



Module 1

Fact Tables and Dimension Tables

A reality or fact table's record could be a combination of attributes from totally different dimension tables. The **Fact Table or Reality Table** helps the user to investigate the business dimensions that helps him in call taking to enhance his business.

On the opposite hand, **Dimension Tables** facilitate the reality table or fact table to gather dimensions on that the measures needs to be taken.

The main difference between fact table or reality table and the Dimension table is that dimension table contains attributes on that measures are taken actually table.

Dimension Table

Product
Product_Key
Product_Name
Brand
Colour

Fact Table

Order_Measure
Product_Key
Time_Key
Store_Key
Order_Dollars
Order_Quantity

Dimension Table

Time
Time_Key
Date
Day
Month

Store
Store_Key
City
Store_name
Phone_no

Dimension Table

Difference between Fact Table and Dimension Table:



S.NO	Fact Table	Dimension Table
1.	Fact table contains the measuring of the attributes of a dimension table.	Dimension table contains the attributes on that truth table calculates the metric.
2.	In fact table, There is less attributes than dimension table.	While in dimension table, There is more attributes than fact table.
3.	In fact table, There is more records than dimension table.	While in dimension table, There is less records than fact table.
4.	Fact table forms a vertical table.	While dimension table forms a horizontal table.
5.	The attribute format of fact table is in numerical format and text format.	While the attribute format of dimension table is in text format.
6.	It comes after dimension table.	While it comes before fact table.
7.	The number of fact table is less than dimension table in a schema.	While the number of dimension is more than fact table in a schema.
8.	It is used for analysis purpose and decision making.	While the main task of dimension table is to store the information about a business and its process.

Also let us see what Aggregate Fact Tables are,

Factless Fact Table

Factless tables simply mean the key available in the fact that no remedies are available. Factless fact tables are only used to establish relationships between elements of different dimensions. And are also useful for describing events and coverage, meaning tables contain information that nothing has happened. It often represents many-to-many relationships.

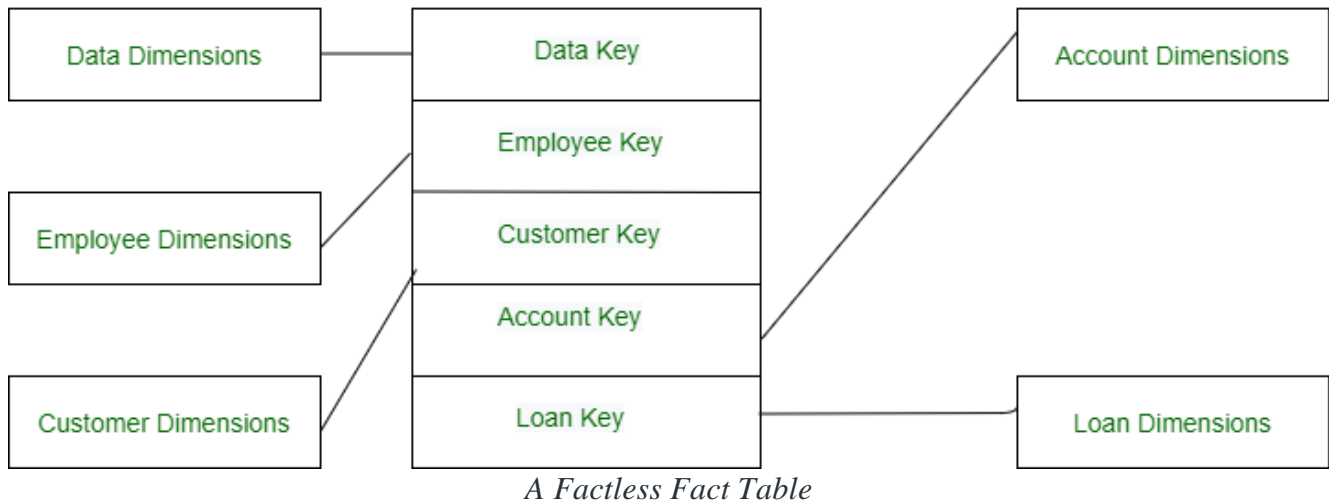
The only thing they have is an abbreviated key. They still represent a focal phenomenon that is identified by the combination referenced in the dimension tables.



