Temporal Issues in Data Warehousing

Presented By

Rokade Arun M.

(Guide : Prof. N. L. Sarda)

Department of Computer Science and Engineering
Indian Institute of Technology

Temporal Issues in Data Warehousing

- Dimension Evolution
- State-based Facts
- New Time Operators
- Schema Evolution

Dimension Evolution

- Dimension tables change with time.
- As an example, consider the data warehouse of student grades shown in fig 1.
- Suppose the roll_number of the student changes. It can be handled in three ways
  - overwrite the old roll_number with the new one - this method does not keep the history of the roll_number.
  - add a new student record with the modified roll_number - history is not co-relatable in this method.
  - make the dimension table temporal - this method keeps the history as well as allows us to query the dimension table to find out when the changes took place.

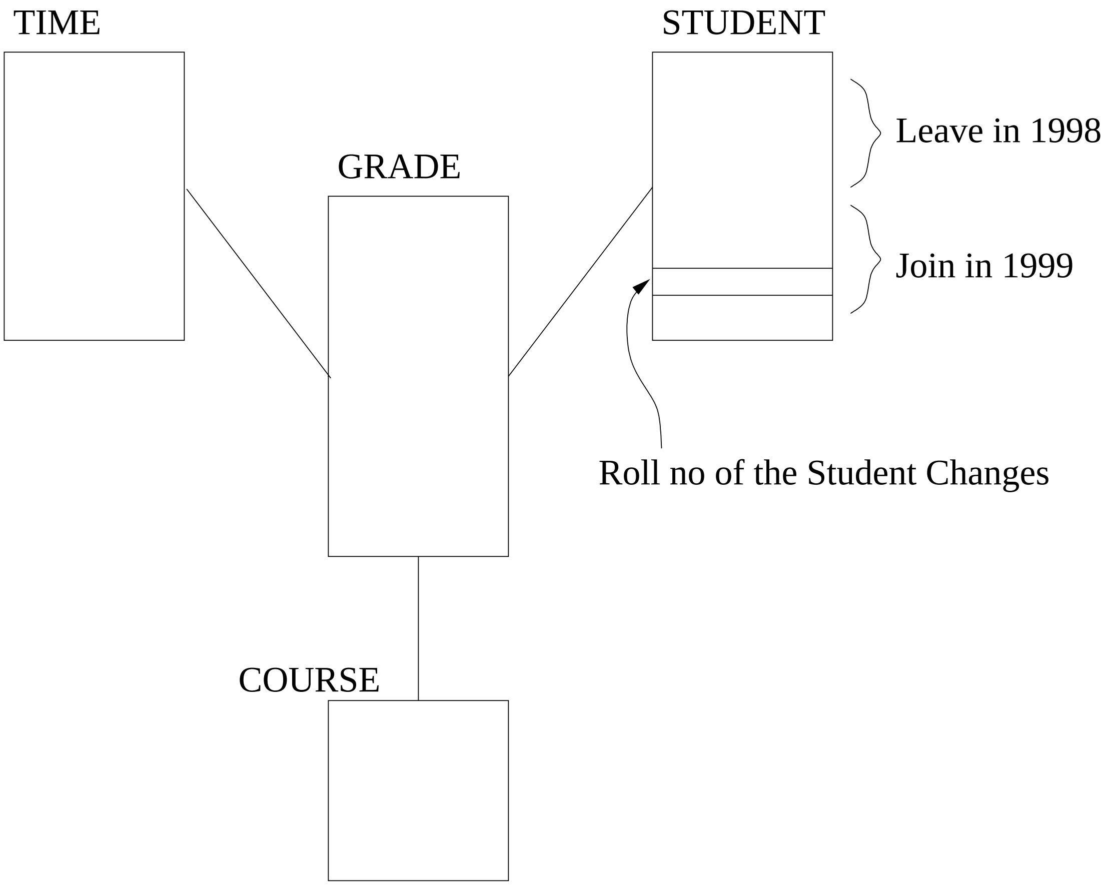

Figure 1: Dimension Evolution for IIT Student Gardes warehouse

State based Fact Table

- Facts may represent states.
- States are defined by start_time and end_time. (see fig 2)

