**What is a Data warehouse? Why we need Data warehouse?**

**According to,**

**Ralph Kimball: A**data warehouse is a relational database that is designed for querying and analyzing the business but not for transaction processing.

It usually contains historical data derived from transactional data( different source systems).

**According to,**

**W.H.Inmon:** A Data warehouse is a Subject oriented, integrated, time variant and non-volatile collection of Data used to support strategic decision Making process.

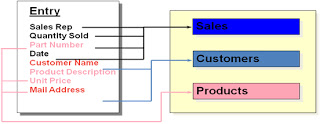
Characteristic features of a Data warehouse

* [Subject Oriente](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49840)d
* [Integrated](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49842)
* [Nonvolatile](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49871)
* [Time Variant](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49853)

**Note**: The first data warehousing system is implemented in 1987 by W.H.Inmon

[**Subject Oriented**](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49840)**:**The data warehouses are**designed**as a Subject-oriented that are used to analyze the business by top level management, or middle level management, or for a individual department in an enterprise.

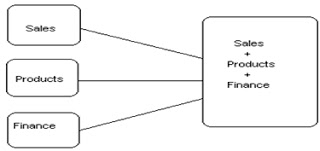
                                            Process Oriented                 Subject Oriented

[](http://3.bp.blogspot.com/_8QgP_p1pq74/TL8eLT6TwwI/AAAAAAAAAAQ/6mZYc0gEI2w/s1600/1.bmp)

**Transactional Storage       Data Warehouse Storage**

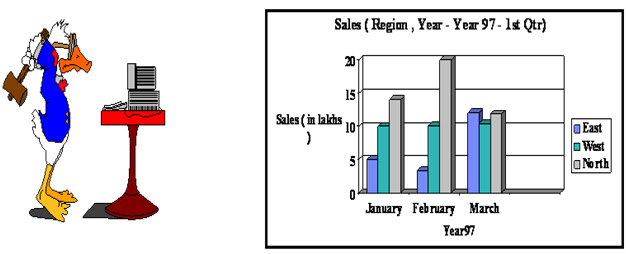
**For example**, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?" This ability to define a data warehouse by subject matter, sales in this case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

[**Integrated**](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49842)**: A** data warehouse is an integrated database which contains the business information collected from various operational data sources.

[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL8eyPtTQpI/AAAAAAAAAAU/2_5gDdKH1qU/s1600/2.bmp)

[**Time Variant**](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49853)**: A**Data warehouse is a time variant database which allows you to analyze and compare the business with respect to various time periods (Year, Quarter, Month, Week, Day) because which maintains historical data.

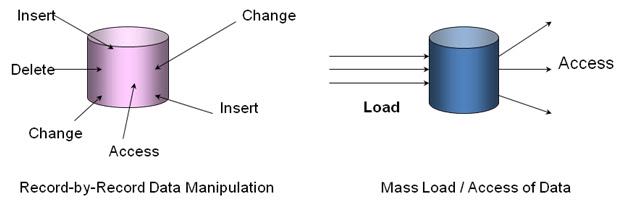
              Current Data                                                              Historical Data

[](http://2.bp.blogspot.com/_8QgP_p1pq74/TL8fRSDKbwI/AAAAAAAAAAY/h9vgSQMWLRI/s1600/3.bmp)

**Transactional Storage                                             Data Warehouse Storage**

[**Nonvolatile**](http://www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96520/concept.htm#49871)**:  A**Data warehouse is a non-volatile database. That means once the data entered into data warehouse can not change. It doesn’t reflect to the changes taken place in operational database. Hence the data is static.

                         Volatile                                                                   Non- Volatile

[](http://3.bp.blogspot.com/_8QgP_p1pq74/TL8fxfOOXFI/AAAAAAAAAAc/i5NHbFn09Nc/s1600/4.bmp)

**According to,**

**Babcock -**Data Warehouse is a repository of data summarized or aggregated in simplified form from operational systems. End user orientated data access and reporting tools let user get at the data for decision support.

**Why we need Data warehouse?**

*          To Store Large Volumes of Historical Detail Data from Mission Critical Applications
*          Better business intelligence for end-users
*          Data Security - To prevent unauthorized access to sensitive data
*          Replacement of older, less-responsive decision support systems
*          Reduction in time to locate, access, and analyze information

**Evaluation:**

         60’s:  Batch reports

  hard to find and analyze information

  inflexible and expensive, reprogram every new request

         70’s: Terminal-based DSS and EIS (executive information systems)

o   still inflexible, not integrated with desktop tools

         80’s:  Desktop data access and analysis tools

o   query tools, spreadsheets, GUIs

o   easier to use, but only access operational databases

         90’s:  Data warehousing with integrated OLAP engines and tools

**What is an Operational System? OR What is OLTP?**

*          Operational systems are the systems that help us run the day-to-day enterprise operations.
*          On Line Transactional Processing   systems not built to hold history data.
*          The data in these systems are having current data only.
*          The data in these systems are maintained in 3 NF. The data is used for running the business that doesn’t used for analyzing the business.
*          The examples are online reservations, credit-card authorizations, and ATM  withdrawals etc.,