Semester : V

Subject :DWM

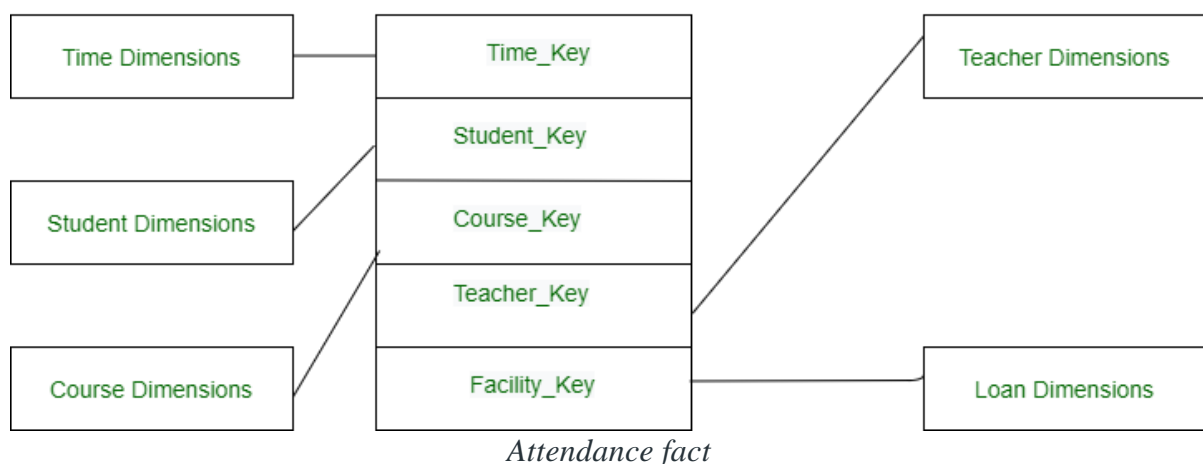
Academic Year: 2023 - 2024



There are two types of factless table :

1. Event Tracking Tables –

Use a factless fact table to track events of interest to the organization. For example, attendance at a cultural event can be tracked by creating a fact table that contains the following foreign keys (i.e. links to dimension tables) event identifier speaker/entertainment identifier, participant identifier, event type; Date. This table can then be searched for information, such as the most popular ones. Which cultural program or program type. The following example shows a factless fact table that records each time a student attends a course or which class has the maximum attendance? Or what is the average number of attendance of a given course? All questions are based on COUNT () with group BY questions. So we can first count and then implement other aggregate functions like Aggress, Max, Min.





2. Coverage Tables –

The second type of factless fact table is called a coverage table by *Ralph*. It is used to support negative analysis reports. For example, to create a report that a store did not sell a product for a certain period of time, you should have a fact table to capture all possible combinations. Then you can find out what is missing.

Common examples of factless fact table:

Ex-Visitors to the office.

List of people for the web click.

Tracking student attendance or registration events.

Update to Dimension Table :

- Every day, more and more sales take place, so more and more rows are added to the fact table.
- Updating due to the change in fact table happens very rarely.
- Dimension tables are more stable as compared to the fact tables.
- Dimension table changes due to the change in attributes themselves, but not because of an increase in the number of rows.

Slowly Changing Dimensions :

- Dimensions are generally constant over time, but if not constant, then they change slowly. The customer ID of the record remains the same but the marital status or location of the customer may change over time.
- In the OLTP system, whenever such a change in attribute values happens, the old values replace the new values by overwriting the old ones.
- But in a data warehouse, overwriting of attributes is not the solution as historical data for analysis is always required.

