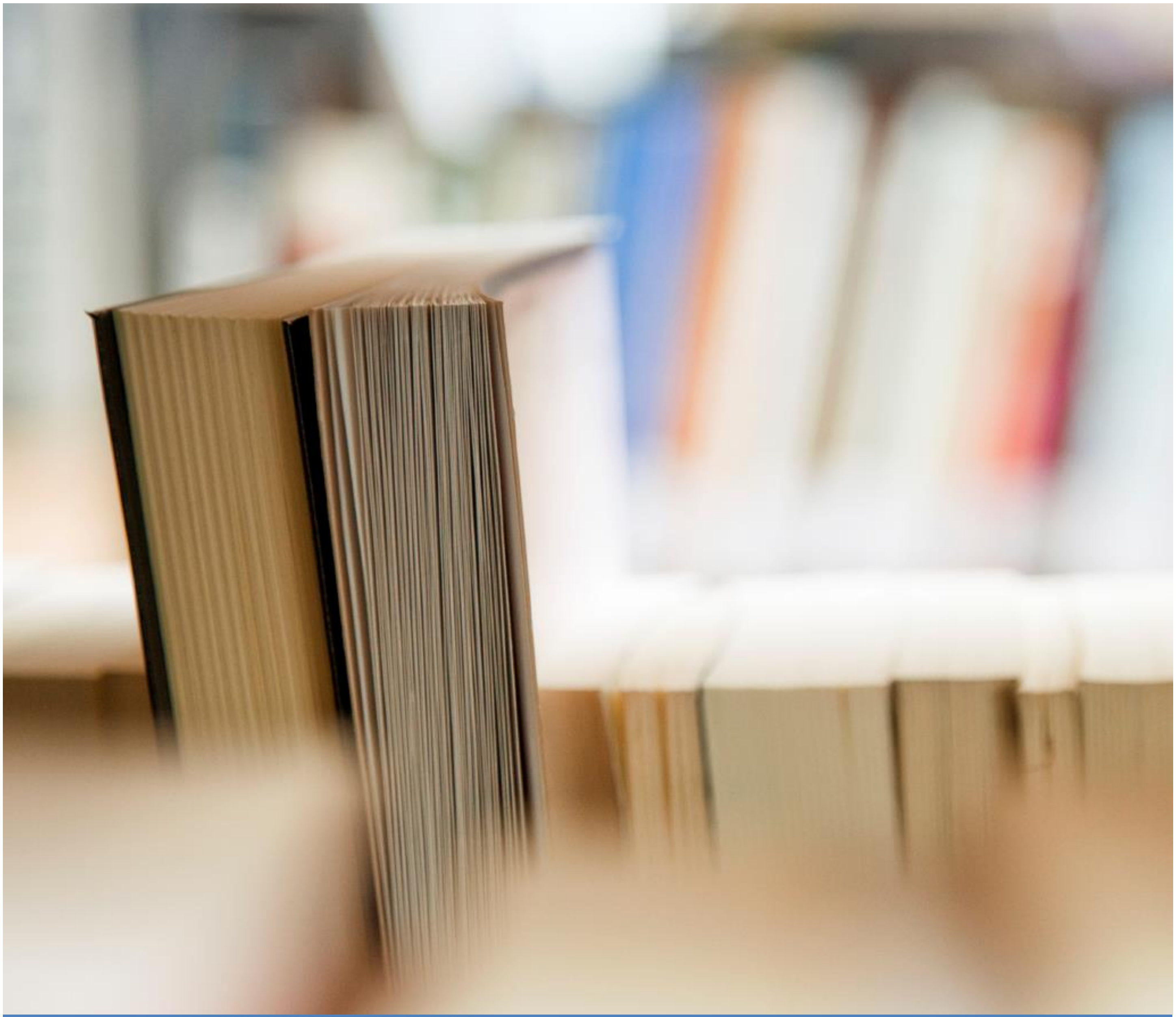


Data Warehouse Concepts

Handbook

Sid

IDWBI.COM

Table of Contents

What is BI?

Business Intelligence refers to a set of methods and techniques that are used by organizations for tactical and strategic decision making. It leverages methods and technologies that focus on counts, statistics and business objectives to improve business performance.

The objective of Business Intelligence is to better understand customers and improve customer service, make the supply and distribution chain more efficient, and to identify and address business problems and opportunities quickly.

Warehouse is used for high level data analysis purpose. It is used for predictions, time series analysis, financial Analysis, what -if simulations etc. Basically it is used for better decision making.

Why BI?

Make more informed business decisions:

- Competitive and location analysis
- Customer behavior analysis
- Targeted marketing and sales strategies
- Business scenarios and forecasting
- Business service management
- Business planning and operation optimization
- Financial management and compliance

What is a Data Warehouse?

