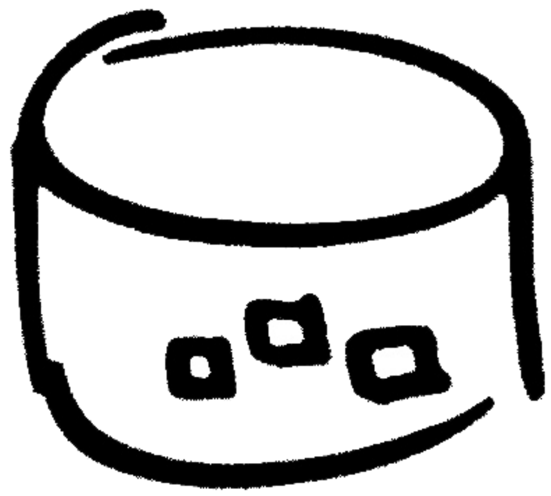


The Teradata Database

Part 1: Database Basics

Welcome to the first in a series of courses designed to provide you with an introduction to the Teradata Database.

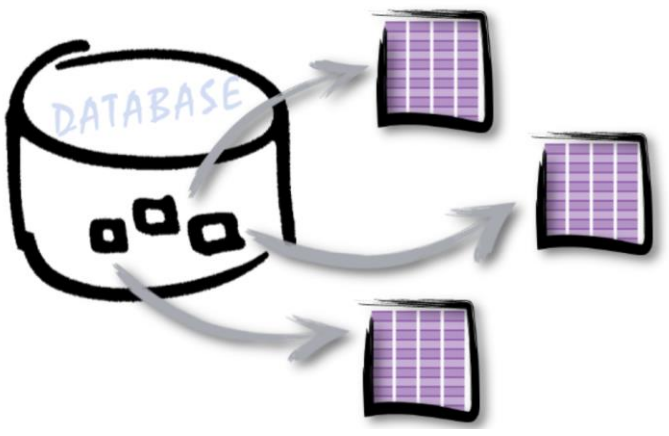
This course, Database Basics, serves as a refresher, or provides a foundation, to those who are new to the Teradata Database.

By the end of this course you should be able to:

- Describe how a database stores and organizes data.
- Identify the characteristics of a relational database.
- Discuss how relationships are established between tables.
- Define the function of a Relational Database Management System (RDBMS).
- Identify common RDBMS programs.
- Discuss reasons for deploying a data warehouse architecture.
- List the main components of a data warehouse.
- Describe how data flows in a data warehouse.
- List the deployment options for data warehouses.

Module 1 - Database Concepts

What is a Database?

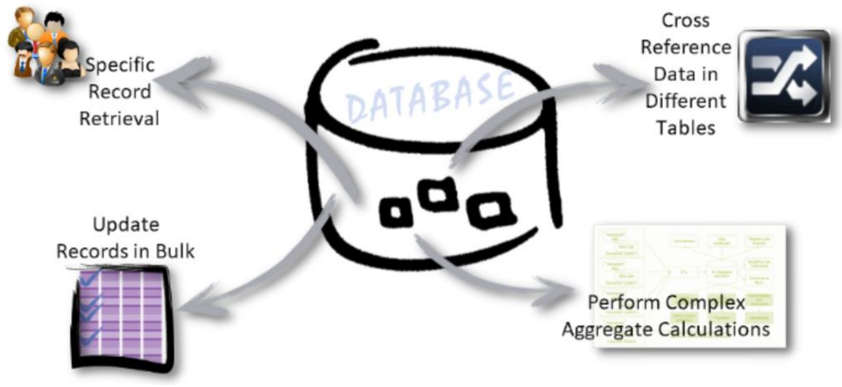


Databases provide an organized mechanism for storing, managing and retrieving data. Most modern databases are made up of tables.

To help you visualize what a table looks like, think about a spreadsheet, like Microsoft Excel.

An Excel spreadsheet contains rows, and columns, and organizes data in a similar way to a table in a database and would be considered an example of a flat file.

How You Can Use a Database


Specific Record Retrieval
DATABASE
Cross Reference Data in Different Tables
Update Records in Bulk
Perform Complex Aggregate Calculations

It is important to note that not all databases can perform the same actions. Here are just a few of the actions that you can perform:

- Retrieve all records that match certain criteria,
- Update records in bulk,
- Cross-reference records in different tables, and
- Perform complex aggregate calculations.

Databases in Your Life

You encounter the power of databases all the time in your daily life. For example, when you log into your online banking account, it is a database operating behind the scenes that evaluates your username, and password combination, and then provides you access to your account.

The database can filter your transactions to display the details by date or type, as you request.

Common Types of Databases

All databases provide a way for people to access data to solve problems, obtain an answer, or conduct research. Databases are typically referred to as one of the two types shown below:

Relational
A relational database is structured to recognize relations between stored items of information. Some of the more common relational databases that you might be familiar with are Teradata and Oracle. The standard user and application program interface to a relational database is the structured query language (SQL).

NoSQL
If your goal is only to retrieve information a NoSQL database will likely suffice. DynamoDB and MongoDB are two of the more well known today.

Summary

So far you have learned some of the basic elements of a database.
In the next module you will learn about a specific type of database – the Relational Database.

