Data Warehousing Concepts

Adapted from Chapter 31 Database Systems:
A Practical Approach to Design :Implementation and Management by Connolly Begg 2015 © Pearson Education

Oracle Database Data Warehousing Guide 11*g Release 2*:
Chapter 8 Basic Materialized Views

The Evolution of Data Warehousing

- Since 1970s, organisations focus on OLTP
  - To gain competitive advantage through automating business processes to offer more efficient and cost-effective services to the customer.
- This resulted in accumulation of growing amounts of data in operational databases (typically RDBMS)
- Data Warehousing Focus is on ways to use operational data to support decision-making, as a means of gaining competitive advantage.

2

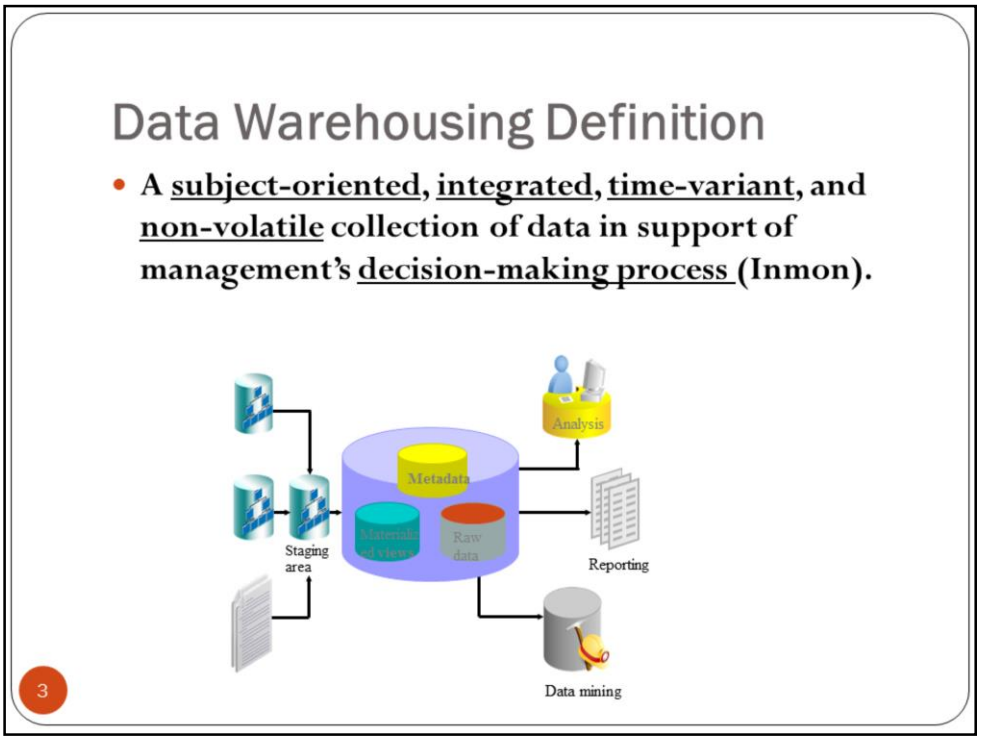
BUT operational systems were never designed to support such business activities.

Businesses typically have numerous OLTP systems with overlapping sometimes contradictory definitions.

Need to turn their archives of data into a source of knowledge, so that a single integrated / consolidated view of the organization's data is presented to the user.

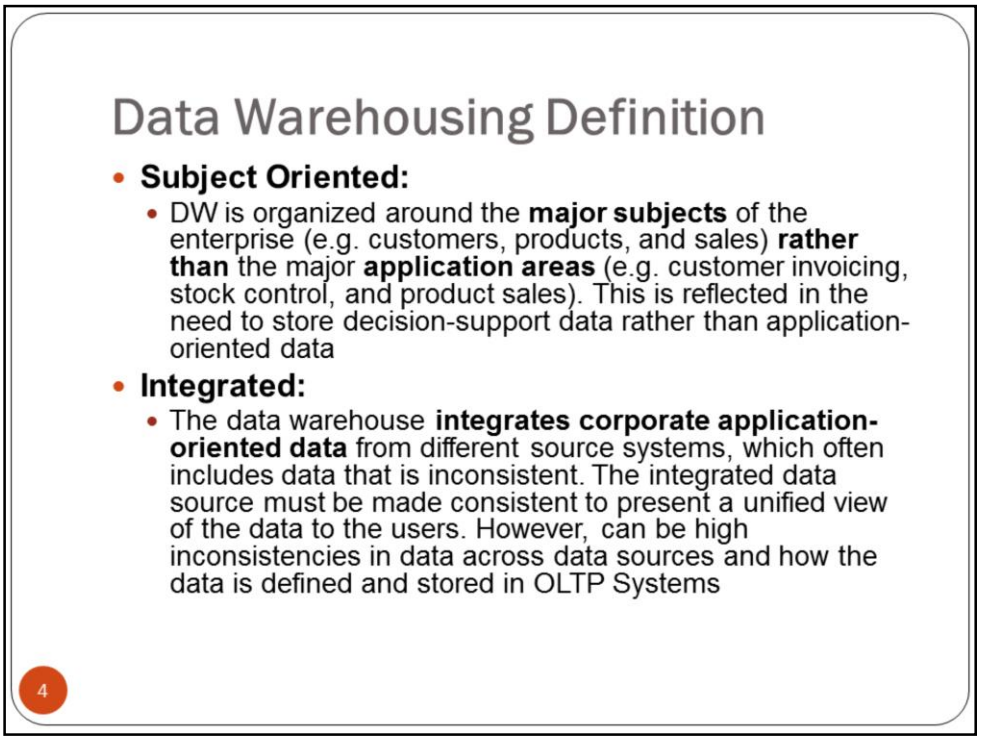
A data warehouse was deemed the solution to meet the requirements of a system capable of supporting decision-making.

A data warehouse to meet the requirements of a system capable of supporting decision-making, receiving data from multiple operational data sources.