Data Warehouse is a "Subject-Oriented, Integrated, Time-Variant Nonvolatile collection of data in support of decision making".

In terms of design data warehouse and data mart are almost the same.

In general, a Data Warehouse is used on an enterprise level and a Data Marts is used on a business division/department level.

Subject Oriented:

Data that gives information about a particular subject instead of about a company's ongoing operations.

Integrated:

Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.

Time-variant:

All data in the data warehouse is identified with a particular time period.

Non-volatile:

Data is stable in a data warehouse. More data is added but data is never removed.

What is a DataMart?

Data mart is usually sponsored at the department level and developed with a specific details or subject in mind, a Data Mart is a subset of data warehouse with a focused objective.

What is the difference between a data warehouse and a data mart?

In terms of design data warehouse and data mart are almost the same.

In general, a Data Warehouse is used on an enterprise level and a Data Marts is used on a business division/department level.

A data mart only contains data specific to a particular subject area.

Difference between data mart and data warehouse

Data Mart	Data Warehouse
Data mart is usually sponsored at the department level and developed with a specific issue or subject in mind, a data mart is a data warehouse with a focused objective.	Data warehouse is a "Subject-Oriented, Integrated, Time-Variant, Nonvolatile collection of data in support of decision making".
A data mart is usually focuses on a single subject area of a business like Sales, HR or Finance	A data warehouse is used on an enterprise level
A Data Mart is a subset of data from a Data Warehouse. Data Marts are built for specific user groups.	A Data Warehouse is simply an integrated consolidation of data from a variety of sources that is specially designed to support strategic and tactical decision making.
By providing decision makers with only a subset of data from the Data Warehouse, Privacy, Performance and Clarity Objectives can be attained.	The main objective of Data Warehouse is to provide an integrated environment and coherent picture of the business at a point in time.

What is fact less fact table?

A fact table that contains only primary keys from the dimension tables, and that do not contain any measures that type of fact table is called fact less fact table.

Example:

A student attendance in a class or a FIR filed against an accident.

What is a Schema?

User of a Database (scott, James, Sales)

Graphical Representation of the data structure.

What are the most important features of a data warehouse?

DRILL DOWN, DRILL ACROSS, Graphs, PI charts, dashboards and TIME HANDLING

To be able to drill down/drill across is the most basic requirement of an end user in a data warehouse.

Drill down is digging for more granular detail, if a sales report is generated for a country and you want to look at how southern region has performed then you drill 1 level down (Region level) to see the performance of that region.

What does it mean by grain of the star schema?

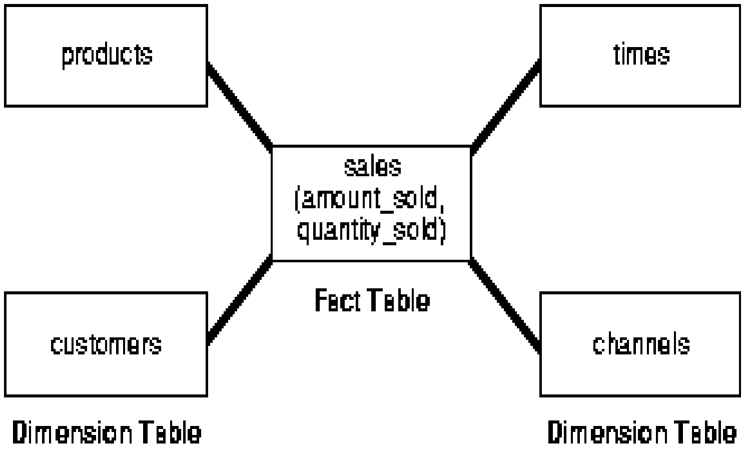
In Data warehousing grain refers to the level of detail available in a given fact table as well as to the level of detail provided by a star schema.

It is usually given as the number of records per key within the table. In general, the grain of the fact table is the grain of the star schema.

What is a star schema?

Star schema is a data warehouse schema where there is only one "fact table" and many de-normalized dimension tables.

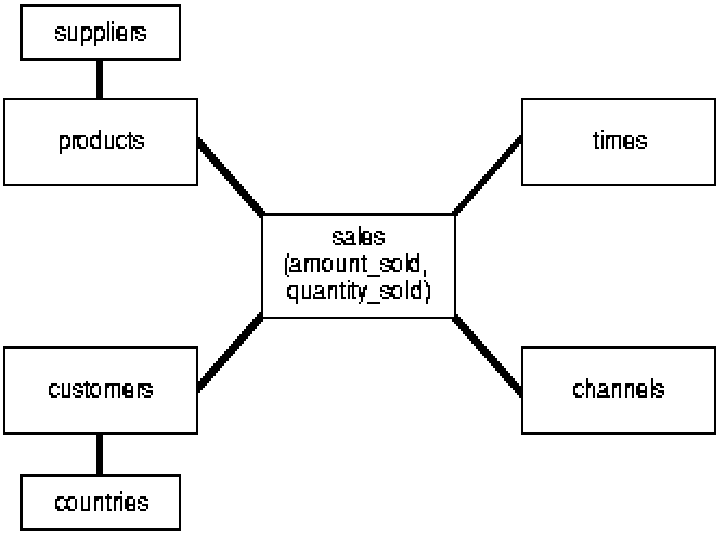
Fact table contains primary keys from all the dimension tables and other statistical columns (facts)



What is a snowflake schema?

