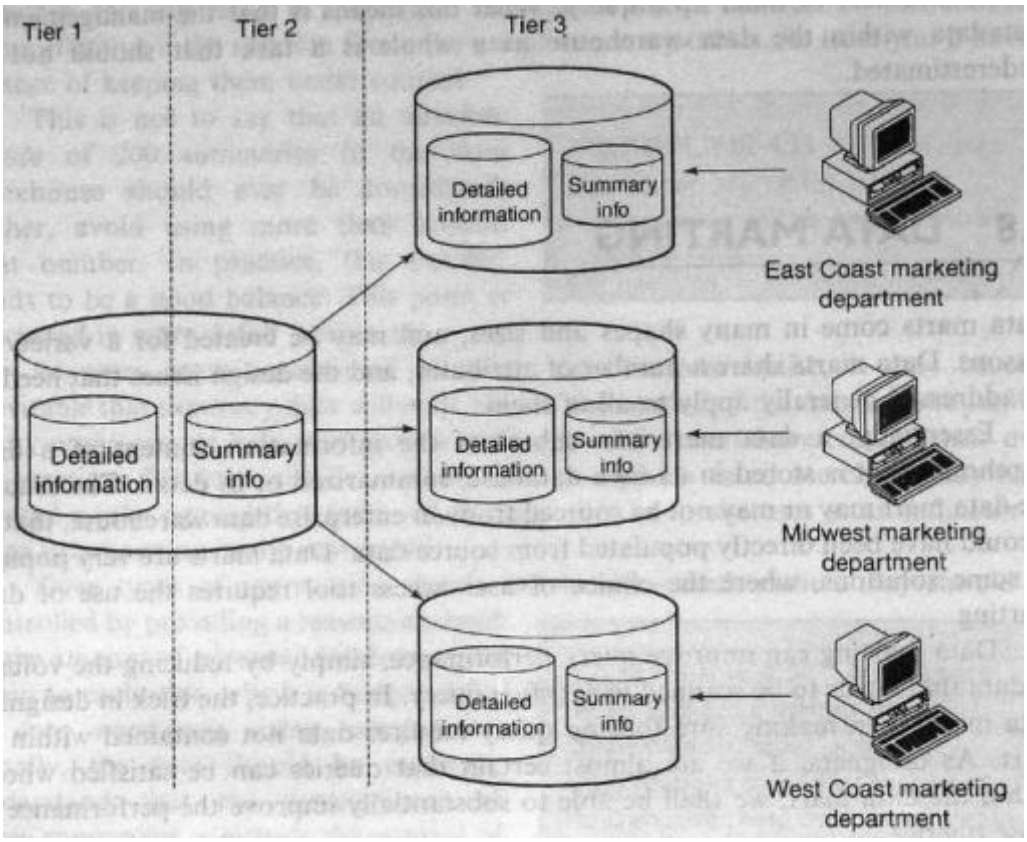
Used for a variety of purposes

- map data sources to the common view of information within the data warehouse
- automate the production of summary tables
- direct a query to the most appropriate data source

Data Marting

- data mart is a subset of the information content of a data warehouse that is stored in its own database, summarized or in detail.
- could have been directly populated from source data
- the choice of user access tool requires the use of data marting
- Data marting can improve query performance, simply by reducing the volume of data that needs to be scanned to satisfy a query
- making sure that no query requires data not contained within its mart
- Typically created along functional or departmental lines, in order to exploit a natural break of the data

Three tiers of decision support information

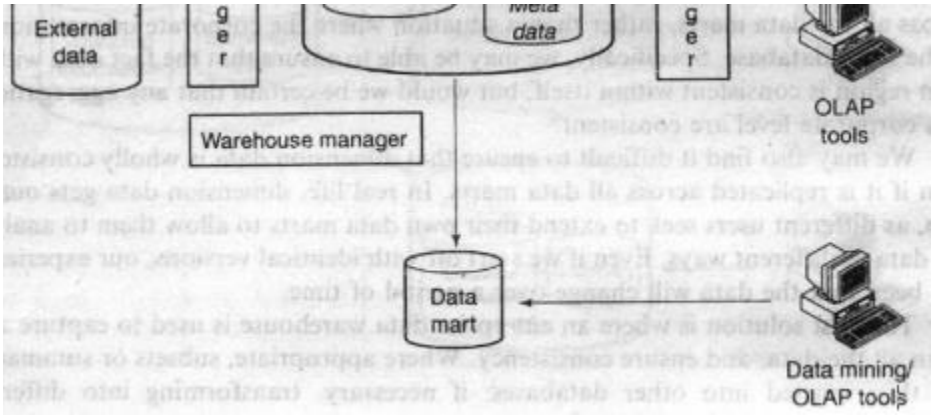


A number of user access tools create their own data mart designed to support their specific functionality·

Can We Do Without An Enterprise Data Warehouse

- No. it is very difficult to make data consistent with information that may not be present in the same database
- The best solution is where an enterprise data warehouse is used to capture and clean all the data, and ensure consistency
- design for data retention within the data warehouse

A data marting strategy for user access tools



Disclaimer

- Intended for educational purposes only. Not intended for any sort of commercial use
- Purely created to help students with limited preparation time. No guarantee on the adequacy or accuracy
- Text and picture used were taken from the reference items

Reference

Sam Anabory & Dennis Murray , "Data Warehousing in the real world", Addison Wesley, 1997
http://books.google.co.in/books?id=f_C30AM0NkIC&printsec=frontcover#v=onepage&q&f=false

Credits

Thanks to my family members who supported me, while I spent considerable amount of time to prepare these notes.
Feedback is always welcome at GHCRajan@gmail.com