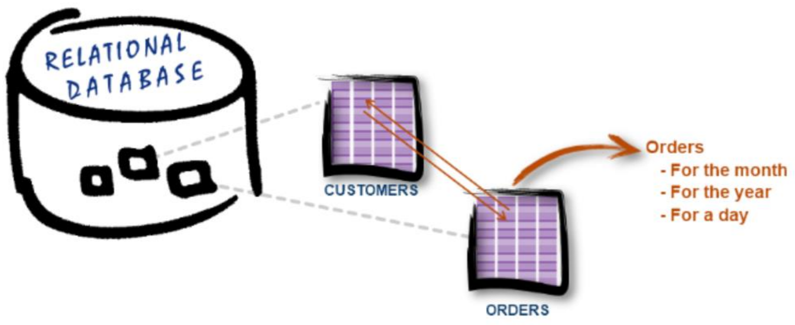
Module 2 - Relational Databases

What is a Relational Database?

If the database is relational, which most databases are, it can cross-reference records in different tables. This means that you can create relationships between tables.

For instance, if you can link the Customers table, with the Orders table. You can then find all purchase orders in the Orders table that an individual customer from the Customers table ever processed.

You can further refine it to return only those orders processed in a particular time period – a month, or a year, or a day.



You can use almost any combination you can imagine, like only certain shipping locations, or only purchase orders that took longer than a week to fulfill.

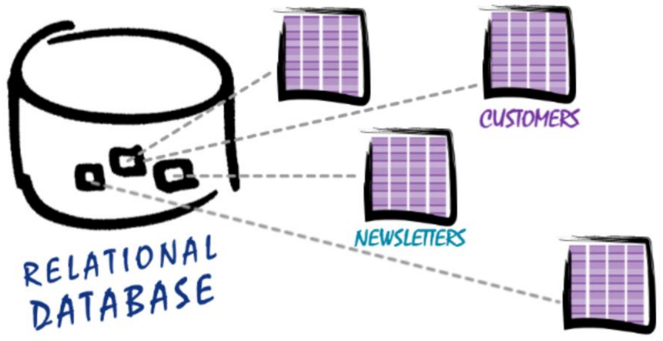
A Relational model for database management is based on concepts from mathematical set theory.

Relational Concept	
Entity	A person, place, or thing about which the table contains information.
Cardinality	The number of rows.
Degree	The number of columns.

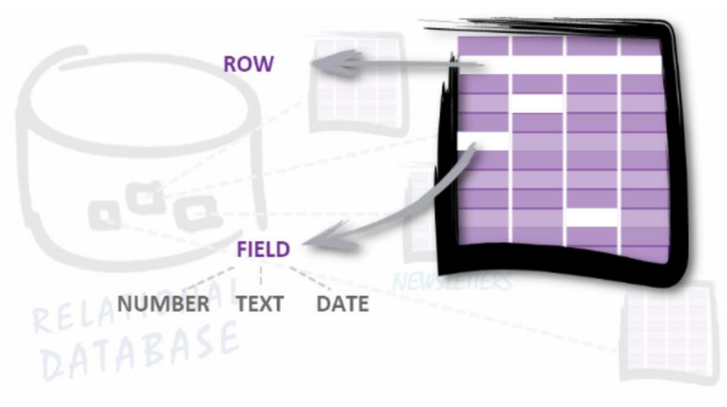
Mathematical Concept	
Relation	Table
Tuple	Row
Attribute	Column

Now let's look at how a relational database organizes data.

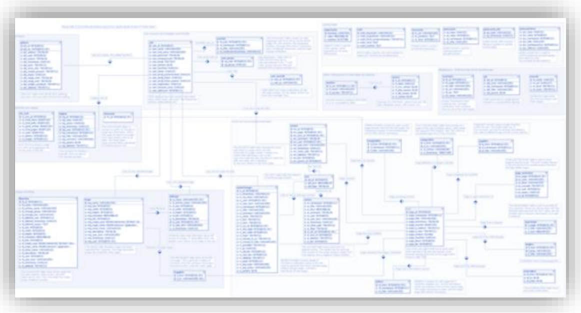
How is the Data Organized?



In a relational database, data is separated by categories into tables. For example, a business might have one table for customer contact information, and one table containing newsletters that are available. Tables are two-dimensional objects consisting of rows and columns.

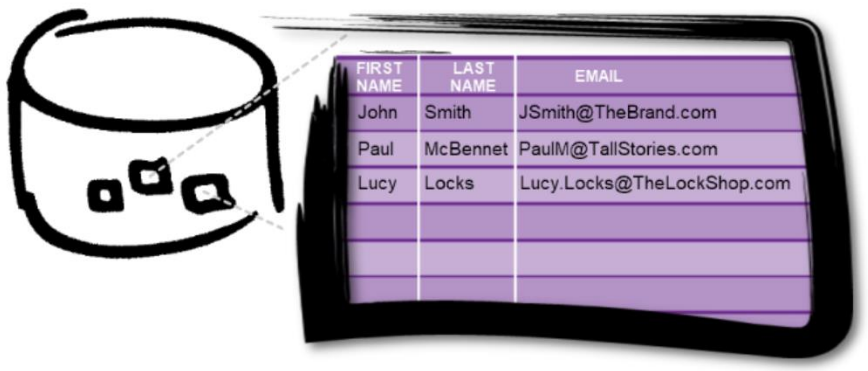


Each record in a table is called a row, and columns can also be called fields. Each field is designed to hold a specific type of data such as number, text or a date. Each row contains one instance of all the columns in a table.