Unlike Star-Schema, Snowflake schema contain normalized dimension tables in a tree like structure with many nesting levels.

Snowflake schema is easier to maintain but queries require more joins.



What is the difference between snow flake and star schema?

Star Schema	**Snow Flake Schema**
The star schema is the simplest data warehouse scheme.	Snowflake schema is a more complex data warehouse model than a star schema.
In star schema each of the dimensions is represented in a single table. It should not have any hierarchies between dims.	In snow flake schema at least one hierarchy should exist between dimension tables.
It contains a fact table surrounded by dimension tables. If the dimensions are de-normalized, we say it is a star schema design.	It contains a fact table surrounded by dimension tables. If a dimension is normalized, we say it is a snow flaked design.
In star schema only one join establishes the relationship between the fact table and any one of the dimension tables.	In snow flake schema since there is relationship between the dimension tables it has to do many joins to fetch the data.
A star schema optimizes the performance by keeping queries simple and providing fast response time. All the information about each level is stored in one row.	Snowflake schemas normalize dimensions to eliminated redundancy. The result is more complex queries and reduced query performance.
It is called a star schema because the diagram resembles a star.	It is called a snowflake schema because the diagram resembles a snowflake.

What is Fact and Dimension?

A "fact" is a numeric value that a business wishes to count or sum. A "dimension" is essentially an entry point for getting at the facts. Dimensions are things of interest to the business.

A set of level properties that describe a specific aspect of a business, used for analyzing the factual measures.

What is Fact Table?

A Fact Table in a dimensional model consists of one or more numeric facts of importance to a business. Examples of facts are as follows:

- the number of products sold
- the value of products sold
- the number of products produced
- the number of service calls received

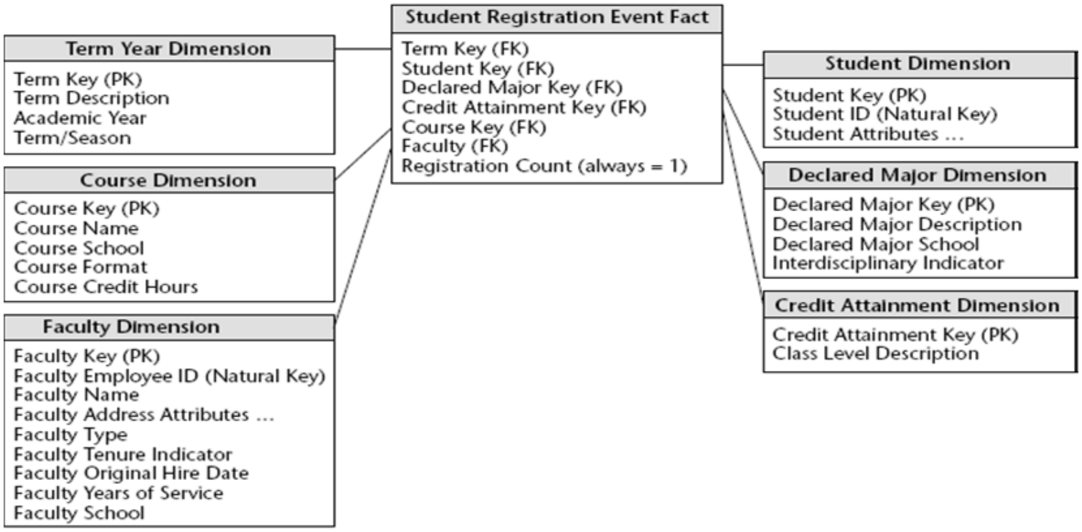
What is Fact Less Fact Table?

Fact less fact table captures the many-to-many relationships between dimensions, but contains no numeric or textual facts. They are often used to record events or coverage information.

Common examples of fact less fact tables include:

- Identifying product promotion events (to determine promoted products that didn't sell)
- Tracking student attendance or registration events
- Tracking insurance-related accident events

- Fact without any measured fact
- Event tracking and coverage Factless fact tables



Term Year Dimension
Term Key (PK)
Term Description
Academic Year
Term/Season

Course Dimension
Course Key (PK)
Course Name
Course School
Course Format
Course Credit Hours

Faculty Dimension
Faculty Key (PK)
Faculty Employee ID (Natural Key)
Faculty Name
Faculty Address Attributes ...
Faculty Type
Faculty Tenure Indicator
Faculty Original Hire Date
Faculty Years of Service
Faculty School

Student Registration Event Fact
Term Key (FK)
Student Key (FK)
Declared Major Key (FK)
Credit Attainment Key (FK)
Course Key (FK)
Faculty (FK)
Registration Count (always = 1)

Student Dimension
Student Key (PK)
Student ID (Natural Key)
Student Attributes ...