Large Dimension Tables :

- Large dimension tables are very deep and wide.
- Deep means it has a very large number of rows and wide means it may have many attributes or columns.
- To handle large dimensions, one can take out some mini dimensions from a large dimension as per the interest. These mini-dimensions can be represented in the form of a star schema.
- For example, the above-mentioned order analysis star schema is one of the mini-dimensions of a manufacturing company in which the marketing department of the company is interested.
- Customers and products are generally large in dimensions.
- Large dimensions are generally slow and inefficient due to their size. They tend to have multiple hierarchies to perform various OLAP operations like drill down or roll-up.

Rapidly Changing or Large Slowly Changing Dimensions :

- In type 2 changes, a new row is created with the new value of the changed attribute. This preserves the history or old values of attributes.
- If there is a change again in some attribute, then again a new dimension table row is created within the new value.
- This is feasible if the dimension changes infrequently, like once or twice a year. For example, the product dimension, which has rows in thousands, changes rarely so it is manageable.



Semester : V

Subject :DWM

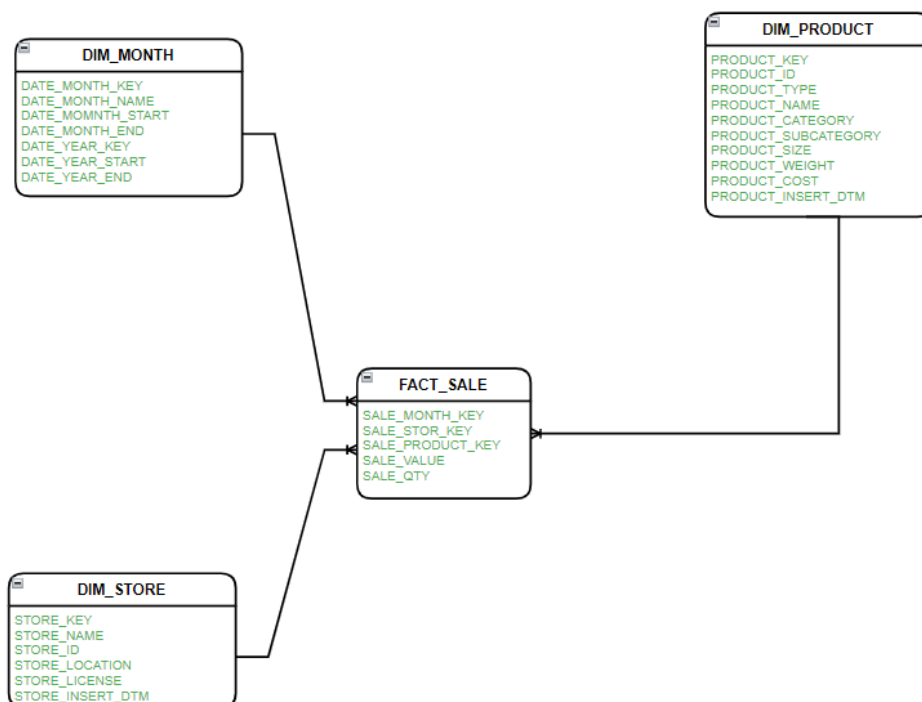
Academic Year: 2023 - 2024

- But in the case of customer dimensions, where a number of rows are millions and changes infrequently, then type 2 changes are feasible and not very difficult. If customer dimensions change rapidly, then Type 2 changes are problematic and difficult.
- If the dimension table is rapidly changing and large, then break that dimension table into one or more smaller dimension tables.

Aggregate Fact Tables:

- Aggregate fact tables are a special kind of fact tables in a data warehouse which contains new metrics which are been derived from one or more aggregate functions (COUNT, AVERAGE, MIN, MAX, etc.) or from some specialized functions whose outputs are totally derived from a grouping of base data.
- Aggregates are basically summarization of the fact related data which are been used as a purpose to improve the performance.
- These new metrics, called as “aggregate facts” or “summary statistics” are been stored and maintained in database of the data warehouse in special fact table at the grain of the aggregation.
- In similar way, the corresponding dimensions are been rolled up and compressed to match the new grain of the fact.
- These specialized tables are been used as an substitutions whenever possible for returning user queries. The reason is the speed.
- Querying a neat aggregate table is much faster and uses less of the disk I/O than the base, atomic fact table, especially when the dimensions are large as well.
- If you want to amaze your users then start adding the aggregates.
- Even you can use this technique in your operational systems as well, giving boost to the foundational reports.