**Difference between OLTP and Data warehouse (OLAP)**

**In general we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.**

|  |  |
| --- | --- |
| **Operational System (OLTP)** | **Data warehouse (OLAP)** |
| It is designed to support business transactional processing. | It is designed to support decision making process. |
| Application oriented data | Subject oriented  data |
| Current data | Historical data |
| Detailed data | Summary data |
| Volatile data | Non-volatile data |
| Less history (3-6 months) | More history (5-10 years) |
| Normalization data | De-normalization data |
| Designed for running the business | Designed for analyzing the business |
| Supports E-R modeling | Supports Dimensional modeling |
| Clerical users can access this data | Knowledge users can access this data |
| DB Size –  100MB-GB | DB Size –  100GB-TB |
| Few Indexes | Many Indexes |
| Many Joins | Some Joins |

**Advantages of Data Warehousing:**

         High query performance

         Queries not visible outside warehouse

         Can operate when sources unavailable

         Can query data not stored in a DBMS

         Extra information at warehouse

o   Modify, summarize (store aggregates)

o   Add historical information

         Improves the quality and accessibility of data.

         Reduce the requirements of users to access operational data.

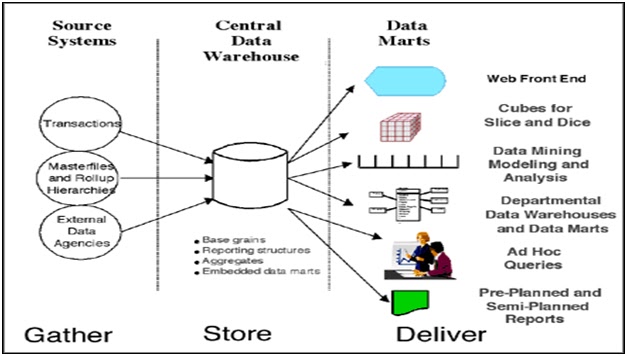
         Allows new reports and studies to be introduced without disrupting operational systems.

         Increases the amount of information available to users

**Types of Data warehouse:**

There are three types of data warehouses

  **Centralized data warehouse:** A centralised DW is one in which data is stored in a single, large primary database. This database can be queried directly or used to feed data marts.

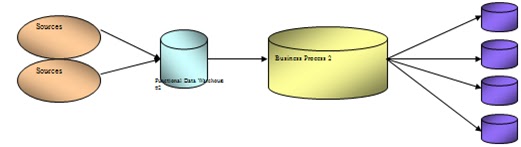
[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL8gn2R-N7I/AAAAAAAAAAg/n620j95RaBQ/s1600/5.bmp)

  **Functional data warehouse**: A functional DW is dedicated to a subset of the business, such as a marketing or finance business function.

         Separate DWs for different business capabilities

         Easier to build initially

         Several disadvantages – Redundancy, Cross-functional reporting not supported

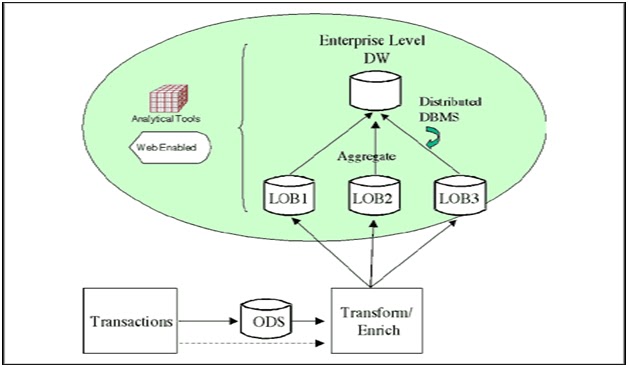
[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL8hQRbiAnI/AAAAAAAAAAk/BWJnQSP18zk/s1600/6.bmp)

  **Federated data warehouse**: A federated DW is an active union and cooperation across separate DWs.

         Different DWs communicate

         Requires active cooperation across multiple DWs

         No passive coexistence of separate systems

[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL8huBB48fI/AAAAAAAAAAo/QGoWr0OAAuw/s1600/7.bmp)

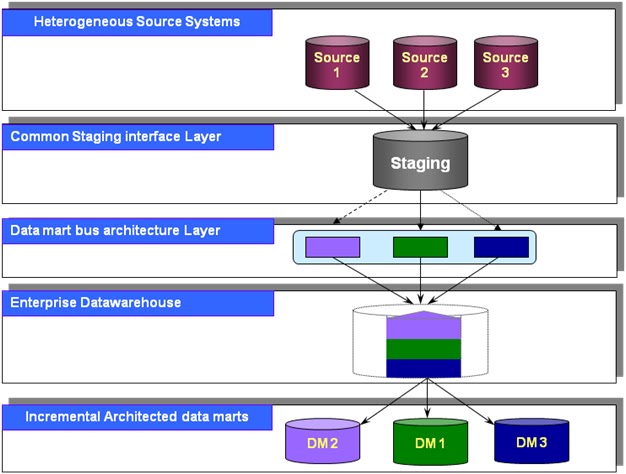
**Data Warehouse Implementation Approaches**

To build data warehouse there are two approaches:

