Fact Tables:

type of fact tables

fact-less fact table

fact table

dimension tables:

type of dimension tables

Conformed dimensions:In data warehousing, a conformed dimension is a dimension that has the same meaning to every fact with which it relates. Conformed dimensions allow facts and measures to be categorized and described in the same way across multiple facts and/or data marts, ensuring consistent reporting across the enterprise.

lined dimensions: these dimension are available in another database server and these are readonly dimension. if some thing is changed then the relationship has to be recreaded

Junk Dimension:

Role Playing Dimension: date dim

Slowly Changing Dimension (SCD): type 1 to 4

Degenerated Dimension:

----------------------------------------------------------

DBMS | Keys in Relational Model (Candidate, Super, Primary, Alternate and Foreign)

----------------------------------------------------------

Super Key:

Definition: A super key is a set or one of more columns (attributes) to uniquely identify rows in a table. Often people get confused between super key and candidate key, so we will also discuss a little about candidate key here.

Definition - What does Superkey mean?

A superkey is a combination of columns that uniquely identifies any row within a relational database management system (RDBMS) table. A candidate key is a closely related concept where the superkey is reduced to the minimum number of columns required to uniquely identify each row.

Techopedia explains Superkey

As an example, a table used to store customer master details could contain columns such as:

Customer name

Customer ID

Social Security number (SSN)

Address

Date of birth

A certain set of columns may be extracted and guaranteed unique to each customer. Examples of superkeys are as follows:

Name+SSN+Birthdate

ID+Name+SSN

However, this process may be further reduced. It can be assumed that the customer ID is unique to each customer. Therefore, the superkey may be reduced to just one field, customer ID, which is the candidate key. However, to ensure absolute uniqueness, a composite candidate key may be formed by combining customer ID with SSN.

Super Key: The set of attributes which can uniquely identify a tuple is known as Super Key. For Example, STUD\_NO, (STUD\_NO, STUD\_NAME) etc.

Adding zero or more attributes to candidate key generates super key.

A candidate key is a super key but vice versa is not true.

----------------------------------------------------------

Candidate Key:

The minimal set of attribute which can uniquely identify a tuple is known as candidate key. For Example, STUD\_NO in STUDENT relation.

The value of Candidate Key is unique and non-null for every tuple.

There can be more than one candidate key in a relation. For Example, STUD\_NO as well as STUD\_PHONE both are candidate keys for relation STUDENT.

The candidate key can be simple (having only one attribute) or composite as well. For Example, {STUD\_NO, COURSE\_NO} is a composite candidate key for relation STUDENT\_COURSE.

Note – In Sql Server a unique constraint that has a nullable column, allows the value ‘null‘ in that column only once. That’s why STUD\_PHONE attribute as candidate here, but can not be ‘null’ values in primary key attribute.

----------------------------------------------------------

How candidate key is different from super key?

Answer is simple – Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is: It should not have any redundant attribute. That’s the reason they are also termed as minimal super key.

Super key vs Candidate Key

1. All the candidate keys are super keys. This is because the candidate keys are chosen out of the super keys.

2. How we choose candidate keys from the set of super keys?

We look for those keys from which we cannot remove any fields.

In the above example, we have not chosen {Emp\_SSN, Emp\_Name} as candidate key because {Emp\_SSN} alone can identify a unique row in the table and Emp\_Name is redundant.

----------------------------------------------------------

Primary key:

Primary key is being selected from the sets of candidate keys by database designer. So Either {Emp\_SSN} or {Emp\_Number} can be the primary key.

Primary Key: There can be more than one candidate key in a relation out of which one can be chosen as primary key. For Example, STUD\_NO as well as STUD\_PHONE both are candidate keys for relation STUDENT but STUD\_NO can be chosen as primary key (only one out of many candidate keys).

Alternate Key: The candidate key other than primary key is called as alternate key. For Example, STUD\_NO as well as STUD\_PHONE both are candidate keys for relation STUDENT but STUD\_PHONE will be alternate key (only one out of many candidate keys).

Foreign Key: If an attribute can only take the values which are present as values of some other attribute, it will be foreign key to the attribute to which it refers. The relation which is being referenced is called referenced relation and corresponding attribute is called referenced attribute and the relation which refers to referenced relation is called referencing relation and corresponding attribute is called referencing attribute. Referenced attribute of referencing attribute should be primary key. For Example, STUD\_NO in STUDENT\_COURSE is a foreign key to STUD\_NO in STUDENT relation.