Characteristics of a Data Warehouse

A data warehouse is a database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but can include data from other sources. Data warehouses separate analysis workload from transaction workload and enable an organization to consolidate data from several sources. The primary role that the data warehouse plays within a business entity is as an analytical tool. In addition to a database, a data warehouse environment includes an extraction (and transportation), transformation, and loading (ETL) solution, online analytical processing (OLAP) and data mining capabilities, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.



Inmon's Data warehouse Definition

Subject Oriented: The warehouse is organized around the major subjects of the enterprise (e.g. customers, products, and sales) rather than the major application areas (e.g. customer invoicing, stock control, and product sales). This is reflected in the need to store decision-support data rather than application-oriented data.

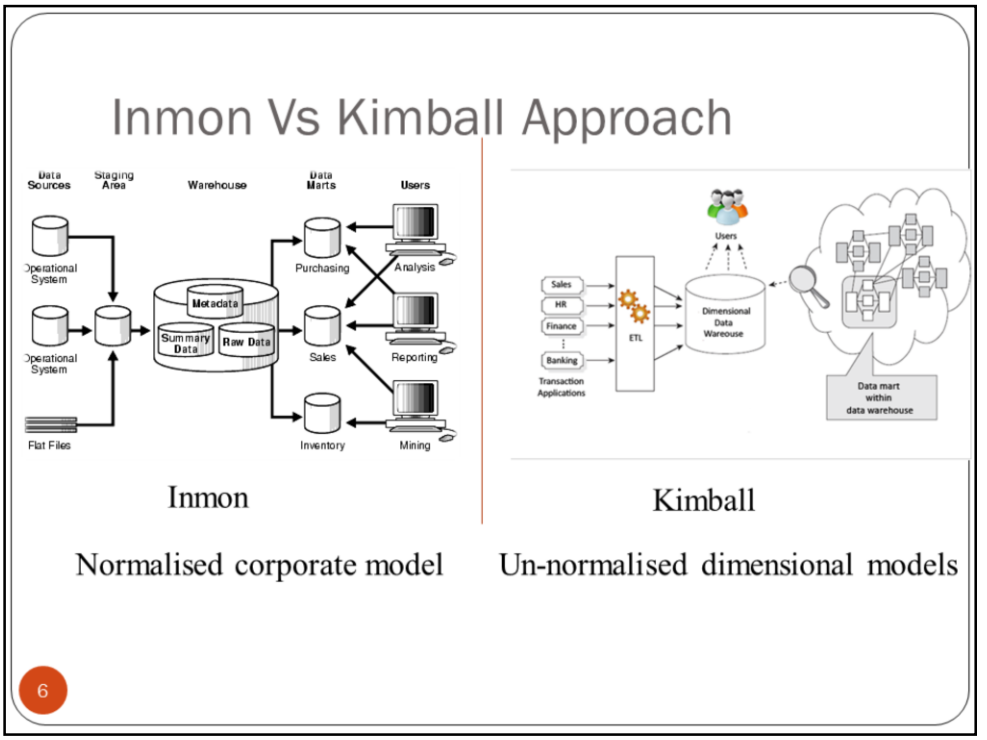
Integrated: The data warehouse integrates corporate application-oriented data from different source systems, which often includes data that is inconsistent. The integrated data source must be made consistent to present a unified view of the data to the users. However, can be high inconsistencies in data across data sources and how the data is defined and stored in OLTP Systems

Time-Variant: Data in the warehouse is only accurate and valid at some point in time or over some time interval. Time-variance is also shown in the extended time that the data is held, the implicit or explicit association of time with all data, and the fact that the data represents a series of snapshots.

Non-Volatile Data: Data in the warehouse is not updated in real-time but is refreshed from operational systems on a regular basis.New data is always added as a supplement to the database, rather than a replacement.

Data Warehousing Definition

- **Time-Variant:**
  - **Data** in the warehouse is only **accurate and valid at some point in time** or over some time interval. Time-variance is also shown in the extended time that the data is held, the **implicit or explicit association of time** with all data, and the fact that the data represents a series of snapshots.
- **Non-Volatile Data:**
  - Data in the warehouse is **not updated in real-time but is refreshed from operational systems on a regular basis**. **New data is always added as a supplement** to the database, rather than a replacement.

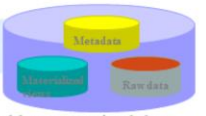


The **Inmon Approach** to building a data warehouse begins with the corporate data model. This model identifies the key subject areas, and most importantly, the key entities the business operates with and cares about, like customer, product, vendor, etc. The logical and physical implementation of the data warehouse is **normalized**. **This is what Inmon calls as a 'data warehouse**,' and here is where the **single version of truth for the enterprise** is managed. This normalized model **makes loading the data less complex**, but using this structure for **querying is hard as it involves many tables and joins**. The data marts will be designed specifically for Finance, Sales, etc., and the **data marts can have de-normalized data to help with reporting** (Breslin, 2004). Any data that comes into the data warehouse is integrated, and the data warehouse is the only source of data for the different data marts. This ensures that the integrity and consist