*   **Top-Down  Approach**
*   **Bottom-Up Approach**

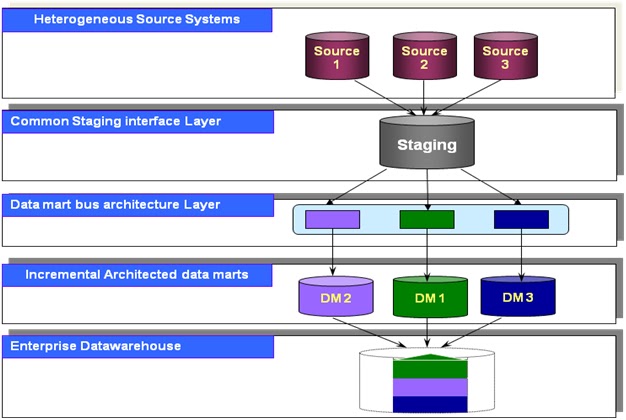
**Top-Down Approach:**

This approach  is developed by W.H.Inmon. According to him first we need to develop enterprise data warehouse. Then from that enterprise data warehouse develop subject-orient databases those are called as Data Marts.

[](http://4.bp.blogspot.com/_8QgP_p1pq74/TL8imAQaI_I/AAAAAAAAAAs/0Gy9_cEeVH8/s1600/8.bmp)

**Bottom-Up Approach:**

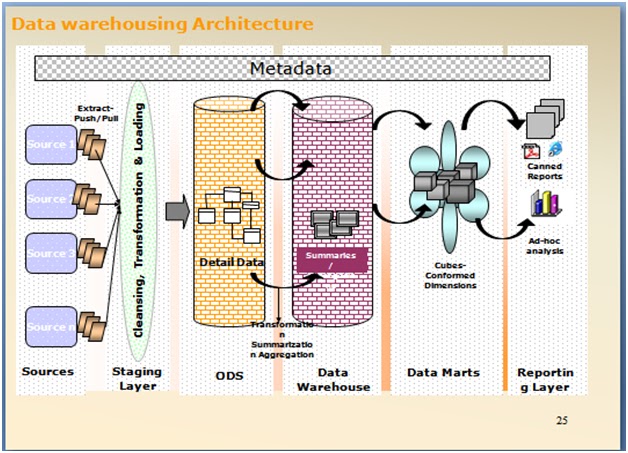
This approach is developed by Ralph Kimball. According to him first we need to develop the data marts to support the business needs of middle-level management. Then integrate all the data marts into an enterprise data warehouse.

[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL8wZVn9wnI/AAAAAAAAAAw/v0IPEyjvtmo/s1600/9.bmp)

**Top-Down Vs Bottom Up**

|  |  |
| --- | --- |
| **Top Down** | **Bottom Up** |
| 1. More planning and design initially 2. Involve people from different work-groups, departments 3. Data marts may be built later from Global DW 4. Overall data model to be decided up-front 5. High cost, lengthy process, time consuming. 6. Involved people from different work groups, departments. | 1. Can plan initially without waiting for global infrastructure 2. built incrementally 3. can be built before or in parallel with Global DW 4. Less complexity in design 5. Low cost of Hardware and other resources. 6. It is built in the incremental manner. |

**Data Warehouse Architecture**

[](http://3.bp.blogspot.com/_8QgP_p1pq74/TL8x2HTlpMI/AAAAAAAAAA0/DMVawBW_qnQ/s1600/10.bmp)

**Source System:** A System which provides the data to build a DW is known as Source System. Which are of two types.

* Internal Source
* External Source

**Data Acquisition:**

It is the process of extracting the relevant business information, transforming the data into the required business format and loading into the data warehouse.

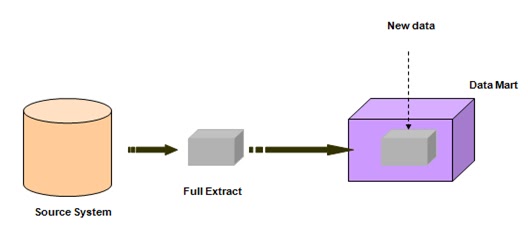
 A data acquisition is defined with the following process.

*          Extraction
*          Transformation
*          Loading

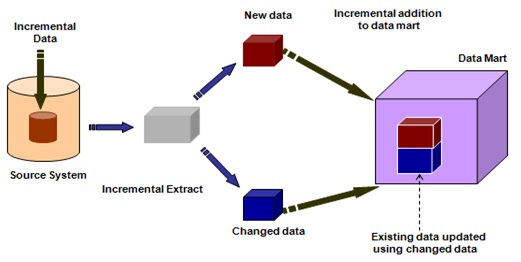
**Extraction:**The first part of an ETL process involves extracting the data from the different source systems like Operational Sources, XML files, Flat files, COBOL files, SAP, People soft, Sybase etc.,

There are two types of Extraction:

a.       Full Extraction

[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL80C_yDNtI/AAAAAAAAAA4/rUtdNwU7zy8/s1600/11.bmp)

b.      Periodic/Incremental Extraction

[](http://2.bp.blogspot.com/_8QgP_p1pq74/TL80egqAp8I/AAAAAAAAAA8/ww72ljp9Ky4/s1600/12.bmp)

**Transformation**: It is the process of transforming the data into a required business format. The following are the data transformation activities takes place in staging area.