Declared Major Dimension
Declared Major Key (PK)
Declared Major Description
Declared Major School
Interdisciplinary Indicator

Credit Attainment Dimension
Credit Attainment Key (PK)
Class Level Description

Different types of facts?

There are three types of facts:

- **Additive**: Additive facts are facts that can be summed up through all of the dimensions in the fact table (Quantity Sold, Dollars Sold)
- **Semi-Additive**: Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others (Account Balance, Inventory levels)
- **Non-Additive**: Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table (Room Temperature)

What is Granularity?

Principle: create fact tables with the most granular data possible to support analysis of the business process.

In Data warehousing grain refers to the level of detail available in a given fact table as well as to the level of detail provided by a star schema.

It is usually given as the number of records per key within the table. In general, the grain of the fact table is the grain of the star schema.

Facts: Facts must be consistent with the grain. All facts are at a uniform grain.

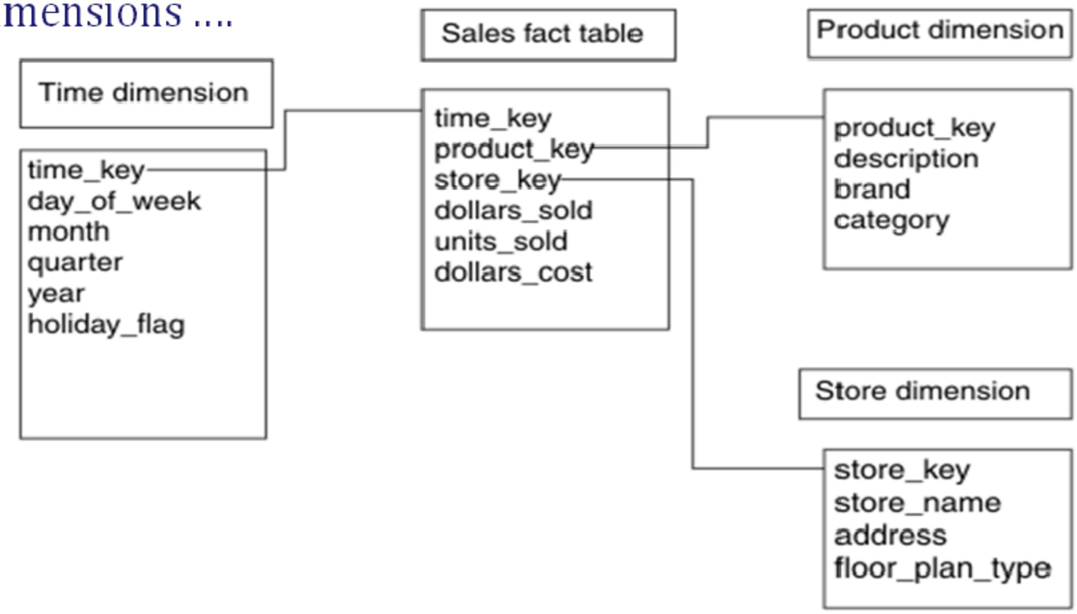
- Watch for facts of mixed granularity
- Total sales for day & monthly total

Dimensions: each dimension associated with fact table must take on a single value for each fact row.

- Each dimension attribute must take on one value.
- Outriggers are the exception, not the rule.

Dimensional Model

A Dimensional Model of a Business : Time, Product, and Store Dimensions ....



Time dimension

time_key
day_of_week
month
quarter
year
holiday_flag

Sales fact table

time_key
product_key
store_key
dollars_sold
units_sold
dollars_cost

Product dimension

product_key
description
brand
category

Store dimension

store_key
store_name
address
floor_plan_type

What is slowly Changing Dimension?

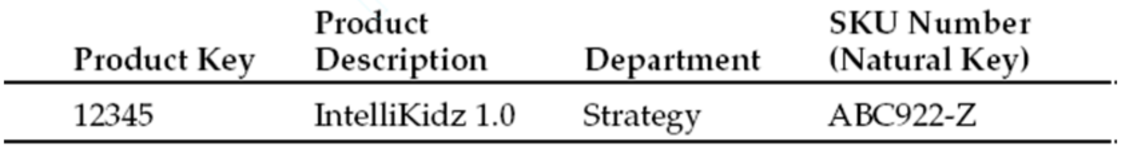
Slowly changing dimensions refers to the change in dimensional attributes over time.