EXAMPLE:





EXAMPLE

Limitations of Aggregate Fact Tables:

- Does not support exploratory analysis.
- Must be reaggregated each and every time when there is been certain change in source data so that the changes can be reflected in the data warehouse.
- The narrow capability leads to low and limited interactive use.

What is a Primary Key?

A primary key is one or more columns in a table that are used to uniquely identify the row.

When you're working with a relational database, you have multiple tables and you need to link them to each other. There needs to be a way to identify records, even if data changes. The way to do this is using a primary key.

It's kind of like how when you speak to your bank, insurance company, or government department, they ask for an ID:

- Banks ask for a customer or account number.
- Insurance companies ask for your member number.
- Government departments ask for your Tax File Number (here in Australia) or perhaps your Social Security Number in the US.

That way, it doesn't matter if you change your name or address, there is always an identifier that can be used to find your information.

Let's look at an example – a bank account.

Say you've gone to a bank, filled out some forms, and created an account with them. You have a customer record with them, that might look like this:

- Customer Number: 2458760357
- First Name: Ben
- Last Name: Brumm
- Address: 123 Main St, Melbourne, VIC, Australia 3000
- Phone: 0412 345 678
- ...

Surrogate Key

What is a Surrogate Key?



The word surrogate means substitute. A surrogate key is an attribute that is invented or made up for the sole purpose of being used as the primary key. It has no value to the business or the real world.

An example of a surrogate key is an “address ID” for a table of addresses. Outside of the system, an address ID has no value to anyone. But for the database, it’s used to uniquely identify the record.

A lot of examples online use this concept for a primary key.

Advantages of a Surrogate Key

There are several advantages using a surrogate key for your primary key:

- It has no business or real world value, which means you have control over the format and usage of the value. No matter what happens to the other data in the record (e.g. change in formats or types), this key won’t change.
- It’s easy to define. There’s no need to work out which of the existing attributes can make a key, as you can use a new column for the identifier.

Disadvantages of a Surrogate Key

There are a few disadvantages of using a surrogate key as your primary key:

- It’s another extra column in your table, which increases storage space. However, storage is cheap so this may not be an issue.
- You can’t look at the key and determine anything about the record, unlike a Social Security Number or a Country Name.

I’ll share my recommendation for primary keys later in this post after we’ve looked at different types of keys.

Composite Key

What is a Composite Key?

A composite key is a primary key, or unique identifier, that is made up of more than one attribute.

The examples we’ve looked at so far mention a single attribute. However, I believe most databases let you use more than one column as a primary key. If you do, this is called a composite key.

An example of this would be identifying an address by combining the street number, street name, city, and postal code. Depending on your database, this may uniquely identify an address. If so, it could be used as a composite key.



These fields could not be used individually to identify an address. Street numbers can be repeated, and so can street names. However, the combination of all of these fields would be unique.

Another example could be the combination of first name, last name, and date of birth. It seems pretty unique, but it may not be (as mentioned below in the disadvantages).

Advantages of a Composite Key

The advantages of using a composite key as a primary key are:

- The attributes already exist, so you don't need to create any new attributes or columns on your table like you do with the surrogate key. This simplifies the table and reduces storage.
- The combination of the attributes ensures the record is unique, like the natural key
- It allows you to use business values to identify a record if a single value does not uniquely identify a record. If a natural key doesn't work, you can use a composite key.
-

Disadvantages of a Composite Key

The disadvantages of using a composite key as a primary key are:

- Risk of change to the business rules. Like a natural key, if any of the business rules change (type of field, format of data) then this will need to be updated. Also, the data may no longer uniquely identify a row so you may need to expand your composite key.
- All fields are needed in related tables. When you want to refer to this record from another table, you'll need to store all of the fields in the composite key in another table.