If you hear the term "schema" it is referring to the organization of data as a blueprint of how the database is divided into tables. The formal definition of a database schema is a set of formulas or sentences called integrity constraints imposed on a database.

Let’s go back to our table called contacts. This table would contain the contact information for the businesses customers. In addition to our tables being uniquely named within the database, so also, are the columns in the table. This doesn't mean each column in the entire database needs to be unique. Only the columns within a given table need to be unique.



FIRST NAME	LAST NAME	EMAIL
John	Smith	JSmith@TheBrand.com
Paul	McBennet	PaulM@TallStories.com
Lucy	Locks	Lucy.Locks@TheLockShop.com

So in our “Contacts” table, we will have one column called first name, one called last name, and another called email.

- John Smith - j smith @ The brand dot com
- Paul McBennet - Paul M @ Tall stories dot com
- Lucie Locks - Lucie dot Locks @ The Lock Shop dot com

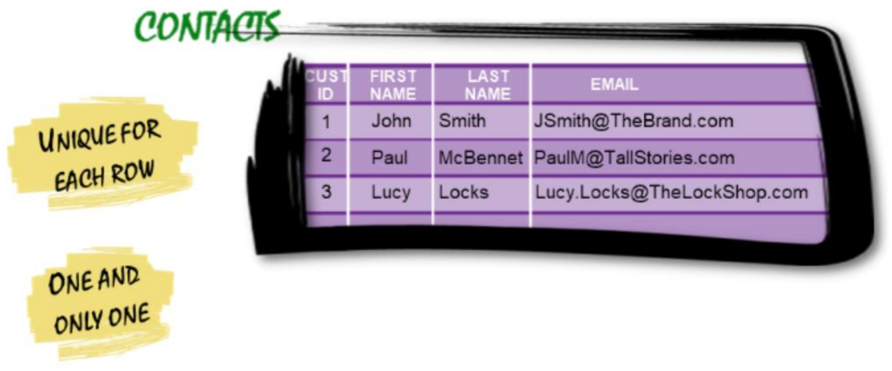
The Primary Key

The next thing to understand about your table is something called the Primary Key.

In a Relational Database, a primary key, is a unique identifier for each record or row. For example, a driver license number in a table containing licensed drivers, a telephone number (including area code) in a table containing contact information, or vehicle identification number (VIN) in a table containing cars and other vehicles.

A table in a relational database must always have one and only one primary key.

What is the Primary Key in the table shown here?



CUST ID	FIRST NAME	LAST NAME	EMAIL
1	John	Smith	JSmith@TheBrand.com
2	Paul	McBennet	PaulM@TallStories.com
3	Lucy	Locks	Lucy.Locks@TheLockShop.com

There is none, because nothing can be guaranteed unique. There are many people that might share these same last names, and or first names, and those people could be added to the database in the future.

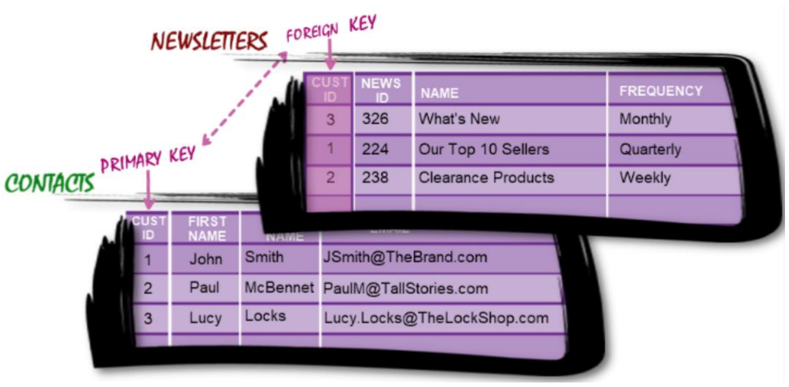
The email address is much more likely to be unique, but again, what if two people shared the same email?

Sometimes, a business will choose to allow the computer to generate a unique number to use as the Primary Key. For example, you may allow the computer to generate the customer id numbers in the customer table, since it can ensure that it never uses the same number twice and then the newsletter id in our newsletter table.

Database Relationships

Ok, the people are in one table and the newsletters in another table. But what if you want to know who is subscribed to what? One of the most powerful features of a database is the ability to create relationships between tables using something called a foreign key.

Establishing a Relationship Between Tables



CUST ID	NEWS ID	NAME	FREQUENCY
3	326	What's New	Monthly
1	224	Our Top 10 Sellers	Quarterly
2	238	Clearance Products	Weekly

CUST ID	FIRST NAME	NAME	
1	John	Smith	JSmith@TheBrand.com
2	Paul	McBennet	PaulM@TallStories.com
3	Lucy	Locks	Lucy.Locks@TheLockShop.com

A foreign key is a column or group of columns in a relational database table that define the relationship between the data in two tables. Foreign keys and their implementation require careful planning.

When a Primary Key, also referred to as a PK, is included in another table, it becomes a foreign key in the other table. Referential integrity is a relational database concept, which states that table relationships must always be consistent.