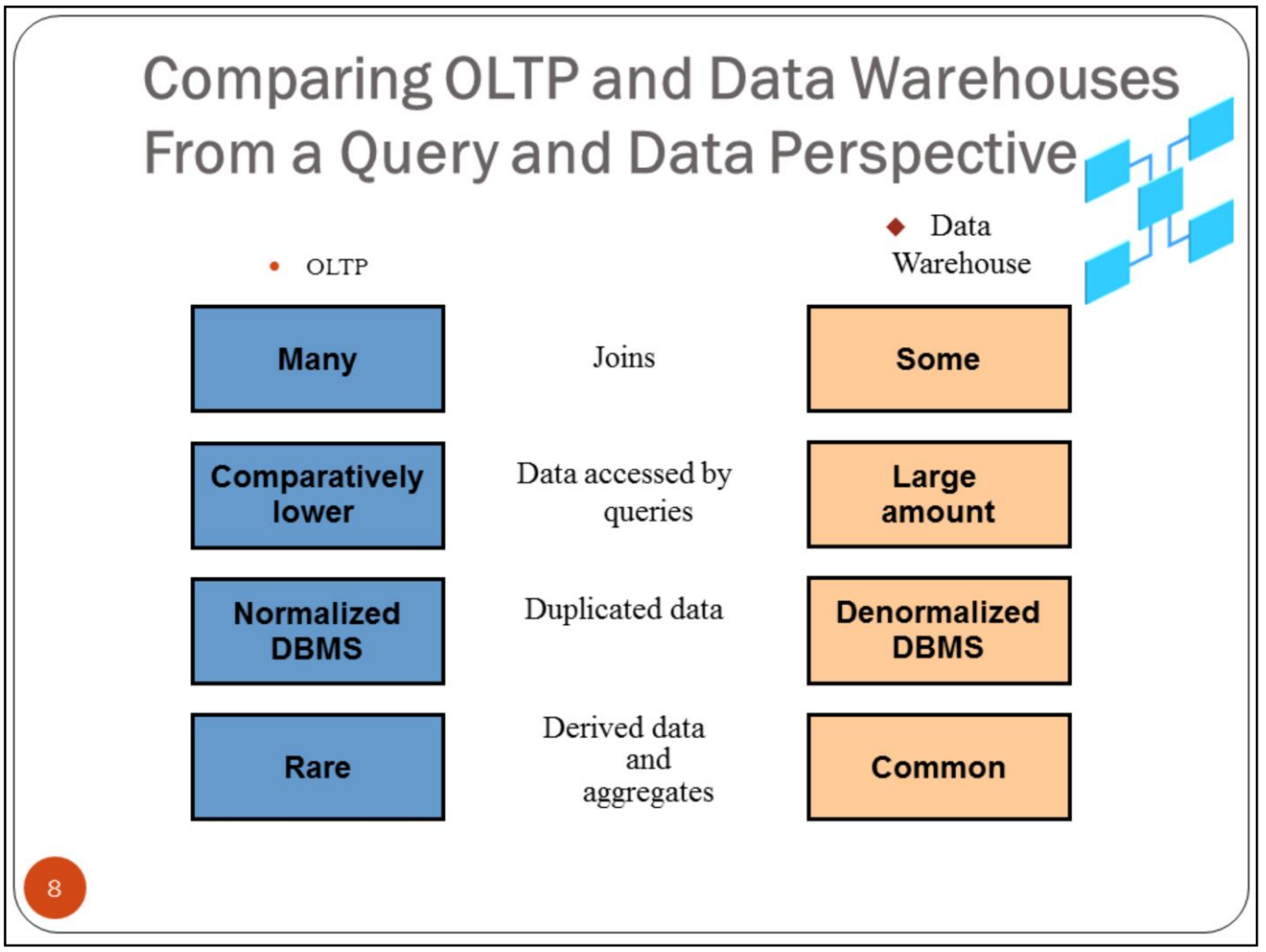
Kimball argues that data is loaded into a dimensional model. Here the comes the key difference: the model proposed by Kimball for data warehousing—**the dimensional model—is not normalized.** The fundamental concept of dimensional modeling is the star schema. In the star schema, there is typically a fact table surrounded by many dimensions. The fact table has all the measures that are relevant to the subject area, and it also has the foreign

keys from the different dimensions that surround the fact. The dimensions are denormalized completely so that the user can drill up and drill down without joining to another table. Multiple star schemas will be built to satisfy different reporting requirements. So, how is integration achieved in the dimensional model? Here, Kimball proposes the concept of ‘conformed dimensions’.ency of data is kept intact across the organization. Figure 1.2 shows the typical architecture of an Inmon data warehouse.

Comparison of OLTP Systems and Data Warehousing



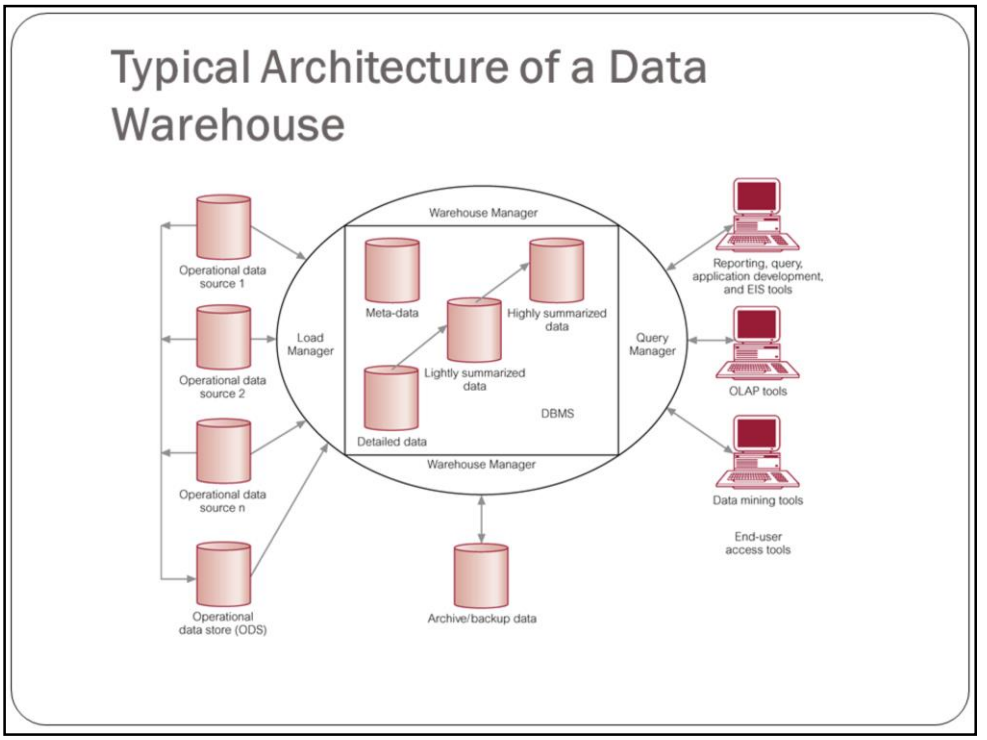
OLTP systems	Data warehousing systems
Holds current data	Holds historical data
Stores detailed data	Stores detailed, lightly, and highly summarized data
Data is dynamic	Data is largely static
Repetitive processing	*Ad hoc*, unstructured, and heuristic processing
High level of transaction throughput	Medium to low level of transaction throughput
Predictable pattern of usage	Unpredictable pattern of usage
Transaction-driven	Analysis driven
Application-oriented	Subject-oriented
Supports day-to-day decisions	Supports strategic decisions
Serves large number of clerical/operational users	Serves relatively low number of managerial users



Comparing OLTP and Data Warehouses

Data warehouses and OLTP systems have very different requirements. Data warehouses are designed to accommodate ad hoc queries. You may not know the workload of your data warehouse in advance. Therefore, a data warehouse should be optimized to perform well for a wide variety of possible query operations. OLTP systems generally support predefined operations. Your applications may be specifically tuned or designed to support only these operations.