*   Data Merging
*   Data Cleansing
*   Data Scrubbing
*   Data Aggregation.

**Data Merging**: It is a process of integrating the data from multiple input pipe lines of similar structure or dissimilar structure into a single output pipeline.

**Data Cleansing:** It is a process of identifying or changing the inconsistencies and inaccuracies.

Inconsistent – If the data is not in a proper format.

Inaccuracy: It is applicable for numeric values.

**Common software products for Name and Address cleansing**

•         Trillium: **:**Used for cleansing the name and address data. The software is able to identify and match households, business contacts and other relationships to eliminate duplicates in large databases using fuzzy matching techniques.

•         First Logic

**Data Scrubbing:** It is a process of deriving the new data definitions from existing source data definitions.

**Data Aggregation:**It is a process where multiple detailed values are summarized into a single summary values.

Ex: SUM, AVERAGE, MAX, MIN etc

**Loading:**It is the process of inserting the data into Data Warehouse. And loading are of two types:

*          Initial load
*          Incremental load

**ETL Products:**

1. 1.      CODE BASED ETL TOOLS
2. 2.      GUI BASED ETL TOOLS

**Code Based ETL Tools:**In these tools, the data acquisition process can be developed with the help of programming languages.

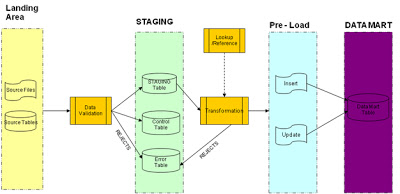
*          SAS ACCESS
*          SAS BASE
*          TERADATA ETL TOOLS
*   BTEQ (**B**atch **TE**radata **Q**uery)
*   TPUMP  (Trickle PUMP)
*   FAST LOAD
*   MULTI LOAD

Here the application Development, Testing, and Maintenance is costlier process compared to GUI based tools.

**GUI based ETL Tools:**

*          Informatica
*          DT/Studio
*          Data Stage
*          Business Objects Data Integrator (BODI)
*          AbInitio
*          Data Junction
*          Oracle Warehouse Builder
*          Microsoft SQL Server  Integration Services
*          IBM DB2 Ware house Center

**ETL Approach:**

[](http://3.bp.blogspot.com/_8QgP_p1pq74/TL81iiVWMhI/AAAAAAAAABA/qs_jhL20dMI/s1600/13.bmp)

1. **Landing Area** – This is the area where the source files and tables will reside.
2. **Staging** – This is where all the staging tables will reside. While loading into staging tables, data validation will be done and if the validation is successful, data will be loaded into the staging table and simultaneously the control table will be updated. If the validation fails, the invalid data details will be captured in the Error Table.
3. **Pre Load** – This is a file layer. After complete transformations, business rule applications and surrogate key assignment, the data will be loaded into individual Insert and Update text files which will be a ready to load snapshot of the Data Mart table. Once files for all the target tables are ready, these files will be bulk loaded into the Data Mart tables. Data will once again be validated for defined business rules and key relationships and if there is a failure then the same should be captured in the Error Table.
4. **Data Mart** – This is the actual Data Mart. Data will be inserted/updated from respective Pre Load files into the Data Mart Target tables.

**Operational Data Store (ODS):**An **operational data store** (or "**ODS**") is a [data base](http://en.wikipedia.org/wiki/Database) designed to integrate data from  multiple sources to make analysis and reporting easier.

Definition:The ODS is defined to be a structure that is:

*          Integrated
*          Subject oriented
*          Volatile, where update can be done
*          Current valued, containing data that is a day or perhaps a month old
*          Contains detailed data only.

**Why We Need Operational Data Store?**

To obtain a “system of record” that contains the best data (Complete, Up to date, Accurate) that exists in a legacy environment as a source of information.

**Note:**An "**ODS**" is not a replacement or substitute for an [enterprise data warehouse](http://en.wikipedia.org/wiki/Data_warehouse) but in turn could become a source.

**Operational Data Store – Data**

*          Detailed data
* o   Records of Business Events
* (e.g. Orders capture)
*          Data from heterogeneous sources
*          Does not store summary data
*          Contains current data

**OLTP Vs ODS Vs DWH**

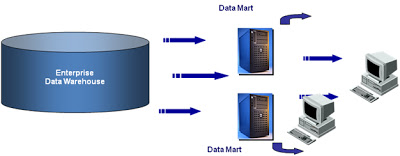
|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **OLTP** | **ODS** | **Data Warehouse** |
| Audience | Operating Personnel | Analysts | Managers and analysts |
| Data access | Individual records, transaction driven | Individual records, transaction or analysis driven | Set of records, analysis driven |
| Data content | Current, real-time | Current and near-current | Historical |
| Data granularity | Detailed | Detailed and lightly summarized | Summarized and derived |
| Data organization | Functional | Subject-oriented | Subject-oriented |
| Data quality | All application specific detailed data needed to support a business activity | All integrated data needed to support a business activity | Data relevant to management information needs |
| Data stability | Dynamic | Somewhat dynamic | Static |

**Data Mart:**

Data mart is a decentralized subset of data found either in a data warehouse or as a standalone subset designed to support the unique business requirements of a specific decision-support system.