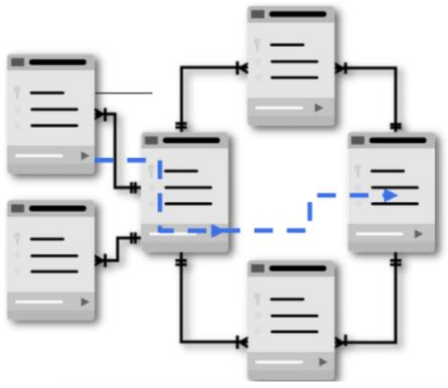
More About Referential Integrity

In order for the table relationships to prevent inconsistencies, the foreign key field must agree with the primary key that is referenced by the foreign key. Thus, any primary key field changes must be applied to all foreign keys, or not at all. The same restriction also applies to foreign keys in that any updates (but not necessarily deletions) must be propagated to the primary parent key.

This is especially important when performing inserts, updates, merges, and deletions.

Data Models

Data Models define how elements of data are connected or related.



As a model is refined, it passes through different states which can be referred to as normal forms. Another term you should be familiar with is third normal form or (3NF). Think of 3NF as rules and guidelines about how the data model should look. In practice this simply means which columns should belong to which tables.

Let's take a look at some of the more common types of models.

Relational Data Models

One common type of model is called a Relational Model. A relational model should reflect business rules.

Let's come back to our example with the Customers and Newsletters. The business event is the subscribing to a newsletter. So what are the business rules associated with that event? You might ask questions like "Does a customer need to buy something from us to be considered a customer?" "Can a customer subscribe to multiple newsletters?" "Can a customer only be a person, or can a customer be a business?" "Do you want to link multiple email addresses with a Customer?"

The actual implementation of these business rules in the model is more complicated than we will get into here, but these are the kinds of things that we consider when creating a relational model.

Dimensional Data Models

On the other hand, the Dimensional model is designed for reporting. It models navigation paths, not business rules and aligns with the work process of business users. A dimensional model emphasizes usability because the tables are defined in a way that a business uses the data, and organized in a way that a given user or group of users think about the data.

Its purpose is to allow efficient, user-friendly filtering, sorting and summing of data specific to the needs of one or more business area(s).

Using the same scenario, we want to be able to report by time, newsletter, and customer. We want to ask questions like “How many customers subscribed to this newsletter last month?” What are the 3 most popular newsletters from last year?”.

More About Dimensional Modelling

Dimensional modeling is a logical design technique that seeks to present the data in a standard, intuitive framework that allows for high-performance access. It adheres to a discipline that uses the relational model with some important restrictions. Every dimensional model is composed of one table with a multipart key, called a fact table, and a set of smaller tables called dimension tables. Each dimension table has a single-part primary key that corresponds exactly to one of the components of the multipart key in the fact table.

Normalization vs. Denormalization

Normalization is the process of reducing a complex database schema into a simple, stable one. In order to normalize, you must be able to uniquely identify a reading. Generally, this process involves removing redundant attributes, keys, and relationships from the conceptual data model.

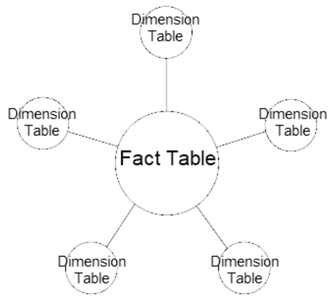
Dimension denormalization supports dimensional modeling's twin objectives of speed and simplicity.

While normalized data is optimized for entity level transactions, de-normalized data is optimized for answering business questions and driving decision making.

What A Dimensional Model Looks Like

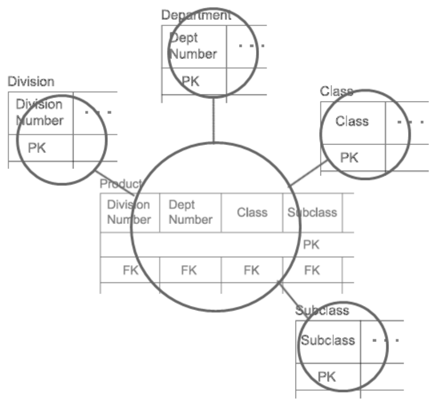
In a dimensional model, fact tables always represent M:M (many to many) relationships.

According to the model, a fact table should contain one or more numerical measures (the "facts" of the fact table) that occur for the combination of keys that define each tuple in the table.



This graphic illustrates the classical star schema.

Every dimensional model is composed of one table with a multipart key, called the fact table, and a set of smaller tables called dimension tables. Each dimension table has a single-part primary key that corresponds exactly to one of the components of the multipart key in the fact table.

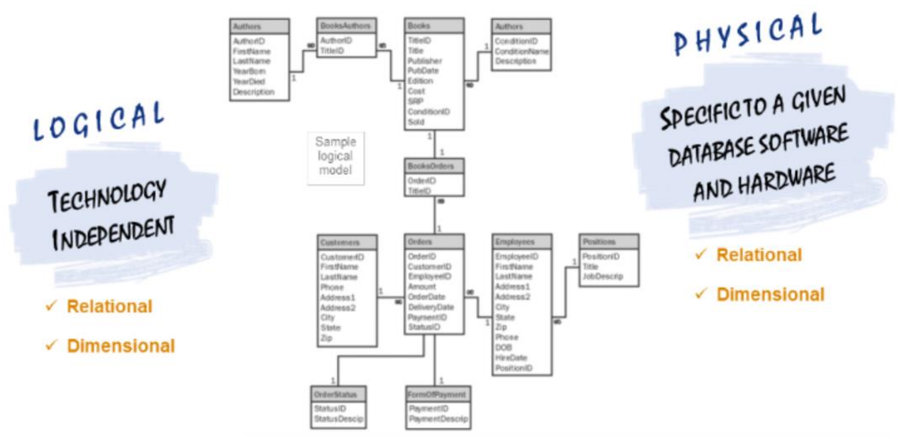


The graphic indicates a simplified example of a fact table (Product) and its associated dimension tables.