A data warehouse is updated on a regular basis by the ETL (extraction, transformation, loading) process; a carefully defined and controlled bulk data loading system. The end users of a data warehouse do not directly update the data warehouse. In OLTP systems, end users routinely issue individual data modification statements to the database. The OLTP database is always up-to-date, and reflects the current state of each business transaction. Data warehouses often use denormalized or partially denormalized schemas (such as a star schema) to optimize query performance. OLTP systems often use fully normalized schemas to optimize update/insert/delete performance, and to guarantee data consistency. Note that the differences outlined here are somewhat generalized and should not be considered firm and unyielding distinctions.



1.Operational Data Sources Containing common OLTP such as order processing, stock management CRM system ERP and so on. Also, external systems to the organisation may be involved Credit rating systems

These sources may reside on different architectures like mainframe first generation hierarchical and network databases, departmental proprietary file systems (e.g. VSAM, RMS) and relational DBMSs (e.g. Informix, Oracle).Private workstations and servers can also be a source

External systems such as the internet, commercially available databases, or databases associated with an organization's suppliers or customers may also be used.

2. Operational Data Store (ODS)

This is a repository of current and integrated operational data used for analysis.

Often structured and supplied with data in the same way as the data warehouse.

May act simply as a staging area for data to be moved into the warehouse.

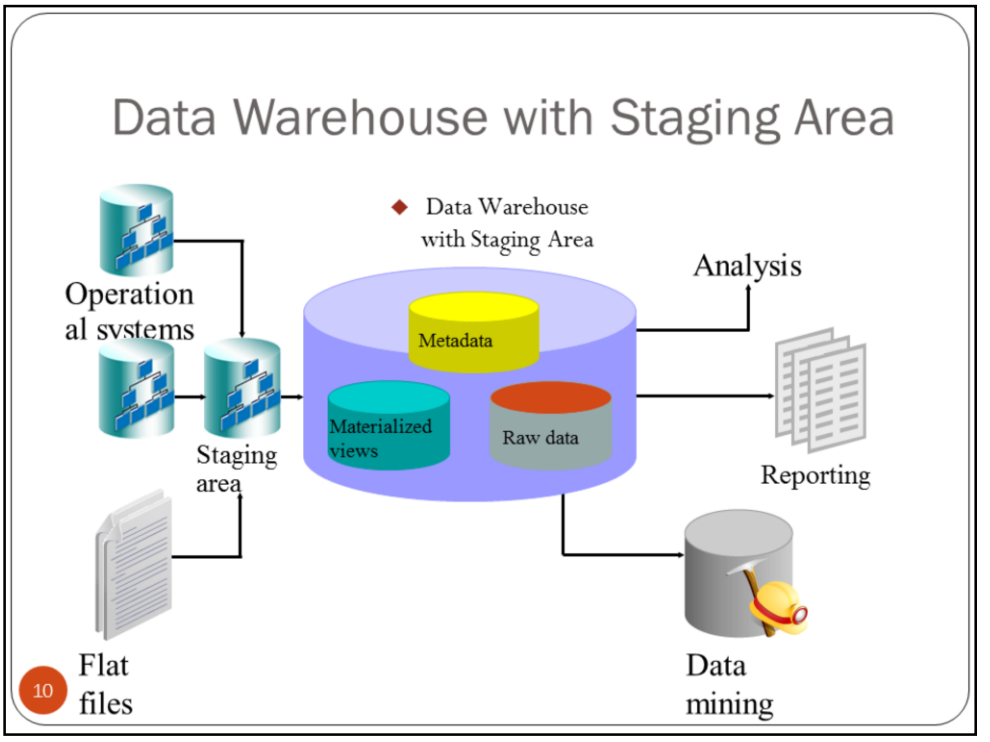
Often created when legacy operational systems are found to be incapable of achieving reporting requirements.

Provides users with the ease-of-use of a relational database while remaining distant from the decision support functions of the data warehouse.

3.Load Manager

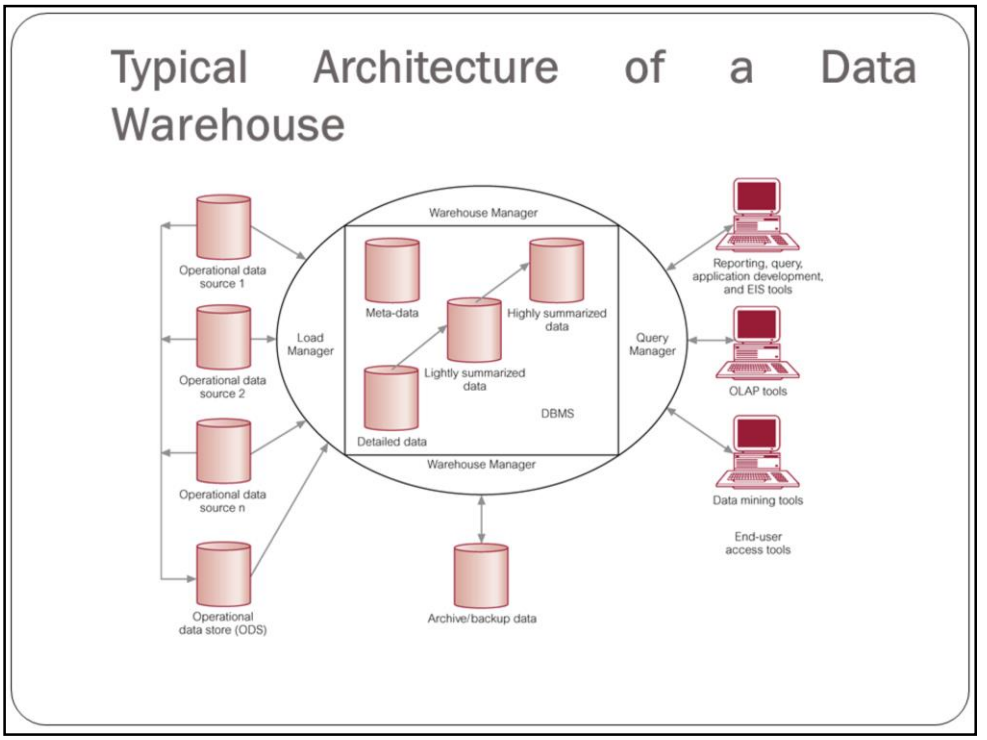
Performs all the operations associated with the extraction and loading of data into the warehouse.