A data mart is a subject-oriented database which supports the business needs of middle management like departments.

A data mart is also called **High Performance Query Structures (HPQS).**

[](http://4.bp.blogspot.com/_8QgP_p1pq74/TL82Yys73SI/AAAAAAAAABE/_BoytDbOuxM/s1600/14.bmp)

There are three types of data marts:

**Dependent data marts**:

In the top-down approach data mart development depends on enterprise data warehouse. Such data marts are called as dependent data marts.

Dependent data marts are marts that are fed directly by the DW, sometimes supplemented with other feeds, such as external data

**Independent data marts**:

In the bottom -up approach data mart development is independent on enterprise data warehouse. Such data marts are called as independent data marts.

Independent data marts are marts that are fed directly by external sources and do not use the DW.

**Embedded data marts** are marts that are stored within the central DW. They can be stored relationally as files or cubes.

**Data Mart Main Features:**

* 1.      Low cost
* 2.      Contain less information than the warehouse
* 3.      Easily understood and navigated than an enterprise data warehouse.
* 4.      Within the range of divisional or departmental budgets

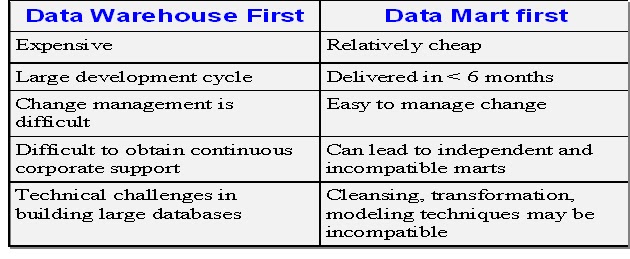
**Data Mart Advantages:**

* a.       Typically single subject area and fewer dimensions
* b.      Focused user needs
* c.       Limited scope
* d.      Optimum model for DW construction
* e.       Very quick time to market (30-120 days)

**Data Mart disadvantages:**

* a.       Does not provide integrated view of business information.
* b.      More number of data marts are complex to maintain
* c.       Scalability issues for large number of users and increased data volume

**Warehouse or Mart First ?**

[](http://2.bp.blogspot.com/_8QgP_p1pq74/TL83GbzM15I/AAAAAAAAABI/2gQAByOi3-I/s1600/15.bmp)

**Data warehouse - Design:**

A data warehouse is designed with the following types of schemas.

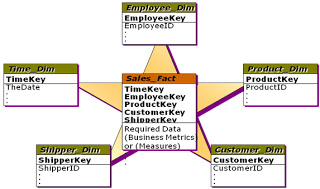
*          Star Schema Database Design
*          Snowflake Schema Database Design
*          Galaxy Schema (Hybrid schema, Integrated Schema, Fact Constellation Schema, Complex schema all these schemas called as galaxy schema)

**Schema** : A schema is a collection of objects including tables, views, indexes and synonyms

**Star Schema:**

**A** Star schema is a logical database design which contains a centrally  located fact table surrounded by dimension tables.

The data base design looks like a star. Hence it is called as Star schema.

[](http://2.bp.blogspot.com/_8QgP_p1pq74/TL83ncdqEcI/AAAAAAAAABM/A04CSS3Do7E/s1600/16.bmp)

The **star schema**  is also called *star-*[*join*](http://en.wikipedia.org/wiki/Join_(SQL))*schema*, *data cube*, or *multi-dimensional schema*

A fact table contains facts and facts are numeric.

Not every numeric is a fact but the numeric’s which are of type key performance indicators are known as facts.

Facts are business measures which are used to evaluate the performance of an enterprise.

A fact table contains the fact information at the lowest level granularity.

The level at which fact information is stored in a fact table is known as grain of fact or fact granularity or fact event level.

A dimension is a descriptive data about the major aspects of our business.

Dimensions are stored in dimension table which provides the answers to the four basic business questions ( who, what, when, where).

Dimensional table contains de-normalized   business information.

If a Star Schema Contains more than one fact table then that Star schema is called “Complex Star Schema”.

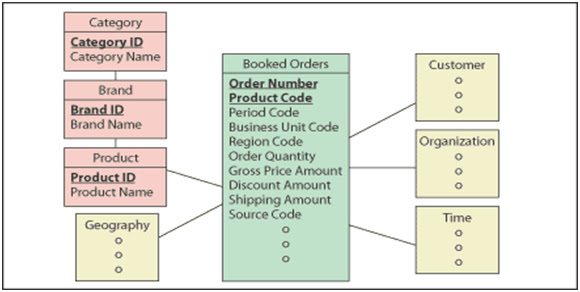
**Advantages of Star schema:**

*          Star schema is very easy to understand , even for non technical business managers
*          Provides better performance and smaller query times.
*          Star schema is easily understandable and will handle future changes easily.

**Snowflake Schema:**

The snowflake schema is similar to the [star schema](http://en.wikipedia.org/wiki/Star_schema). However, in the snowflake schema, dimensions are [normalized](http://en.wikipedia.org/wiki/Normalization_(database)) into multiple related tables, whereas the star schema's dimensions are denormalized with each dimension represented by a single table.