Logical and Physical Data Models

What is the difference between logical and physical data models?

Logical data models are not implemented on any hardware.



A logical model is said to be technology independent. A physical data model, on the other hand, is the technical solution. It is specific to a given database software and hardware. Logical and Physical models can be either Relational or Dimensional.

Summary

So far you have learned some of the basic elements of a database and about Relational Databases.
In the next module you will learn about the Relational Database Management System (RDBMS).

Module 3 - Relational Database Management Systems (RDB

What is an RDBMS?

A database simply holds data. To make real use of the data, you need a Database Management System or DBMS.

A DBMS is the all the software, and system software used for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

A DBMS create reports, enforces database rules and constraints, and maintains the database schema. Without it, a database is just a collection of bits and bytes with little meaning. The most common type of database management system is a relational database management system.

Common Relational Database Management Systems

Here are some of the commonly used relational database management systems:

TERADATA.

The Teradata Database provides a single up-to-date view of the enterprise enabling strategic queries, tactical queries, in-database analytics all controlled with sophisticated systems management.

Microsoft® SQL Server®

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

IBM DB2

IBM DB2 is designed to store, analyze and retrieve the data efficiently. DB2 is extended with the support of Object-Oriented features and non-relational structures with XML.

ORACLE®
DATABASE

Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system produced and marketed by Oracle Corporation.

What is SQL?

Relational database management systems use SQL (structured query language) as a database query language. Just as humans communicate with the help of language, SQL is the language that a database understands. SQL is used to create, manage, manipulate, and query database objects such as tables.

Using our earlier business example if we wanted to find customers with the last name Locks, the SQL query that we could use would be something like the one shown here.

```
SELECT email FROM
contacts WHERE
LastName = 'Locks';
```

Users of graphic user interface (GUI) front end tools such as Business Objects for example, generate and submit SQL to the database however the actual queries will not be visible to the user. This is because the tool does not require the user to know SQL – the tool generates the SQL.

Summary

Great, you have now learned some of the basic elements of a database, about Relational Databases, and you just learned about the Relational Database Management System (RDBMS).

In the next, and final module of this course, you will learn about the Data Warehouse.

Module 4 - The Data Warehouse

What is a Data Warehouse?

So, we've discussed databases in general and also a specific type of database - the Relational Database. But what about if you want to be able to perform analytics?

Generally, a database designed to handle transactions isn't designed to handle, or structured, to do analytics well. To effectively perform analytics, you need a data warehouse.

A data warehouse is a specially constructed data repository where data is organized so that it can be easily accessed by end users for various applications. Many get their data directly from operational systems making the data timely.

Data warehouses are common in corporations where enterprise-wide detailed data is analyzed to make strategic and tactical business decisions. This is because data warehouses often carry many years' worth of detailed data so historical trends may be analyzed. Finally, data warehouses may begin as small in scope and purpose, but often grow quite large if their utility becomes more fully exploited by the enterprise.

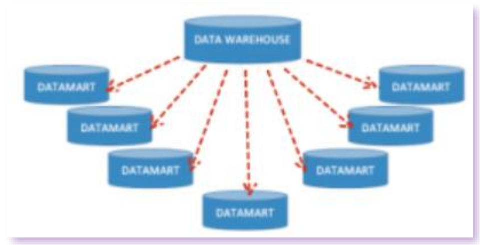
It is important to understand that data warehousing is a process, not a product. It is a technique to properly assemble and manage data from various sources to answer business questions not previously possible or known.

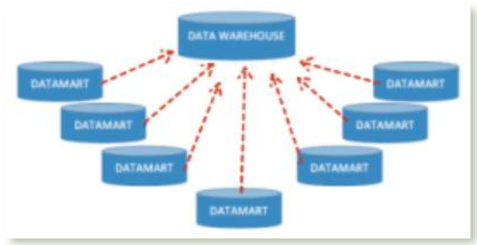
Data Marts

Another term you may hear when discussing date warehouses is data mart. A data mart is a subset of the data warehouse and is usually oriented to a specific business line or team. Whereas data warehouses have an enterprise-wide depth, the information in data marts pertains to a single department.

DATA WAREHOUSE
DATAMART
DATAMART
DATAMART
DATAMART

Data marts can be **independent of**, **dependent on**, or **part of** a data warehouse.





Analytic Processing

A data warehouse was originally defined as a decision support system (DSS). As we previously discussed, data warehouses are computer-based information systems that support business or organizational decision-making activities. OLAP and OLTP are sets of operations that can be performed on a data set.

OLAP (On-Line Analytic Processing) operations are the complex analysis that are performed on a data set within a data warehouse. These operations include data mining, querying, pivoting, slicing, dicing, drilling, reporting and other decision-support applications. You can have a data warehouse and not use OLAP at all, you just run reports.