----------------------------------------------------------

fact table:

All the dimension tables are connected to the fact table. The primary key in all dimension tables acts as a foreign key in the fact table.

The star schema is quite simple, flexible and it is in de-normalized form.

In a snowflake schema, the level of normalization increases. The fact table here remains the same as in star schema.

However, the dimension tables are normalized. Due to several layers of dimension tables, it looks like a snowflake and thus it is named as snowflake schema.

The choice of a schema always depends upon the project requirements & scenarios.

Since star schema is in de-normalized form, you require fewer joins for a query. The query is simple and runs faster in a star schema. Coming to the snowflake schema, since it is in normalized form, it will require a number of joins as compared to a star schema, the query will be complex and execution will be slower than star schema.

Another significant difference between these two schemas is that snowflake schema does not contain redundant data and thus it is easy to maintain. On the contrary, star schema has a high level of redundancy and thus it is difficult to maintain.

In my project, we used snowflake schema because we had to do analysis across several dimensions and generate summary reports for the business. Another reason for using snowflake schema was it is less memory consumption.

The choice of a schema always depends upon the project requirements & scenarios.

Since star schema is in de-normalized form, you require fewer joins for a query. The query is simple and runs faster in a star schema. Coming to the snowflake schema, since it is in normalized form, it will require a number of joins as compared to a star schema, the query will be complex and execution will be slower than star schema.

Another significant difference between these two schemas is that snowflake schema does not contain redundant data and thus it is easy to maintain. On the contrary, star schema has a high level of redundancy and thus it is difficult to maintain.

Now, which one to choose for your project? If the purpose of your project is to do more of dimension analysis, you should go for snowflake schema. For example, if you need to find out that “how many subscribers are tied to a particular plan are currently active?” – go with snowflake model.

If the purpose of your project is to do more of metrics analysis, you should go with a star schema. For example, if you need to find out that “what is the claim amount paid to a particular subscriber?” – go with a star schema.

In my project, we used snowflake schema because we had to do analysis across several dimensions and generate summary reports for the business. Another reason for using snowflake schema was it is less memory consumption.

Dimensions represent qualitative data. For example– plan, product, class are all dimensions. A dimension table contains descriptive or textual attributes. For example, product category & product name are the attributes of product dimension.

Facts represent quantitative data. For example – net amount due is a fact. A fact table contains numerical data and foreign keys from related dimensional tables. An example of the fact table can be seen from Figure 2 shown above.

There are typically five types of dimensions.

Conformed dimensions:

Junk Dimension

Role Playing Dimension: date dim

Slowly Changing Dimension (SCD): type 1 to 4

Degenerated Dimension:

Slowly Changing Dimension (SCD): These are most important amongst all the dimensions. These are the dimensions where attribute values vary with time. Below are the varies types of SCDs

Factless fact table is a fact table that contains no fact measure in it. It has only the dimension keys in it.

We have three types of measures

Non-additive measures

Semi-additive measures

Additive measures

Non-additive measures are the ones on top of which no aggregation function can be applied. For example, a ratio or a percentage column; a flag or an indicator column present in fact table holding values like Y/N, etc. is a non-additive measure.

Semi-additive measures are the ones on top of which some (but not all) aggregation functions can be applied. Example – fee rate or account balance.

Additive measures are the ones on top of which all aggregation functions cab be applied. Example- units purchased.

Surrogate key is a unique identifier or a system generated sequence number key that can act as a primary key. It can be a column or a combination of columns. Unlike a primary key, it is not picked up from the existing application data fields.

recursive relationship

Metadata is data about data. It tells you that what kind of data is actually stored in the system, what is its purpose and for whom it is intended for.

Hierarchical database model

Relational model

Network model

Object-oriented database model

Entity-relationship model

Document model

Entity-attribute-value model

Star schema

The object-relational model, which combines the two that make up its name

Structured Query Language (SQL). in 1970

What does data model contain?

Logical Data Model: Entity, Attributes, Super Type, Sub Type, Primary Key, Alternate Key, Inversion Key Entry, Rule, Relationship, Definition, business rule, etc

Physical Data Model: Table, Column, Primary key Constraint, Unique Constraint or Unique Index, Non Unique Index, Check Constraint, Default Value, Foreign Key, comment etc.

table (entity)

column (attribute)

Q. What is a row?

Row also known as tuple or record is the horizontal alignment of the data.