A large dimension table is split into multiple dimension tables.

[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL84JVqRAeI/AAAAAAAAABQ/W2EK3eadJ78/s1600/17.bmp)

**Star Schema Vs Snowflake Schema:**

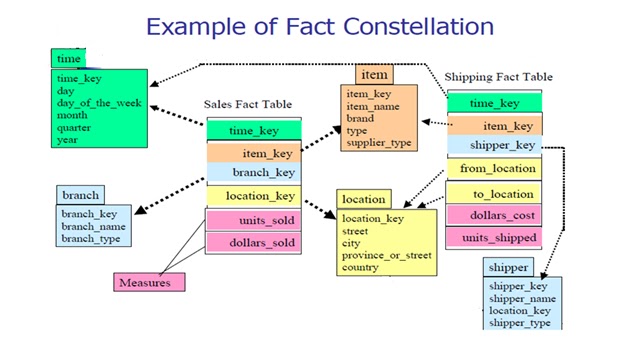
|  |  |
| --- | --- |
| **Star Schema** | **Snowflake** |
| A centralized fact table and surrounded by different dimensions | In the same star schema dimensions split into another dimensions |
| contains Highly Demoralized Data | contains Partially normalized |
| can not have parent table | contain parent tables |
| Contains less joiners | Contains more joiners |
| Comparatively performance is not very low because less joiners is there | performance is very low because more joiners is there |

**What is fact constellation schema?**

For each star schema it is possible to construct fact constellation schema(for example by splitting the original star schema into more star schemes each of them describes facts on another level of dimension hierarchies).

The fact constellation architecture contains multiple fact tables that share many dimension tables viewed as a collection of stars, therefore called **galaxy schema**.

The main shortcoming of the fact constellation schema is a more complicated design because many variants for particular kinds of aggregation must be considered and selected. Moreover, dimension tables are still large.

[](http://3.bp.blogspot.com/_8QgP_p1pq74/TL84pUDSrtI/AAAAAAAAABU/95dnt_r4EMU/s1600/18.bmp)

**What is a Dimension Table:**

If  a table contains primary keys and it gives the detailed information about business then such a table is called dimension table.

A Dimension table is a table which holds a list of attributes or qualities of the dimension most often used in queries and reports.

E.g. the “Store” dimension can have attributes such as the street and block number, thecity, the region and the country where it is located in addition to its name.

Dimension tables are ENTRY POINTS into the fact table. Typically -

•         The number of rows selected and processed from the fact table depends on the conditions (“WHERE” clauses) the user applies on the dimensional attributes selected

Dimension tables are typically DE-NORMALIZED in order to reduce the number of joins in resulting queries.

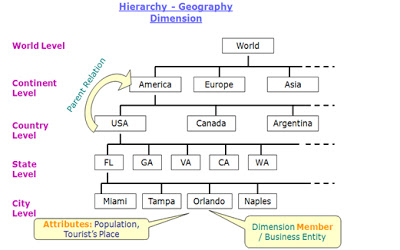
Dimension table attributes are generally STATIC, DESCRIPTIVE fields describing aspects of the dimension

Dimension tables typically designed to hold IN-FREQUENT CHANGES to attribute values over time using SCD concepts

Dimension tables are TYPICALLY used in GROUP BY SQL queries

Every column in the dimension table is TYPICALLY either the primary key or a dimensional attribute

Every non-key column in the dimension table is typically used in the GROUP BY clause of a SQL Query

[](http://1.bp.blogspot.com/_8QgP_p1pq74/TL85PbOSf1I/AAAAAAAAABY/K5RD96ChBpE/s1600/19.bmp)

* A hierarchy can have one or more levels or grain
* Each level can have one or more members
* Each member can have one or more attributes

**Types of Dimension table:**

**Conformed Dimension**: If a dimension table is shared by multiple fact tables then that dimension is known as conformed dimension table.

**Junk dimension:**Junk dimensions are dimensions that contain miscellaneous data like flags, gender, text values etc and which are not useful to generate reports .

**Dirty dimension:**In this dimension table records are maintained more than once by the difference of  non-key attributes .

**Slowly changing dimension:**If  the data values are changed slowly in a column or in a row over the period of time then that dimension table is called as slowly changing dimension.

Ex: Interest rate, Address of customer etc

There are three types of SCD’s:

* Type – 1 SCD: A type-1 dimension keeps the most recent data in the target.
* Type – II SCD: keeps full history in the target. For every update it keeps a new record in the target.
* Type – III SCD: keeps the current and previous information in the target (partial history).

**Note:** To implement SCD we use Surrogate key.

**Surrogate key:**

*          Surrogate Key is an artificial identifier for an entity. In surrogate key values are generated by the system sequentially (Like Identity property in SQL Server and Sequence in Oracle). They do not describe anything.
*          Joins between fact and dimension tables should be based on surrogate keys
*          Surrogate keys should not be composed of natural keys glued together
*          Users should not obtain any information by looking at these keys
*          These keys should be simple integers
*          Using surrogate key will be faster
*          Can handle Slowly Changing dimensions well

**Degenerated dimension:**A degenerate dimension is data that is dimensional in nature but stored in a fact table. For example, if you have a dimension that only has Order Number and Order Line Number, you would have a 1:1 relationship with the Fact table. Therefore, this would be a degenerate dimension and Order Number and Order Line Number would be stored in the Fact table.