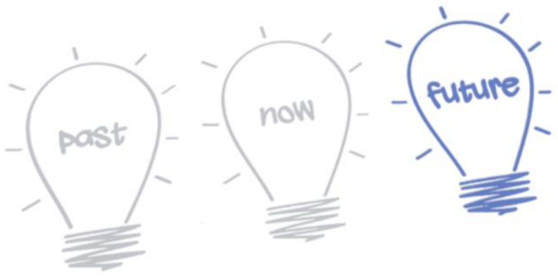
OLTP (On-Line Transaction Processing) manages business applications and collects the business's day-to-day data like order entry, financial transactions, or sales. This is different than a data warehouse. OLTP accesses a small number of rows (or records), or a few of many possible tables, in a matter of seconds or less.

ATM
Card is validated at the ATM.
Debit transaction at the ATM.
Balance is updated real time.

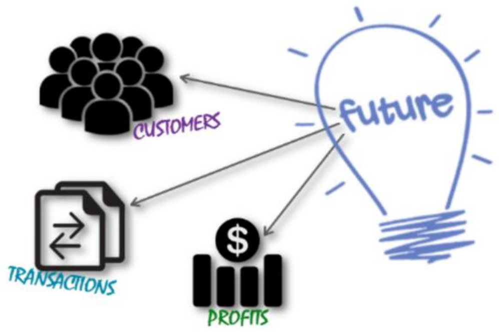
Most of you will be familiar with this example – taking out money at an ATM. Once your card is validated, a debit transaction takes place against your current balance and your balance gets updated. We expect these transactions to be performed quickly. They must occur in real-time.

Reasons for Deploying a Data Warehouse

For companies to be successful they must make good business decisions. Making good business decisions requires businesses to have the ability to analyze their related, relevant business data.

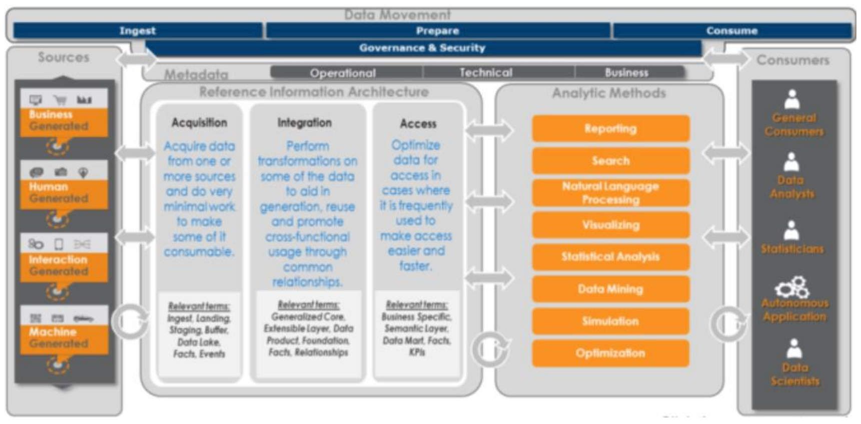


A data warehouse is an indispensable tool for businesses. While business data can be useful, it means little, if it cannot be stored and analyzed in a useful way. A key driver of analytics is strategic queries – business questions that are intended to provide a strategic advantage through forecasting.



Businesses can also learn more about their customers, transactions, and profits – studying these items to find patterns which can assist then in avoiding costly mistakes, enhance business productivity, and manage customer relationships.

The Data Warehouse Architecture - Layers


Data Movement
Ingest
Prepare
Consume
Governance & Security
Sources
Metadata
Operational
Technical
Business
Consumers
Reference Information Architecture
Analytic Methods
Business Generated
Human Generated
Interaction Generated
Machine Generated
Acquisition
Acquire data from one or more sources and do very minimal work to make some of it consumable.
Integration
Perform transformations on some of the data to aid in generation, reuse and promote cross-functional usage through common relationships.
Access
Optimize data for access in cases where it is frequently used to make access easier and faster.
Relevant terms: Ingest, Landing, Staging, Buffer, Data Lake, Facts, Events
Relevant terms: Generalized Core, Extensible Layer, Data Product, Foundation, Facts, Relationships
Relevant terms: Business Specific, Semantic Layer, Data Mart, Facts, KPIs
Reporting
Search
Natural Language Processing
Visualizing
Statistical Analysis
Data Mining
Simulation
Optimization
General Consumers
Data Analysts
Statisticians
Autonomous Application
Data Scientists

This diagram shows the source of the data and what happens to it.

The reference information architecture is broken down into 3 major data layers, i.e. acquisition, integration and access.

Acquisition:
As the first point of entry into the data warehouse, raw data is acquired from various source systems such as mainframe, server, etc.

Integration:
The integration layer is primarily responsible for integrating data from multiple systems both normalized and potentially de-normalized. It also may create common metrics and summaries which are widely used within an organization.