Size and complexity will vary between data warehouses and may be constructed using a combination of vendor data loading tools and custom-built programs.



Data Warehouse with Staging Area

In the basic data warehouse illustrated on the previous page, you must clean and process your operational data before putting it into the warehouse. This task can be done programmatically, although most data warehouses use a staging area instead. **A staging area simplifies building summaries or materialized views and general warehouse management**. Staging areas are often useful for **performing intensive calculations** or **data-cleansing operations** that **may adversely affect the production query environment**. The data warehouse architecture shown in the slide illustrates this typical architecture.



4.Warehouse Manager

Performs all the operations associated with the management of the data in the warehouse. Constructed using vendor data management tools and custom-built programs. In some cases, also generates query profiles to determine which indexes and aggregations are appropriate. A query profile can be generated for each user, group of users, or the data warehouse and is based on information that describes the characteristics of the queries such as frequency, target table(s), and size of results set.

Operations performed include

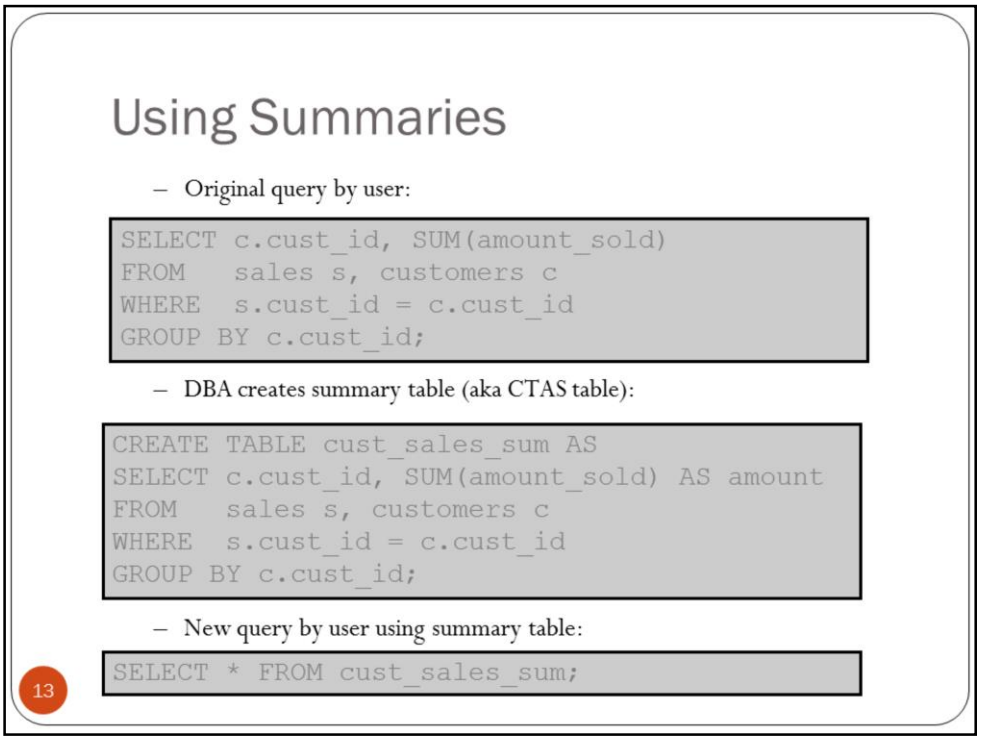
- Analysis of data to ensure consistency.
- Transformation and merging of source data from temporary storage into data warehouse tables.
- Creation of indexes and views on base tables.
- Generation of denormalizations, (if necessary).
- Generation of aggregations with summary tables or materialized views, (if necessary).
- Backing-up and archiving data.

5. Query Manager

Performs all the operations associated with the management of user queries. Typically constructed using vendor end-user data access tools, data warehouse monitoring tools, database facilities, and custom-built programs.Complexity determined by the facilities provided by the end-user access tools and the database.

Key Tasks (ETL)

- Extraction of data from each information source
- Transformation
  - Translation of the data
    - Resolving any incompatibilities between the same data from different information sources so that the data can be integrated
  - Cleaning of the Data
    - Deals with detection and removal of errors and inconsistencies in the data from each information source
- Load
  - Integration of the Data
    - Combines the extracted , translated and cleaned data from all the information sources



