

Time Reporting

How to work with time
when using stupid
computers

Time Relations

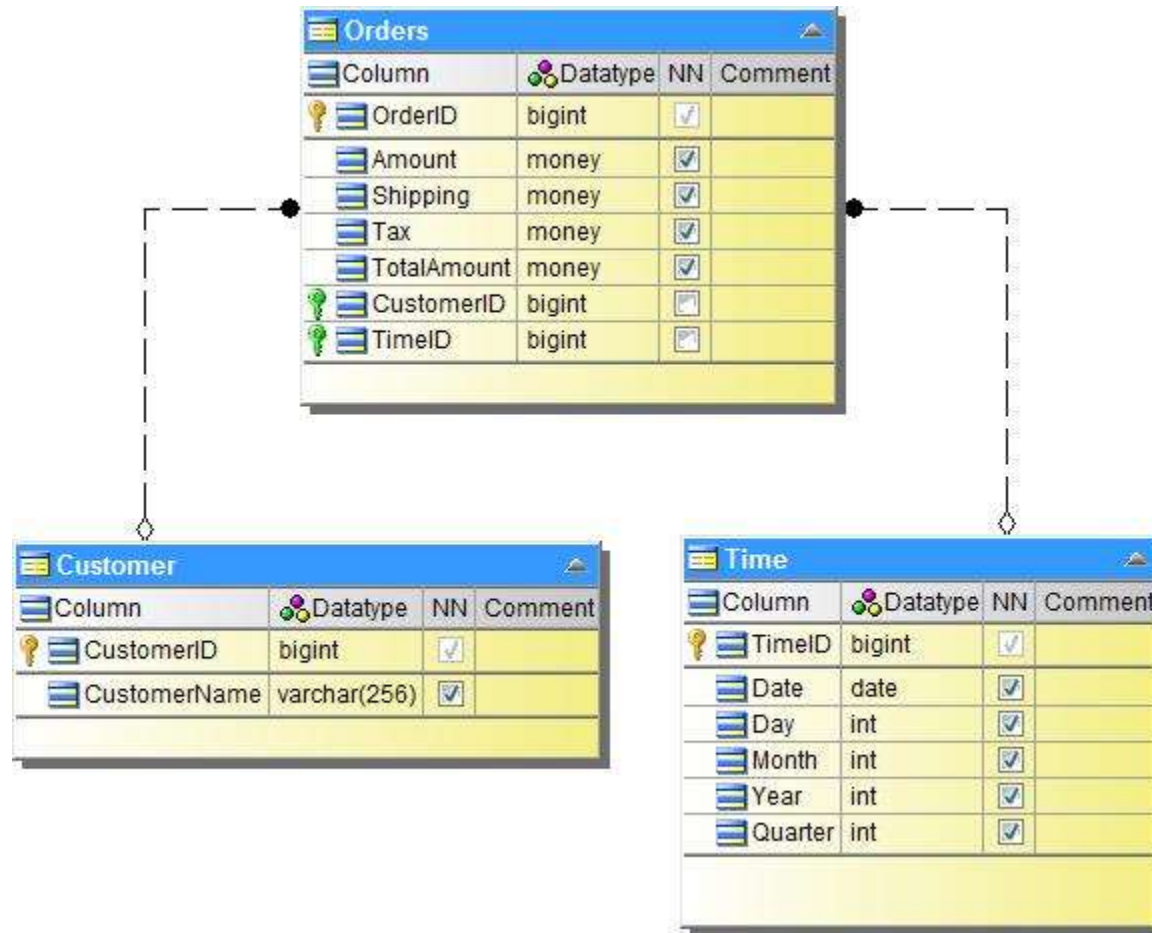
Those who choose to use time as a relationship compares two different time factors based on the smallest component of time relevant to their needs. This is the typical implementation of a data warehouse using a Time Dimension

Time Factor 1	Time Factor 2	Intersection
Day 1-Start		
Day 2		
Day 3	Day 3-Start	Day 3-Start
Day 4	Day 4	Day 4
Day 5-End	Day 5	Day 5-End
	Day 6	
	Day 7-End	

Problem with Time Dimension

- Using a time dimension you must create a record for the smallest fragment of time you wish to track.
- This can result in a huge amount of data if the period is very small
- The SQL Server DateTime data type handles time parts as small as 3ms. That means that if you needed to break time out by the millisecond for one day you would have 28,800,000 separate records
- The SQL Server Time data type can be much smaller still.

Snowflake Schema with Time Dimension



One Solution

- Use Time Segments Instead of Points in Time
- A time segment tracks two points of time
 - A start time which is the earliest point in time associated with an entity
 - An end time which is the last point in time associated with an entity

Time Segment Comparison

- Time segments can be evaluated against a single point in time

A is between B Start and B End



Time Segment Comparison

Time segments can be evaluated against another time segment

1. Segment A Start between Segment B Start and End



2. Segment A End between Segment B Start and End



3. Segment B Start between Segment A Start and End



4. Segment B End between Segment A Start and End



Types of Time Segments

- **Static**

Static time segments have a fixed start point in time and end point in time

- **Continuous**

Continuous time segments have a fixed start point in time and an open end point in time

- **Terminating**

- Terminating time segments have an open start point in time and a fixed end point in time

Time Segments in Reference to Time

Any particular time segment changes meaning based on a time point of reference. A day is one of the easiest to demonstrate

- Tomorrow, today will be yesterday
- Today is Tuesday
- Today is the 17th day of this month
- Today is in March
- Today is in the first quarter
- Today is in the Calendar year
- Today is in the 4th fiscal quarter
- Today is in the prior month to April
- Today is in the next month to February

Importance of Time Segment

Relationship to Point of Reference

Think of some common reports we like to create to help us understand how something changes over time

- Quarterly Sales Report By Month
- Hourly Utilization Report
- Monthly Sales by Region
- Month To Date, Daily Sales Report
- Two Month Variance Report
- Two Year Variance by quarter report

Creating Virtual Time Segments

All of these reports utilize time segments that can be derived from any point in time

In order to compute the necessary time segments you need three inputs

- 1) A reference point in time
- 2) The number of periods to calculate
- 3) The time duration used to calculate each period (millisecond, second, minute, hour, day, week, month, year, etc.)

Demonstration

Table Value Function generating time segments based on user input

- Starts from a Point of Reference Date
- Generates a set number of segments
- Uses Sequence table to generate incremental time segments
- Uses DATEADD SQL function to calculate start and end of each segment from the Point of Reference Date
- The net result is a series of date segments for the desired number of periods, spanning the desired period of time in each segment