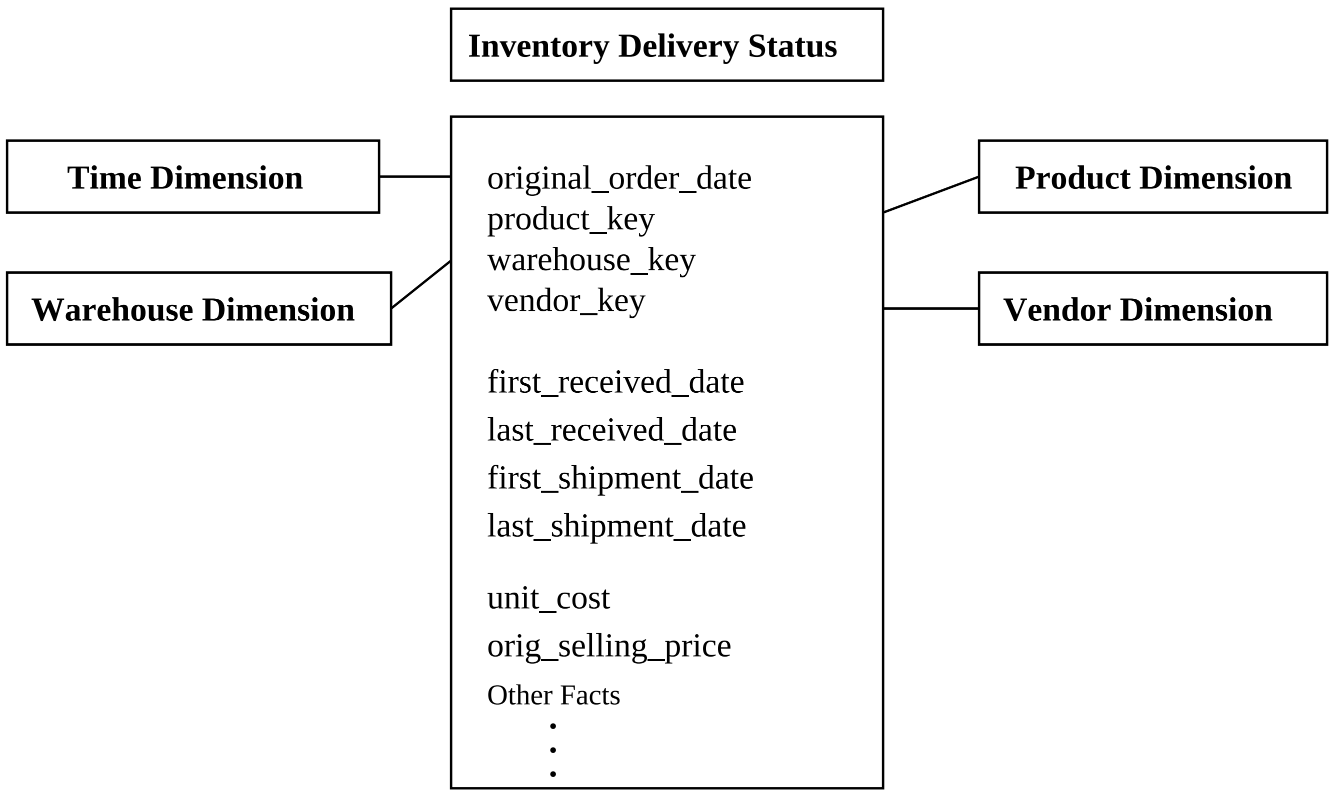

Figure 2: Delivery Status Model Star Schema

The challenging queries here are :

- facts valid at a given time.
- duration-based queries.

New Time Operators

- Time-series computation - since a data warehouse stores time-series data, it should support time-series operations like calculating the average, max, standard deviation and other statistical properties of the data over a given time period.
- Time slice operator - it takes time t as argument and returns all the records that where valid at time t. It is required if we add a time dimension to the dimension tables.
- Moving Window time-series computation - It computes statistical functions like max, average, standard deviation, etc over a time window and moves this window after each computation.

Addition of new attribute to Fact Table

- Conventionally this requires
  - creating a new column for the new attribute.
  - setting its value as NULL for existing records.
- Fact tables are usually very large. This leads to
  - High wastage of storage space.
  - The operation of adding the new attribute requires considerable time.
- We need efficient Schema Evolution technique to overcome above problem.