Using Summaries

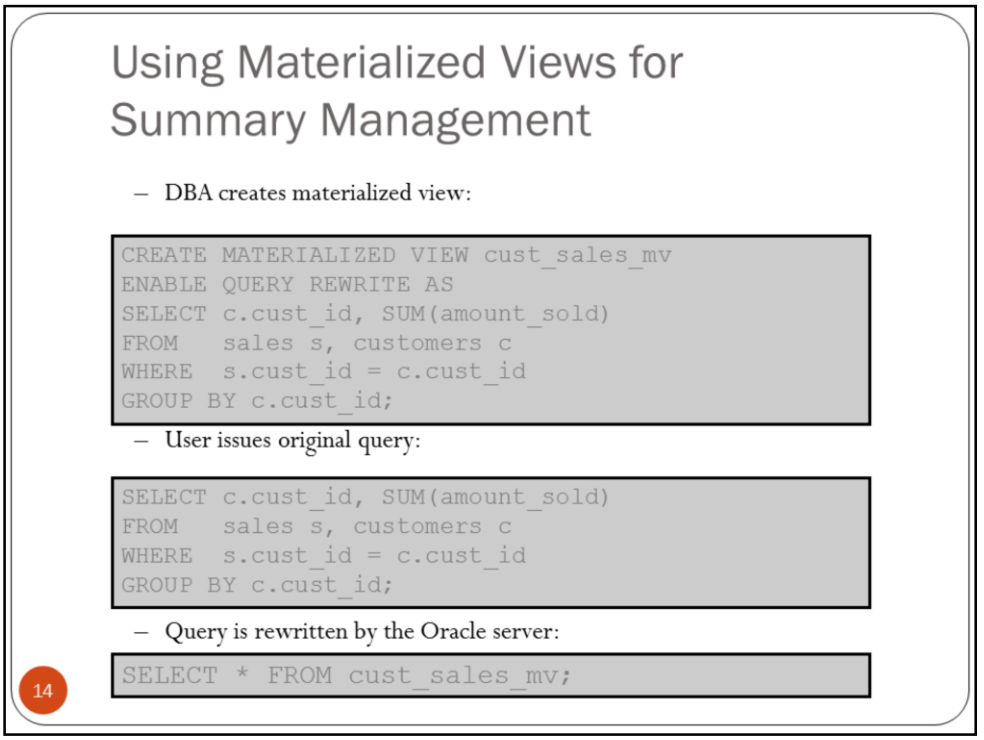
Before the introduction of materialized views in Oracle Database, organizations using summaries spent a significant amount of time creating summaries manually. They had to identify which summaries to create, index the summaries, update them, and advise their users on which ones to use.

In the example in the slide, the DBA creates a summary table called `CUST_SALES_SUM`, to improve the performance of the initial query shown. The DBA informs the users of its existence, and users then query the summary table rather than executing the original query.

The time required to execute the SQL query by using the summary table is minimal compared to the original SQL query.

However, users must be made aware of summary tables, and they need to rewrite their applications to use them.

In addition, the DBA must manually refresh the summary tables to keep them up-to-date with the corresponding original tables.

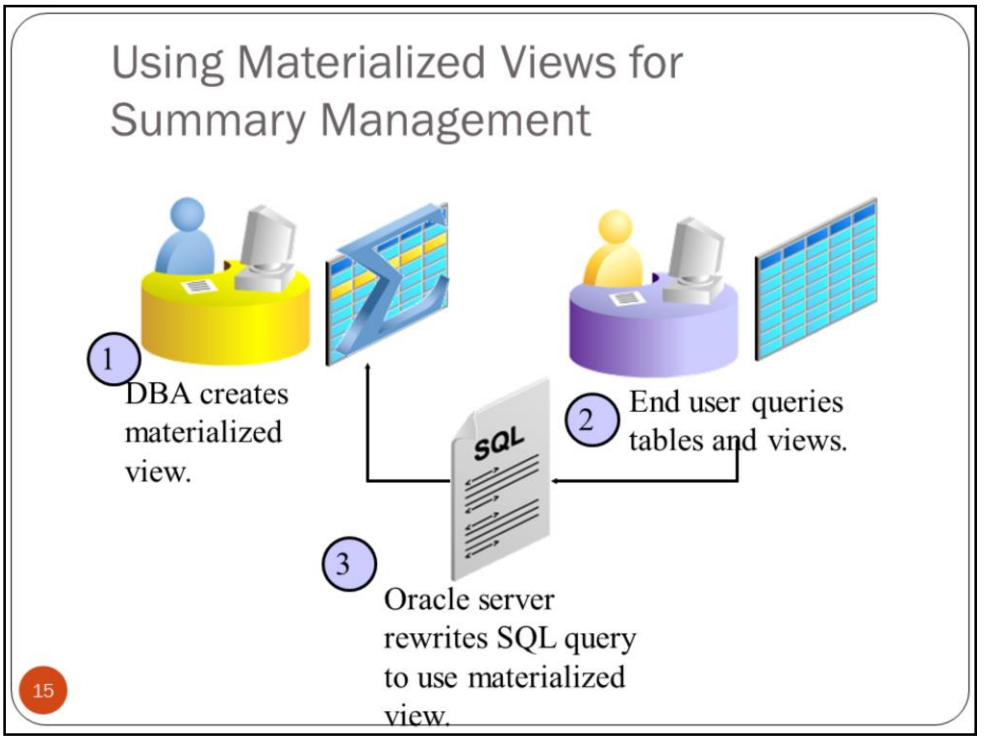


Using Materialized Views for Summary Management

Summary management in Oracle Database eases the workload of the database administrator and eliminates the need for end users to be aware of the summaries that have been defined. The database administrator creates one or more materialized views, which are the equivalent of summary tables. The advantage of using a materialized view rather than a CTAS is that a materialized view not only materializes the result of a query into a database table, but also generates metadata information used by the query rewrite engine to automatically rewrite the SQL query to use the summary tables. Materialized views within the data warehouse are transparent to the end user and the database application. Also, a materialized view optionally offers another important possibility: refreshing data automatically.

In the slide example, the user is able to execute the original query after the DBA created the materialized view called `CUST_SALES_MV`. Whenever the user or application executes the SQL query, Oracle Database transparently rewrites it to use the materialized view.

The query response time is the same as the CTAS approach, but the application need not be rewritten. The rewrite phase is automatically handled by the system. Also, the SQL statement that defines the materialized view does not have to match the SQL statement of the query itself.