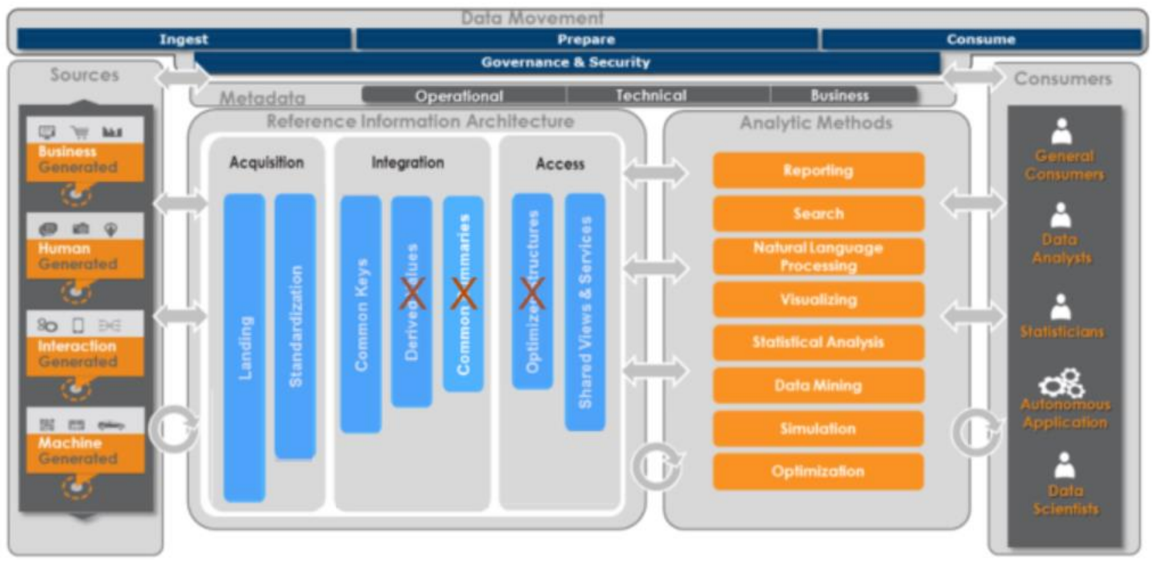
Access:
The access layer's primary responsibility is to provide easy access to the data using various analytic methods.

One example of this is business intelligence, or BI, tools. Through interaction with the tool, the Consumers can perform multi-dimensional analysis, or create custom reports etc. depending on their needs.

Data Warehouse Architecture-Tiers

Next, we will take a look at the data tiers.

Data layers are further broken down into data tiers. Data tiers provide a finer grain of how data progresses through the data layers.



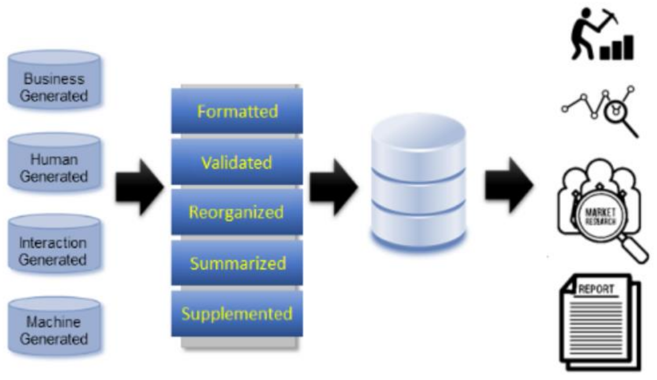
Not all the data tiers are used for every feed coming in from a data source. For example, there may not be a business need for derived values. Or if performance is not an issue, common summaries and/or optimized structures data tiers would not be needed. However, if these types of needs surface, these data tiers are specifically designed to address them.

Refer to the blue tiers above:

- The landing tier houses data in its raw, unprocessed form at the lowest level of granularity.
- The Standardization tier processes data into a "consumable" format.
- Common keys - This tier standardizes heavily reused keys that are the basis for connecting subject areas.
- The derived values tier is where enterprise, governed Key Performance Indicators (KPIs) are defined and automated.
- Summarization both for performance and consistency happens in the common summaries tier.
- Optimized Structures: This tier in the Access Layer is about performance,
- Standard Views and Services - Techniques such as materialized views and metadata services are created to assist users in navigating and consuming the data.

How Data Flows Through a Data Warehouse

To summarize, as was previously mentioned, data is extracted from various different data sources including those shown here.

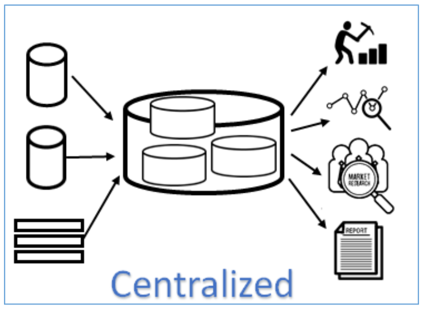


When the data is moved to a dedicated server that contains a data warehouse it can be formatted, validated, reorganized, summarized, and supplemented with data from many other sources as needed.

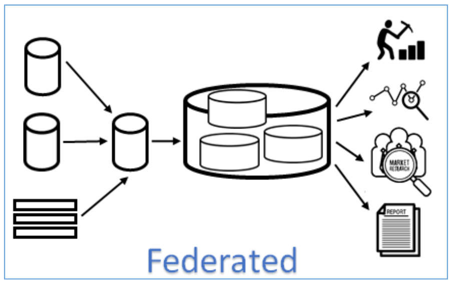
The resulting data warehouse becomes the main source of information for data mining, online analytic processing (OLAP), market research, report generation and analysis via reporting tools. These reporting tools can be used for such things as ad-hoc queries, canned reports, and dashboards.