An example of slowly changing dimension is a Resource dimension where attributes of a particular employee change over time like their designation changes or dept changes etc.

- If any attribute changes over a period of time
- Three basic responses:
  - Type I – Overwrite/Update with no history
  - Type II – Add a dimension row
  - Type III – Add a dimension column

Type-I

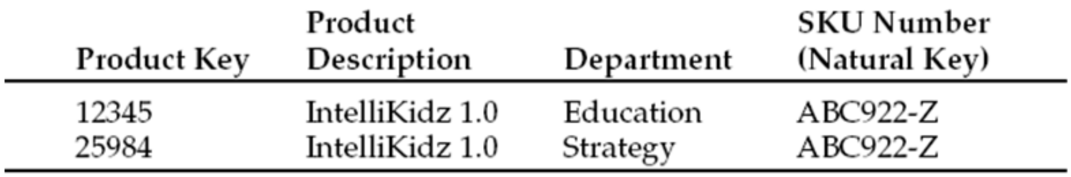
- ❖ Update the existing row
- ❖ No history tracking
- ❖ Fast and Easy



Product Key	Product Description	Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Strategy	ABC922-Z

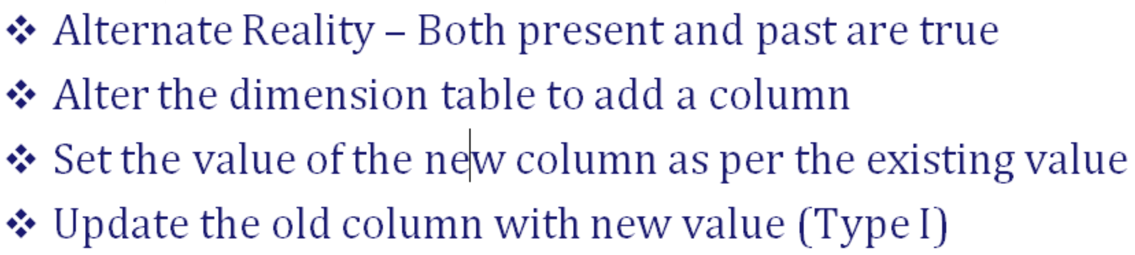
Type-II

- ❖ Insert a new dimension row
- ❖ Maintain history
- ❖ Dimension table grows

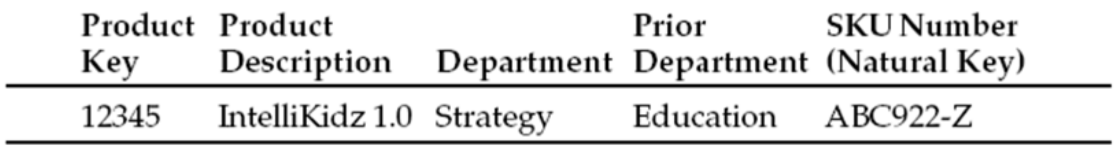


Product Key	Product Description	Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Education	ABC922-Z
25984	IntelliKidz 1.0	Strategy	ABC922-Z

Type-III



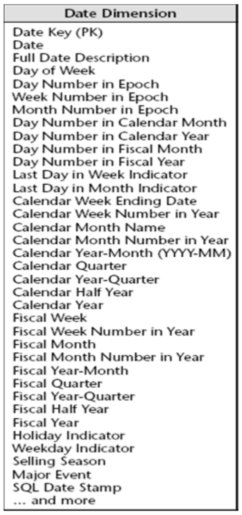
- ❖ Alternate Reality – Both present and past are true
- ❖ Alter the dimension table to add a column
- ❖ Set the value of the new column as per the existing value
- ❖ Update the old column with new value (Type I)



Product Key	Product Description	Department	Prior Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Strategy	Education	ABC922-Z

What is Conformed Dimension?

Conformed Dimensions (CD): these dimensions are something that is built once in your model and can be reused multiple times with different fact tables. For example, consider a model containing multiple fact tables, representing different data marts. Now look for a dimension that is common to these facts tables. In this example let's consider that the product dimension is common and hence can be reused by creating short cuts and joining the different fact tables. Best example is time dimension



Date Dimension
Date Key (PK)
Date
Full Date Description
Day of Week
Day Number in Epoch
Week Number in Epoch
Month Number in Epoch
Day Number in Calendar Month
Day Number in Calendar Year
Day Number in Fiscal Month
Day Number in Fiscal Year
Last Day in Week Indicator
Last Day in Month Indicator
Calendar Week Ending Date
Calendar Week Number in Year
Calendar Month Name
Calendar Month Number in Year
Calendar Year-Month (YYYY-MM)
Calendar Quarter
Calendar Year-Quarter
Calendar Half Year
Calendar Year
Fiscal Week
Fiscal Week Number in Year
Fiscal Month
Fiscal Month Number in Year
Fiscal Year-Month
Fiscal Quarter
Fiscal Year-Quarter
Fiscal Half Year
Fiscal Year
Holiday Indicator
Weekday Indicator
Selling Season
Major Event
SQL Date Stamp
... and more

What is Junk Dimension?