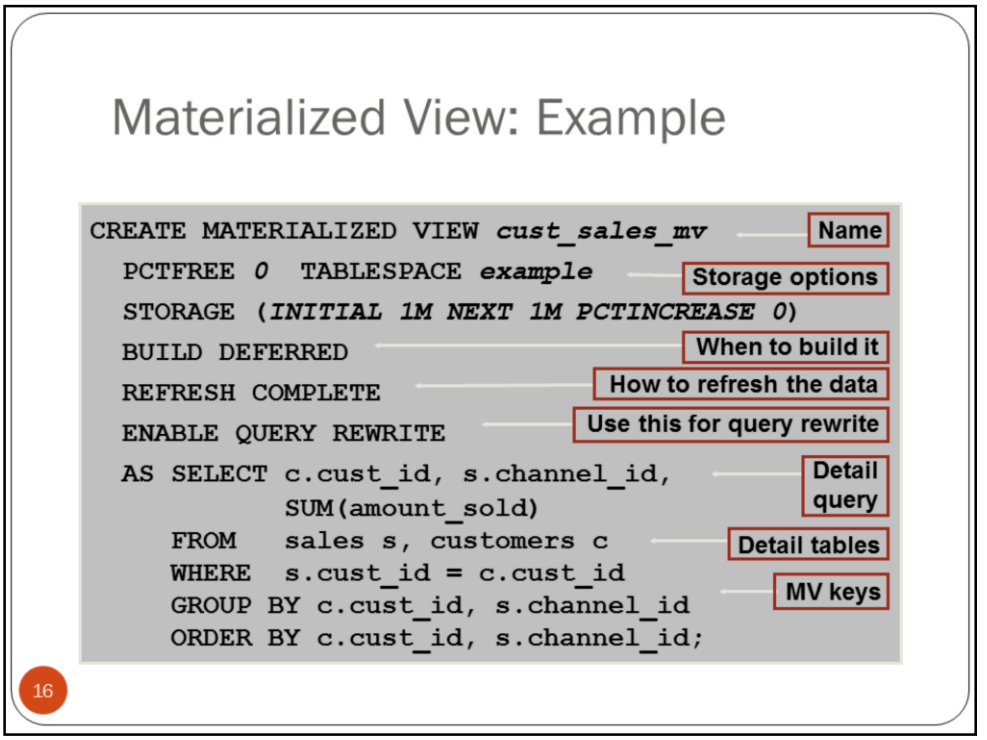


Using Materialized Views for Summary Management (continued)

Remember, in a database management system following the relational model, a view is a virtual table representing the result of a database query. In other words, views are tailored presentations of data contained in one or more tables or views. They do not require any space in the database

Every time the view is queried, the query defining the view is re-executed against the base tables. A materialized view is similar to a view but the data which is returned from the query is actually stored in a separate table.

In a typical use of summary management, the database administrator creates the materialized view. When the end user queries tables and views, the Oracle server's *query rewrite mechanism* automatically rewrites the SQL query to use the summary table. The use of the materialized view is transparent to the end user or application querying the data.



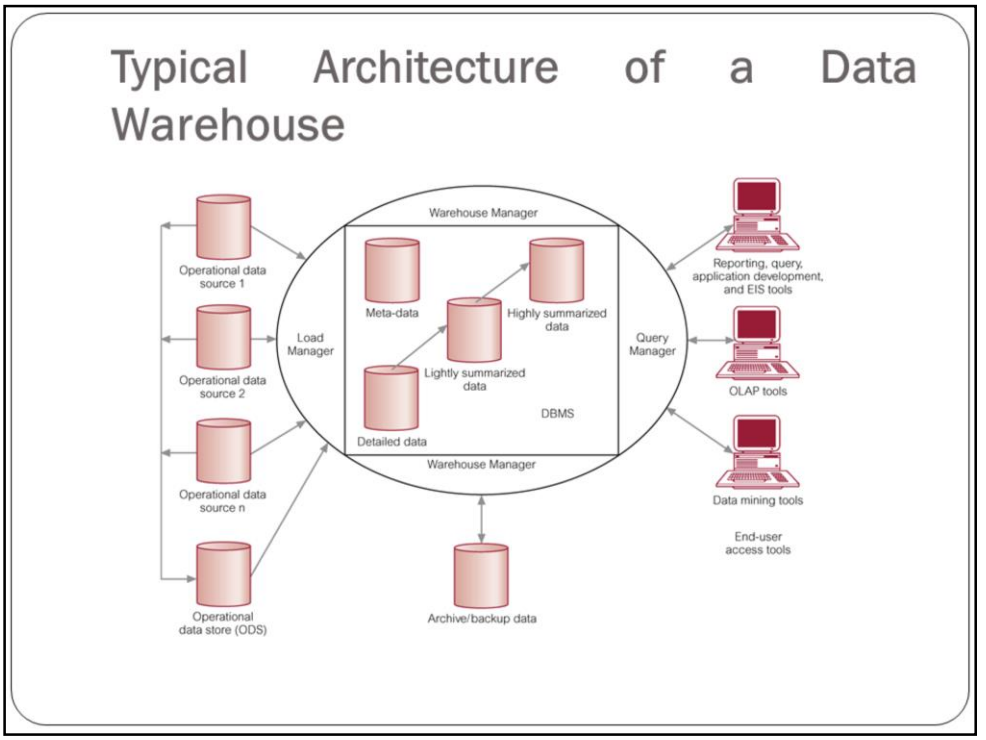
Materialized View: Example

For a complete description of the `CREATE MATERIALIZED VIEW` statement, refer to the *Oracle Database SQL Reference* guide.

Unless the materialized view is based on a user-defined prebuilt table, it requires and occupies storage space in the database.

Although query rewrite is enabled by default globally for the database, you must specify the `ENABLE QUERY REWRITE` clause if the materialized view is to be considered available for rewriting queries. This can be altered later using the `ALTER MATERIALIZED VIEW` statement. Note that you can enable query rewrite only if all user-defined PL/SQL functions in the materialized view are `DETERMINISTIC`.

You can specify an `ORDER BY` clause in the `CREATE MATERIALIZED VIEW` statement. It is used only during the initial creation of the materialized view and is not considered part of the materialized view definition. Storing rows in a specified order may help query performance because it provides physical clustering of the data that is very useful when using indexes.