What is a composite primary key constraint?

When more than one column is a part of the primary key, it is called as composite primary key constraint.

What is a composite foreign key constraint?

When group of columns are in a foreign key, it is called as composite foreign key constraint.

One to One, One to many, and many to many are different types of cardinalities.

In a database, high cardinality means more unique values are stored in a column and vice versa

How many null values can be inserted in a column that has unique constraint?

Many null values can be inserted in an unique constraint column because one null value is not equal to another null value.

Basel Committee of Banking Supervision (BCBS)

Bill Inmon and Ralph Kimball.

“Bill Inmon” is sometimes also referred to as the “father of data warehousing”; his design methodology is based on a top-down approach. In the top-down approach, the data warehouse is designed first and then data mart are built on top of data warehouse.

Ralph Kimball is a renowned author on the subject of data warehousing. His data warehouse design approach is called dimensional modelling or the Kimball methodology. This methodology follows the bottom-up approach.

Basically, Kimball model reverses the Inmon model i.e. Data marts are directly loaded with the data from the source systems and then ETL process is used to load in to Data Warehouse

hierarchies:

A tree shows a hierarchical relationship. In a relational database, the different levels of a hierarchy can be stored in a single table (as in a star schema) or in separate tables (as in a snowflake schema). Each of the elements of a dimension could be summarized using a hierarchy.

A ragged hierarchy is a user-defined hierarchy that has an uneven number of levels. ... A ragged hierarchy is different in that the logical parent of at least one member is not in the level immediately above the member. When this occurs, the hierarchy descends to different levels for different drill down paths.

The hierarchy branches can also have inconsistent depths. An example of an unbalanced hierarchy is an organization chart, which shows reporting relationships among employees in an organization. ... The parent-child relationships on both branches of the hierarchy are consistent.

In balanced hierarchies, the branches of the hierarchy all descend to the same level, with each member's parent being at the level immediately above the member. A balanced hierarchy (balanced/standard) can represent time where the meaning and depth of each level, such as Year, Quarter, and Month, is consistent.

You can have multiple hierarchies in a table

e.g. Calendar date hierarchy and Fiscal Date hierarchy in Date dimension.

Hierarchies are useful to define the drill down/drill up navigation path in BI tools for reporting purpose

There are three main types of hierarchies in a general dimensional data warehouse design.

Balanced Hierarchy.

Unbalanced Hierarchy.

Ragged Hierarchy.

---------------------------------------------------------------------------------------------

1st nf -- Contains only atomic values. There are no repeating groups

2nd nf -- no partial dependencies

All non-key attributes are fully functional dependent on the primary key

3rd nf -- transitive dependencies

There is no transitive functional dependency

4th nf --

by setting KPIs the company enables the team to make smart business decisions about the direction of all current projects

The columns are known as attributes whereas the rows are known as records.

DBMS | Relational Model Introduction and Codd Rules

-------------------------------------------------------------------------------------------

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

It should only have single(atomic) valued attributes/columns.

Values stored in a column should be of the same domain

All the columns in a table should have unique names.

And the order in which data is stored, does not matter.

In the next tutorial, we will discuss about the First Normal Form in details.

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

It should be in the First Normal form.

And, it should not have Partial Dependency.

To understand what is Partial Dependency and how to normalize a table to 2nd normal for, jump to the Second Normal Form tutorial.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

It is in the Second Normal form.

And, it doesn't have Transitive Dependency.

Here is the Third Normal Form tutorial. But we suggest you to first study about the second normal form and then head over to the third normal form.

Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

R must be in 3rd Normal Form

and, for each functional dependency ( X → Y ), X should be a super Key.

To learn about BCNF in detail with a very easy to understand example, head to Boye-Codd Normal Form tutorial.

Fourth Normal Form (4NF)

A table is said to be in the Fourth Normal Form when,

It is in the Boyce-Codd Normal Form.

And, it doesn't have Multi-Valued Dependency.

Here is the Fourth Normal Form tutorial. But we suggest you to understand other normal forms before you head over to the fourth normal form.

5NF (Fifth Normal Form) Rules

A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed into any number of smaller tables without loss of data.