A "junk" dimension is a collection of random transactional codes, flags and/or text attributes that are unrelated to any particular dimension. The junk dimension is simply a structure that provides a convenient place to store the junk attributes. A good example would be a trade fact in a company that brokers equity trades.

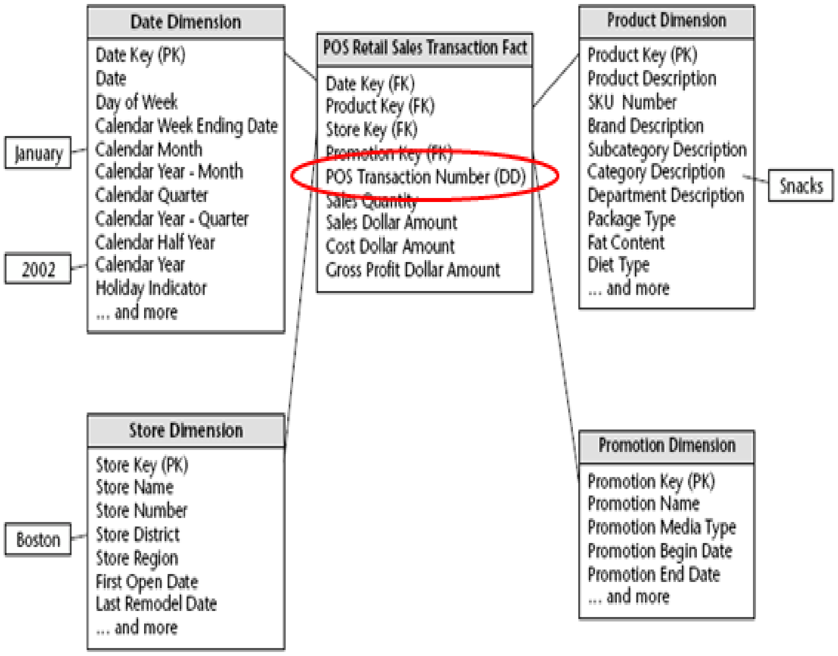
When you consolidate lots of small dimensions and instead of having 100s of small dimensions, that will have few records in them, cluttering your database with these mini 'identifier' tables, all records from all these small dimension tables are loaded into ONE-dimension table and we call this dimension table Junk dimension table. (Since we are storing all the junk in this one table) For example: a company might have handful of manufacture plants, handful of order types, and so on, so forth, and we can consolidate them in one-dimension table called junked dimension table

It's a dimension table which is used to keep junk attributes.

What is De Generated Dimension?

Degenerated Dimension: a dimension which is located in fact table known as Degenerated dimension

- Operational control numbers such as order numbers, invoice numbers usually give rise to empty dimensions
- Dimension keys without corresponding dimension tables



Date Dimension
Date Key (PK)
Date
Day of Week
Calendar Week Ending Date
Calendar Month
Calendar Year - Month
Calendar Quarter
Calendar Year - Quarter
Calendar Half Year
Calendar Year
Holiday Indicator
... and more

January

2002

POS Retail Sales Transaction Fact
Date Key (FK)
Product Key (FK)
Store Key (FK)
Promotion Key (FK)
POS Transaction Number (DD)
Sales Quantity
Sales Dollar Amount
Cost Dollar Amount
Gross Profit Dollar Amount

Product Dimension
Product Key (PK)
Product Description
SKU Number
Brand Description
Subcategory Description
Category Description
Department Description
Package Type
Fat Content
Diet Type
... and more

Snacks

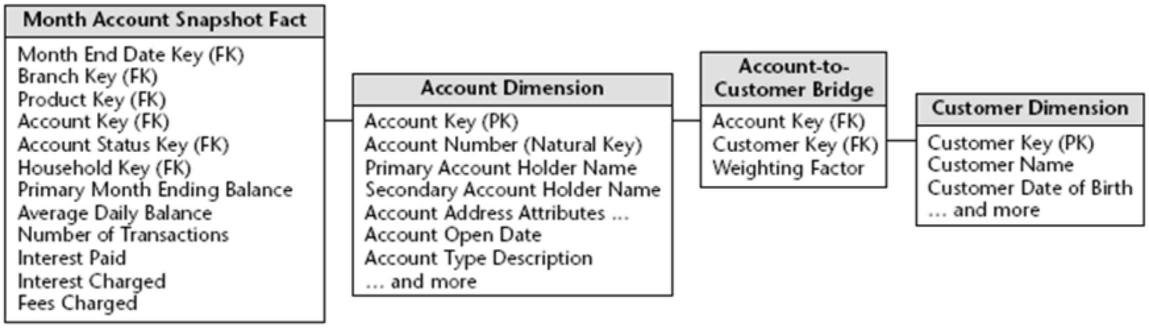
Store Dimension
Store Key (PK)
Store Name
Store Number
Store District
Store Region
First Open Date
Last Remodel Date
... and more

Boston

Promotion Dimension
Promotion Key (PK)
Promotion Name
Promotion Media Type
Promotion Begin Date
Promotion End Date
... and more

What is Multi-Valued Dimension?