Back to the common DW architecture

7. Archive / Backup Data

May need to store Stores detailed and summarized data for the purposes of archiving and backup. May be necessary to backup online summary data if this data is kept beyond the retention period for detailed data. The data can be transferred to storage archives such as magnetic tape or optical disk

8.Metadata Repository

This area of the warehouse stores all the metadata (data about data) definitions used by all the processes in the warehouse

Used by Decision Makers and for use by the data warehouse system itself

Metadata for Decision Makers should include:

- A lexicon of common words used in formal data names and data name abbreviation schemes
- A description of the logical data structure of the data warehouse database, and data integrity rules
- Inventory of the operational data maintained by the system and sources of that data
- A glossary of business words, terms and abbreviations that support use of the data

Can be used for a variety of other purposes

Extraction and loading processes

- The warehouse database schema, the schema of each information source and their mappings to the warehouse database schema
- Rules for extracting and cleaning and integrating data

Warehouse Management process

- Metadata is used to automate the production of summary tables.

Query Management process

- Metadata is used to direct a query to the most appropriate data source

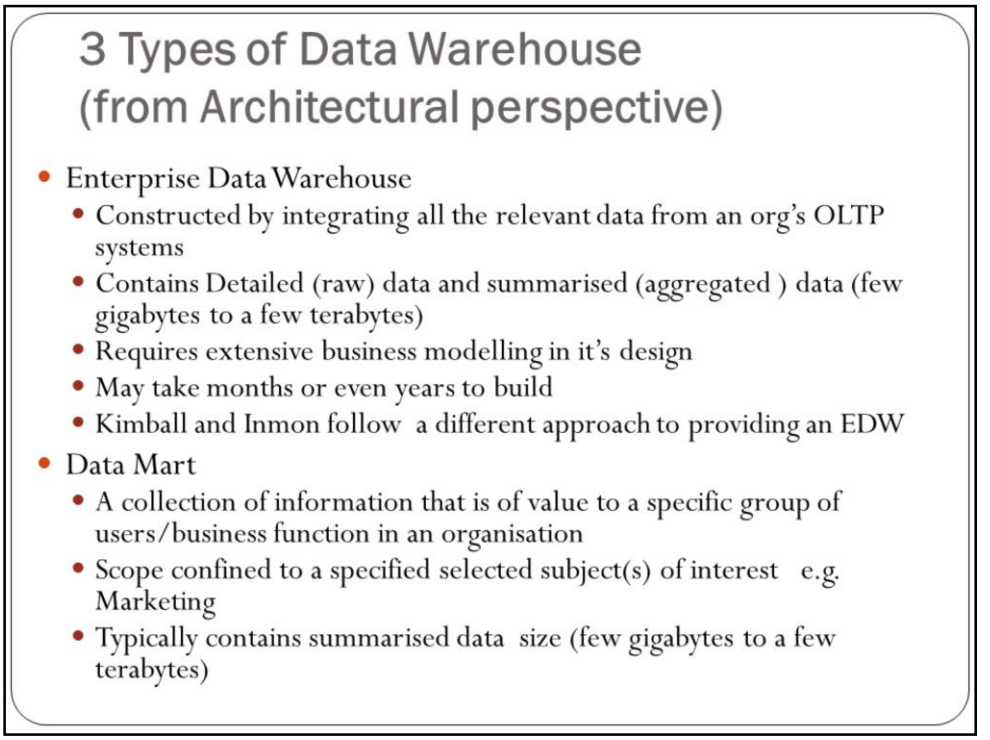
The structure of metadata will differ between each process, because the purpose is different.

Should also contain

- View definition, maintenance and use
- User identifiers, user authorisation and access control policies
- The currency of the data active, archived or purged

End-user access tools use metadata to understand how to build a query.

The management of metadata within the data warehouse is a very complex task that should not be underestimated.



Enterprise Data Warehouse

Note: Enterprise Data Warehouse may exist as a distributed data warehouse where the information is distributed over several independent data marts that collectively cover all subjects of interest.

Data Mart

A collection of information that is of value to a specific group of users within and organisation, or that supports specific products. As data marts contain less data compared with data warehouses, data marts more easily understood and navigated.

2 categories:-

Independent Data Mart

Sourced from data captured from one or more operational systems or external information providers

Dependant Data Mart

Sourced directly from an Enterprise Data Warehouse

Reasons for Creating a Data Mart

It gives users access to the data they need to analyze most often. It can also provide data in a form that matches the collective view of the data by a group of users in a department or business function area. It should also improve end-user response time due to the reduction in the volume of data to be accessed.

It can also provide appropriately structured data as dictated by the requirements of the end-user access tools. Building a data mart is simpler compared with establishing a corporate data warehouse. The cost of implementing data marts is normally less than that required to establish a data warehouse.

The potential users of a data mart more easily targeted to obtain support for a data mart project rather than a enterprise data warehouse project.

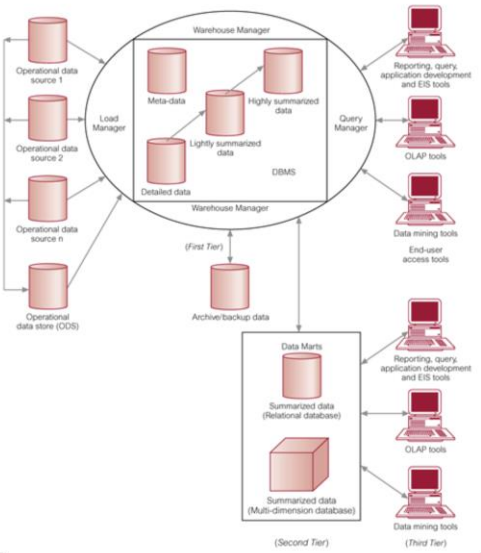
3 Types of DW
(from Architectural perspective)

- Virtual Data Warehouse
  - Decision makers have access to operational information services through a *collection of views* over the operational data.
  - Not considered to be a true Data Warehouse
  - Often used to prototype an Enterprise Data Warehouse on how decision makers may use operational data
  - Can you think of any disadvantages of a Virtual Data Warehouse?

Only current data

Performance Issues

Typical Data Warehouse and Data Mart Architecture



A "Hub and Spoke " Model

Some Problems with Data Warehousing Projects

- Underestimation of resources for data loading
- Required data not captured
- Increased end-user demands
- Data homogenization
- High demand for resources
- Data ownership
- High maintenance
- Long duration projects
- Complexity of integration