Types of Implementation

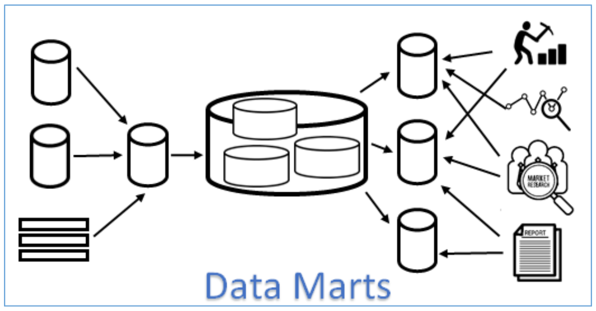
The types of implementations of data warehouses we are about to discuss, are just a few of many possibilities.



A Centralized Data Warehouse is great for small and mid-size data warehouses. It is a single physical repository that can contain organizational data for a specific utility area, department, branch, division or the whole organization. Centralized data warehouses serve the needs of several separate business units within an organization at the same time using a single data model.



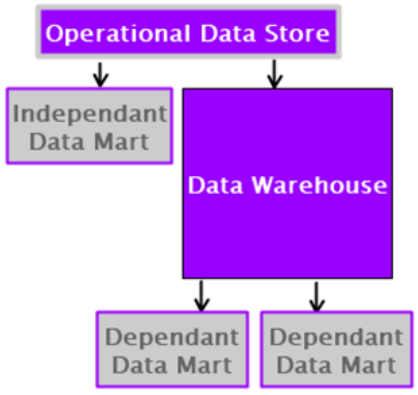
A federated data warehouse shares information among a number of different systems. Critical master files will be shared, and the other systems will be able to use this information. An example of this would be adding financial information to advertising data. This will allow an organization to better market its products to customers. If the data is federated organizations can reduce response time to business needs.



A data mart exists within a single organizational data warehouse repository. Ideally it a condensed and more focused version of the data warehouse that stores data dedicated to a specific business function or department within the organization. This subset of data may span across many, or all, of the organization's functional subject areas. It is common for multiple data marts to be used in order to serve the needs of each individual business unit such as accounting, marketing, sales, etc.

More About Data Marts

Data marts are often categorized into three different types:



- Independent data marts are isolated entities, entirely separate from the enterprise data warehouse. Their data derives from independent sources and they should be viewed as data pirates in the context of the enterprise data warehouse because their independent inputs, which are entirely separate from the enterprise data warehouse, have a high likelihood of producing data that does not match that of the warehouse.
- Dependent data marts are derived from the enterprise data warehouse. Depending on how a dependent data mart is configured, it might or might not be useful. The recommended process uses only data that is derived from the enterprise data warehouse data store and also permits its users to have full access to the enterprise data store when the need to investigate more enterprise-wide issues arises.
- The logical mart is a form of dependent data mart that is constructed virtually from the physical data warehouse. Data is presented to users of the mart using a series of SQL views that make it appear that a physical data mart underlies the data available for analysis.

Deployment Options

There's no right or wrong answer when organizations are choosing whether to deploy a data warehouse on-premises or in a cloud. Some organizations may find that some data and applications are low impact and relatively easy to transition to the cloud. Other critical data and applications may be best kept on-premises. It all depends on the business they are in, the data they possess, and their comfort level with having a third-party manage the risk. It is highly likely that in the foreseeable future we may see an increase in hybrid solutions, which is where on-premises and cloud co-exist. Each organization is different and will need to determine the most appropriate option for them.

In general, data warehouses can be deployed using one of the following strategies.

- The on-premises deployment option involves buying the software and hardware from a data warehouse company. Some examples of these companies would be Teradata, Oracle, or IBM. The on-premises (sometimes abbreviated as "on-prem") data warehouse is installed, runs, and is overseen on the premises of the organization using it. Benefits of this option are that it gives the organization total control, flexibility, accessibility, and predictability, and dependable performance. Some consider this option to also be the most secure, and, might even be required to use on-premises depending on the sensitivity of the organization's data. Examples of this might be highly regulated markets such as healthcare or financial. There are also some countries that restrict access to internet content so on-premises would be the only viable solution.

- If a company chooses to implement its data warehouse in the public or private cloud then it will run on a computing platform that belongs to the cloud provider, and access to it is provided, as a service, by a provider. Some examples of companies that offer this deployment option are Amazon (AWS), or Microsoft (Azure), or Google cloud platform.

  A public cloud solution can be quick to set up, is scalable, accessible, and easy to use.

  As mentioned, while discussing the on-premises solution, security might be an issue for some organizations depending on the market they are in. Trust issues are often the most difficult to overcome for IT executives when considering whether or not to move to a cloud-based data warehouse. One consideration that organizations need to take in to account, is that cloud options require high bandwidth which may not be an option for some smaller organizations.

  A private cloud, such as Teradata's IntelliCloud option, can provide a more custom, secure virtual environment than a public cloud.

Summary

Excellent, you have now completed the last module. You have learned some of the basic elements of a database, about relational databases, and Relational Database Management Systems (RDBMS), and finally about data warehouses. We've come to the end of this course. You now either have a good foundation of knowledge about databases, or have refreshed your memory related to this topic.