- ❖ Many – to – Many Relationship
- ❖ Bridge table is used



Month Account Snapshot Fact
Month End Date Key (FK)
Branch Key (FK)
Product Key (FK)
Account Key (FK)
Account Status Key (FK)
Household Key (FK)
Primary Month Ending Balance
Average Daily Balance
Number of Transactions
Interest Paid
Interest Charged
Fees Charged

Account Dimension
Account Key (PK)
Account Number (Natural Key)
Primary Account Holder Name
Secondary Account Holder Name
Account Address Attributes ...
Account Open Date
Account Type Description
... and more

Account-to-Customer Bridge
Account Key (FK)
Customer Key (FK)
Weighting Factor

Customer Dimension
Customer Key (PK)
Customer Name
Customer Date of Birth
... and more

Dimensional Model:

A type of data modeling suited for data warehousing. In a dimensional model, there are two types of tables: dimensional tables and fact tables. Dimensional table records information on each dimension, and fact table records all the "fact", or measures.

Data modeling

There are three levels of data modeling. They are conceptual, logical, and physical. This section will explain the difference among the three, the order with which each one is created, and how to go from one level to the other.

Conceptual Data Model

Features of conceptual data model include:

- Includes the important entities and the relationships among them.
- No attribute is specified.

- No primary key is specified.

At this level, the data modeler attempts to identify the highest-level relationships among the different entities.

Logical Data Model

Features of logical data model include:

- Includes all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity specified.
- Foreign keys (keys identifying the relationship between different entities) are specified.
- Normalization occurs at this level.

At this level, the data modeler attempts to describe the data in as much detail as possible, without regard to how they will be physically implemented in the database.

In data warehousing, it is common for the conceptual data model and the logical data model to be combined into a single step (deliverable).

The steps for designing the logical data model are as follows:

1. Identify all entities.
2. Specify primary keys for all entities.
3. Find the relationships between different entities.
4. Find all attributes for each entity.
5. Resolve many-to-many relationships.
6. Normalization.

Physical Data Model

Features of physical data model include:

- Specification all tables and columns.
- Foreign keys are used to identify relationships between tables.
- Demoralization may occur based on user requirements.
- Physical considerations may cause the physical data model to be quite different from the logical data model.

At this level, the data modeler will specify how the logical data model will be realized in the database schema.

The steps for physical data model design are as follows:

1. Convert entities into tables.
2. Convert relationships into foreign keys.
3. Convert attributes into columns.

   1. http://www.learndatamodeling.com/dm_standard.htm
   2. Modeling is an efficient and effective way to represent the organization's needs; It provides information in a graphical way to the members of an organization to understand and communicate the business rules and processes. Business Modeling and Data Modeling are the two important types of modeling.

The differences between a logical data model and physical data model is shown below.

Logical vs Physical Data Modeling

Logical Data Model	**Physical Data Model**
Represents business information and defines	Represents the physical implementation of the model in a

business rules	database.
Entity	Table
Attribute	Column
Primary Key	Primary Key Constraint
Alternate Key	Unique Constraint or Unique Index
Inversion Key Entry	Non Unique Index
Rule	Check Constraint, Default Value
Relationship	Foreign Key
Definition	Comment

Types of SCD Implementation:

Type 1 Slowly Changing Dimension

In Type 1 Slowly Changing Dimension, the new information simply overwrites the original information. In other words, no history is kept.

In our example, recall we originally have the following table:

Customer Key	Name	State
1001	Christina	Illinois

After Christina moved from Illinois to California, the new information replaces the new record, and we have the following table:

Customer Key	Name	State
1001	Christina	California

Advantages:

- This is the easiest way to handle the Slowly Changing Dimension problem, since there is no need to keep track of the old information.

Disadvantages:

- All history is lost. By applying this methodology, it is not possible to trace back in history. For example, in this case, the company would not be able to know that Christina lived in Illinois before.
- Usage:

About 50% of the time.

When to use Type 1:

Type 1 slowly changing dimension should be used when it is not necessary for the data warehouse to keep track of historical changes.

Type 2 Slowly Changing Dimension

In Type 2 Slowly Changing Dimension, a new record is added to the table to represent the new information. Therefore, both the original and the new record will be present. The newe record gets its own primary key.

