

Data Warehousing Concepts

This chapter provides an overview of the Oracle data warehousing implementation. It includes:

- [What is a Data Warehouse?](#)
- [Data Warehouse Architectures](#)

Note that this book is meant as a supplement to standard texts about data warehousing. This book focuses on Oracle-specific material and does not reproduce in detail material of a general nature. Two standard texts are:

- *The Data Warehouse Toolkit* by Ralph Kimball (John Wiley and Sons, 1996)
- *Building the Data Warehouse* by William Inmon (John Wiley and Sons, 1996)

What is a Data Warehouse?

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources.

In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.

A common way of introducing data warehousing is to refer to the characteristics of a data warehouse as set forth by William Inmon:

- [Subject Oriented](#)
- [Integrated](#)
- [Nonvolatile](#)
- [Time Variant](#)

Subject Oriented

Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?" This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented.

Integrated

Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated.

Nonvolatile

Nonvolatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable you to analyze what has occurred.

Time Variant

In order to discover trends in business, analysts need large amounts of data. This is very much in contrast to [online transaction processing \(OLTP\)](#) systems, where performance requirements demand that historical data be moved to an archive. A data warehouse's focus on change over time is what is meant by the term time variant.

Contrasting OLTP and Data Warehousing Environments

[Figure 1-1](#) illustrates key differences between an OLTP system and a data warehouse.

Figure 1-1 Contrasting OLTP and Data Warehousing Environments

OLTP		Data Warehouse
Complex data structures (3NF databases)		Multidimensional data structures
Few	Indexes	Many
Many	Joins	Some
Normalized DBMS	Duplicated Data	Denormalized DBMS
Rare	Derived Data and Aggregates	Common

One major difference between the types of system is that data warehouses are not usually in [third normal form \(3NF\)](#), a type of data normalization common in OLTP environments.

Data warehouses and OLTP systems have very different requirements. Here are some examples of differences between typical data warehouses and OLTP systems:

- Workload

Data warehouses are designed to accommodate *ad hoc* queries. You might not know the workload of your data warehouse in advance, so a data warehouse should be optimized to perform well for a wide variety of possible query operations.

OLTP systems support only predefined operations. Your applications might be specifically tuned or designed to support only these operations.

- Data modifications

A data warehouse is updated on a regular basis by the ETL process (run nightly or weekly) using bulk data modification techniques. 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.

- Schema design

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.

- Typical operations

A typical data warehouse query scans thousands or millions of rows. For example, "Find the total sales for all customers last month."

A typical OLTP operation accesses only a handful of records. For example, "Retrieve the current order for this customer."

- Historical data

Data warehouses usually store many months or years of data. This is to support historical analysis.

OLTP systems usually store data from only a few weeks or months. The OLTP system stores only historical data as needed to successfully meet the requirements of the current transaction.

Data Warehouse Architectures

Data warehouses and their architectures vary depending upon the specifics of an organization's situation. Three common architectures are:

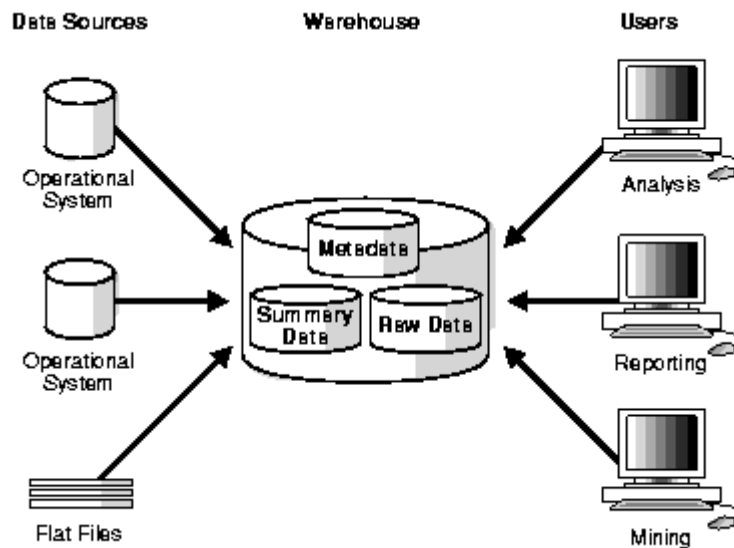
- [Data Warehouse Architecture \(Basic\)](#)

- [Data Warehouse Architecture \(with a Staging Area\)](#)
- [Data Warehouse Architecture \(with a Staging Area and Data Marts\)](#)

Data Warehouse Architecture (Basic)

[Figure 1-2](#) shows a simple architecture for a data warehouse. End users directly access data derived from several source systems through the data warehouse.

Figure 1-2 Architecture of a Data Warehouse

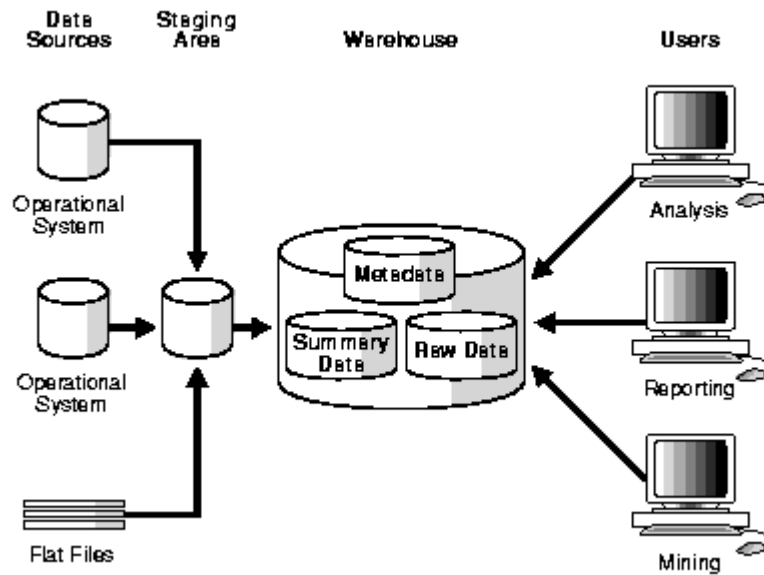


In [Figure 1-2](#), the metadata and raw data of a traditional OLTP system is present, as is an additional type of data, summary data. Summaries are very valuable in data warehouses because they pre-compute long operations in advance. For example, a typical data warehouse query is to retrieve something like August sales. A summary in Oracle is called a [materialized view](#).

Data Warehouse Architecture (with a Staging Area)

In [Figure 1-2](#), you need to clean and process your operational data before putting it into the warehouse. You can do this programmatically, although most data warehouses use a [staging area](#) instead. A staging area simplifies building summaries and general warehouse management. [Figure 1-3](#) illustrates this typical architecture.

Figure 1-3 Architecture of a Data Warehouse with a Staging Area



Data Warehouse Architecture (with a Staging Area and Data Marts)

Although the architecture in [Figure 1-3](#) is quite common, you may want to customize your warehouse's architecture for different groups within your organization. You can do this by adding **data marts**, which are systems designed for a particular line of business. [Figure 1-4](#) illustrates an example where purchasing, sales, and inventories are separated. In this example, a financial analyst might want to analyze historical data for purchases and sales.

Figure 1-4 Architecture of a Data Warehouse with a Staging Area and Data Marts