**Fast Changing Dimension:** A fast changing dimension is a dimension whose attribute or attributes for a record (row) change rapidly over time.

Example: Age of associates, Income, Daily balance etc.

**Fact table:** A fact table which contains foreign keys to dimension tables and numeric facts (called as measurements).

Facts can be detailed level facts or summarized facts.

Each fact typically represents a business item, a business transaction, or an event that can be used in analyzing the business or business process.

The term FACT represents a single business measure. E.g. Sales, Qty Sold

Fact tables express MANY-MANY RELATIONSHIPS between dimensions in dimensional models

•         One product can be sold in many stores while a single store typically sells many different products at the same time

•         The same customer can visit many stores and a single store typically has many customers

The fact table is typically the MOST NORMALIZED TABLE in a dimensional model

•         No repeating groups (1N), No redundant columns (2N)

•         No columns that are not dependent on keys; all facts are dependent on keys (3N)

•         No Independent multiple relationships (4N)

Fact tables can contain HUGE DATA VOLUMES running into millions of rows

All facts within the same fact table must be at the SAME GRAIN.

Every foreign key in a Fact table is usually a DIMENSION TABLE PRIMARY KEY

Every column in a fact table is either a foreign key to a dimension table primary key or a fact

**Types of Facts:**

**T**here are three types of facts:

1. **Additive** - Measures that can be added across all dimensions.
2. **Semi Additive** - Measures that can be added across few dimensions and not with others.
3. **Non Additive** - Measures that cannot be added across all dimensions.

         A fact may be measure, metric or a dollar value. Measure and metric are non additive facts.

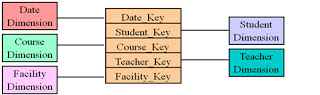
         Dollar value is additive fact. If we want to find out the amount for a particular place for a particular period of time, we can add the dollar amounts and come up with the total amount.

         A non additive fact, for eg measure height(s) for 'citizens by geographical location' , when we rollup 'city' data to 'state' level data we should not add heights of the citizens rather we may want to use it to derive 'count'.

**Types of Fact tables:**

**Fact less fact table:** A fact table without facts(measures) is known as factless fact table.

Ex: see the below diagram

[](http://2.bp.blogspot.com/_8QgP_p1pq74/TL86ikfvM6I/AAAAAAAAABc/1iUf7fV16m0/s1600/21.bmp)

**Factless fact table are of two types:**

1.   First type of factless fact table is called **Event Tracking table** or **records an event** i.e. Attendence of the student. Many event tracking tables in the dimensional DWH turns out to be factless table.
2.   Second type of factless fact table is called **coverage table**. Coverage tables are frequently needed in (Dimensional DWH) when the primary fact table is sparse.

**Conformed fact table:**If two fact tables measures same then that fact table is called conformed fact table.

**Snap-shot fact table**:T his type of fact table describes the state of things in a particular instance of time, and usually includes more semi-additive and non-additive facts.

**Transaction fact table:** A transaction is a set of data fields that record a basic business event.

 e.g. a point-of-sale in a supermarket, attendance in a classroom, an insurance claim, etc.

**Cumulative fact table**:This type of fact table describes what has happened over a period of time. For example, this fact table may describe the total sales by product by store by day. The facts for this type of fact tables are mostly additive facts.

**Dimensional Modeling:**

A dimensional modeling is a design methodology to design a data warehouse. Dimension modeling consists of three phases:

**Conceptual modeling:**

         In this the data modeler needs to understand the scope of the business and business requirements.

         After understand the business requirements modeler needs to identify the lowest level grains such as entities and attributes.

**Logical modeling:**

In this phase after find the entities and attributes ,

*          Design the dimension table with the lowest level grains which are identified at the conceptual modeling.
*          Design fact table with the key performance indicators.
*          Provide the relationship between the dimensional and fact tables using primary key and foreign key

**Physical modeling:**

*          Logical design is what we draw with a pen and paper or design with Oracle Designer or ERWIN before building our warehouse.
*          Physical design is the creation of the database with SQL statements.
*          During the physical design process, you convert the data gathered during the logical design phase into a description of the physical database structure.

[**Modeling tools available in the Market?**](http://www.oraclelove.com/2006/10/26/what-are-modeling-tools-available-in-the-market/)

These tools are used for Data/dimension modeling

*          Oracle Designer
*          Erwin (Entity Relationship for windows)
*          Informatica (Cubes/Dimensions)
*          Embarcadero
*          Power Designer Sybase

**OLAP (OnLine Analytical processing)**

* It is a set of specifications which allows the client applications (Reports) in retrieving the data from data warehouse.
* In the OLAP, there are mainly two different types: Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP). Hybrid OLAP (HOLAP) is the  combination of MOLAP and ROLAP.

**MOLAP**

This is the more traditional way of OLAP analysis. In MOLAP, data is stored in a multidimensional cube. The storage is not in the relational database, but in proprietary formats.