In our example, recall we originally have the following table:

Customer Key	Name	State
1001	Christina	Illinois

After Christina moved from Illinois to California, we add the new information as a new row into the table:

Customer Key	Name	State
1001	Christina	Illinois
1005	Christina	California

Advantages:

- This allows us to accurately keep all historical information.

Disadvantages:

- This will cause the size of the table to grow fast. In cases where the number of rows for the table is very high to start with, storage and performance can become a concern.

- This necessarily complicates the ETL process.

Usage:

About 50% of the time.

When to use Type 2:

Type 2 slowly changing dimension should be used when it is necessary for the data warehouse to track historical changes.

Type 3 Slowly Changing Dimension

In Type 3 Slowly Changing Dimension, there will be two columns to indicate the particular attribute of interest, one indicating the original value, and one indicating the current value. There will also be a column that indicates when the current value becomes active.

In our example, recall we originally have the following table:

Customer Key	Name	State
1001	Christina	Illinois

To accommodate Type 3 Slowly Changing Dimension, we will now have the following columns:

- Customer Key
- Name
- Original State
- Current State
- Effective Date

After Christina moved from Illinois to California, the original information gets updated, and we have the following table (assuming the effective date of change is January 15, 2003):

Customer Key	Name	Original State	Current State	Effective Date
1001	Christina	Illinois	California	15-JAN-2003

Advantages:

- This does not increase the size of the table, since new information is updated.

- This allows us to keep some part of history.

Disadvantages:

- Type 3 will not be able to keep all history where an attribute is changed more than once. For example, if Christina later moves to Texas on December 15, 2003, the California information will be lost.

Usage:

Type 3 is rarely used in actual practice.

When to use Type 3:

Type III slowly changing dimension should only be used when it is necessary for the data warehouse to track historical changes, and when such changes will only occur for a finite number of time.

What is Staging area why we need it in DWH?

If target and source databases are different and target table volume is high, it contains some millions of records in this scenario without staging table we need to design your Informatica using look up to find out whether the record exists or not in the target table since target has huge volumes so its costly to create cache it will hit the performance.

If we create staging tables in the target database, we can simply do outer join in the source qualifier to determine insert/update this approach will give you good performance.

It will avoid full table scan to determine insert/updates on target. And also we can create index on staging tables since these tables were designed for specific application it will not impact to any other schemas/users.

While processing flat files to data warehousing we can perform cleansing.
Data cleansing, also known as data scrubbing, is the process of ensuring that a set of data is correct and accurate. During data cleansing, records are checked for accuracy and consistency.

- Since it is one-to-one mapping from ODS to staging we do truncate and reload.
- We can create indexes in the staging state, to perform our source qualifier best.
- If we have the staging area no need to relay on the informatics transformation to known whether the record exists or not.

Data cleansing

Weeding out unnecessary or unwanted things (characters and spaces etc) from incoming data to make it more meaningful and informative

Data merging

Data can be gathered from heterogeneous systems and put together

Data scrubbing

Data scrubbing is the process of fixing or eliminating individual pieces of data that are incorrect, incomplete or duplicated before the data is passed to end user.

Data scrubbing is aimed at more than eliminating errors and redundancy. The goal is also to bring consistency to various data sets that may have been created with different, incompatible business rules.

ODS (Operational Data Sources):

My understanding of ODS is, its a replica of OLTP system and so the need of this, is to reduce the burden on production system (OLTP) while fetching data for loading targets. Hence its a mandate Requirement for every Warehouse.

So every day do we transfer data to ODS from OLTP to keep it up to date?

OLTP is a sensitive database they should not allow multiple select statements it may impact the performance as well as if something goes wrong while fetching data from OLTP to data warehouse it will directly impact the business.

ODS is the replication of OLTP.

ODS is usually getting refreshed through some oracle jobs.

enables management to gain a consistent picture of the business.

What is a surrogate key?

A surrogate key is a substitution for the natural primary key. It is a unique identifier or number (normally created by a database sequence generator) for each record of a dimension table that can be used for the primary key to the table.

A surrogate key is useful because natural keys may change.

What is the difference between a primary key and a surrogate key?

A **primary key** is a special constraint on a column or set of columns. A primary key constraint ensures that the column(s) so designated have no NULL values, and that every value is unique. Physically, a primary key is implemented by the database system using a unique index, and all the columns in the primary key must have been declared NOT NULL. A table may have only one primary key, but it may be composite (consist of more than one column).

A **surrogate key** is any column or set of columns that can be declared as the primary key instead of a "real" or natural key. Sometimes there can be several natural keys that could be declared as the primary key, and these are all called candidate keys. So a surrogate is a candidate key. A table could actually have more than one surrogate key, although this would be unusual. The most common type of surrogate key is an incrementing integer, such as an auto increment column in MySQL, or a sequence in Oracle, or an identity column in SQL Server.