E.g. Hyperion Essbase, Oracle Express, Cognos PowerPlay (Server)

Advantages:

* Excellent performance: MOLAP cubes are built for fast data retrieval, and is optimal for slicing and dicing operations.
* Can perform complex calculations: All calculations have been pre-generated when the cube is created. Hence, complex calculations are not only doable, but they return quickly.

Disadvantages:

* Limited in the amount of data it can handle: Because all calculations are performed when the cube is built, it is not possible to include a large amount of data in the cube itself. This is not to say that the data in the cube cannot be derived from a large amount of data. Indeed, this is possible. But in this case, only summary-level information will be included in the cube itself.
* Requires additional investment: Cube technology are often proprietary and do not already exist in the organization. Therefore, to adopt MOLAP technology, chances are additional investments in human and capital resources are needed.

**ROLAP**

This methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality. In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.

E.g. Oracle, Microstrategy, SAP BW, SQL Server, Sybase, Informix, DB2

Advantages:

* Can handle large amounts of data: The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on data amount.
* Can leverage functionalities inherent in the relational database: Often, relational database already comes with a host of functionalities. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionalities.

Disadvantages:

* Performance can be slow: Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, the query time can be long if the underlying data size is large.
* Limited by SQL functionalities: Because ROLAP technology mainly relies on generating SQL statements to query the relational database, and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL), ROLAP technologies are therefore traditionally limited by what SQL can do. ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions.

**Multidimensional Vs Relational databases**

**Multidimensional Database**

* highly aggregated data
* dense data
* 95% of the analysis requirements

**Relational** **Database**

* detailed data (sparse)
* 5% of the requirements

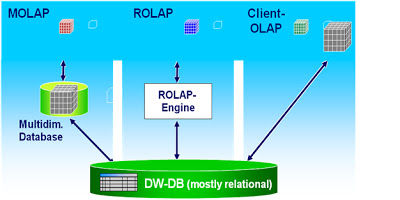
**HOLAP**

HOLAP technologies attempt to combine the advantages of MOLAP and ROLAP. For summary-type information, HOLAP leverages cube technology for faster performance. When detail information is needed, HOLAP can "drill through" from the cube into the underlying relational data.

**Client (Desktop) OLAP:**

* Proprietary data structure on the client
* data stored as file
* mostly RAM based architectures

E.g. Business Objects, Cognos PowerPlay, Foxpro, Clipper, Paradox

[](http://4.bp.blogspot.com/_8QgP_p1pq74/TL87T9M-tFI/AAAAAAAAABg/X4gJCnUFpS8/s1600/22.bmp)

**Project Life Cycle**

*          Requirements Analysis
*          High level Design(HLD)
*          Low level Design(LLD)
*          Development
*          Unit Testing
*          System Integration Testing
*          Peer Review
*          User Acceptance Testing
*          Production
*          Maintenance

**Requirement s Analysis:**

In this phase the client management, client users and development company management, business analyst of that company will sit together and conduct a meeting called “***kick-off***” meeting.

In this meeting the client submitted document to development company with the help of users. The document contains all the requirements. After 10/15/20 days again they will conduct a meeting called “***kick-off 2***”.

In this meeting the development company submits a document to the client with the help of “***Solution Architect***”. These documents contains all the solutions for their requirements.

**High level Design(HLD):**

In this phase, the development company will identify the source system, data warehouse(whether star schema or snow-flake schema) and staging area. And again they identifies which data base they want to use fir target and staging area.

In this phase, the **Project Manager** identifies,

*   How many resources are needed for the project( ETL, OLAP, Modeling and others).
*   What are the Software requirements? (ETL, OLAP, Modeling and others).
*   What are the Hardware requirements ( no.of System capacity, processers, RAM etc).

**Low level Design (LLD):**

In this phase, the data modeler design the database of data warehouse based on source system and  also modeler creates or designs the database for staging area and ODS with the help of modeling tools like ERWIN.

Finally the modeler converts the logical data model into physical model and generates scripts.

Then the DBA takes these specification models and crates tables physically.

Then the ETL lead or senior developer or OLAP lead prepares the “ETL Specification” or “OLAP spec” documents. These specs are the input to the developers.

**Development:**

In the development phase, the developer develops the ETL jobs based on the detailed ETL spec and load that data into target.

**Unit Testing:**

The ETL developers can perform the unit testing based on two inputs:

* 1.      ETL specs
* 2.      ETL jobs

The developer will check the job whether it has been performed correctly or not based on spec. the developer should note down the number of records extracted from source table , number of records loaded into target table, finally both will match or not.

**System Integration Testing:**

The SIT is performed by ETL lead or senior developer. In this phase they will integrate all moules or tables and check whether jobs are running perfectly or not and consider whether the dependencies are considered or not.

**Peer Review:**

In this phase, apart from developer third party person will check all the jobs status whether jobs are running correctly or not.

**User Acceptance Testing:**

In all the project phases this one is very important. In this, users of the client will check all the reports and examine whether the reports are meat our requirements or not. If they satisfied with these reports they will signoff otherwise sent back to development.

**Production:**

The production phase will do after completion of UAT. In this phase the development company checks one or two times and deliver the project to the client.

**Maintenance:**

In this phase, the development company provides the maintenance for 2-5 years based on their agreement.