Slowing Changing Dimension

SalesEmployeeDimention Table

|  |  |
| --- | --- |
| Field | Description |
| SalesEmployeeSSN | This is the employee id number and is the same as employee social security number. Since the employee SSN is unique to each employee, it does not change, and can be represented in integer format, the employee SSN is used in the transactional database as employee id number, and is retained in the dimension table as dimension table primary key. |
| SalesEmployeeName | This is the First Name, Last Name of the employee. Over course of employment, there are cases where either employee’s first name or last name or both have changed. |
| SalesEmployeePosition | This is the position title of the sales employee and can take different values such as sales associate, senior sales associate, account executive etc. Over course of employment as a sales person, an employee may hold different position titles over time. |
| SalesDistrict | The sales territory is organized into a hierarchy of regions and districts. The district and region to which a sales employee is assigned may change over time. Each sales district belongs to a region. However, the sales territory can be re-organized and the assignments of districts to regions can also vary over time. |
| SalesRegion | A sales region consists of sales districts. The assignment of sales districts to sales regions can change when there are reorganizations. Such reorganizations are infrequent. |

1. What are the pros/cons of modeling the changes to employee dimension as type 1, 2 or 3 SCD. What is the best way? Can it be implemented using the current dimension design? What changes would need to be made to the dimension table design?

Type1: Overwrite

Pro: no need to change the table structure;

Con: no historical values will be stored;

Can it be implemented using the current dimensions design: yes, we can replace the old values by the new values;

Type2: Add new row

Pro: keep both historical values and new values, only need to add only two or three new columns;

Cons: adding new rows will take more storage space; if you added effective date and expiration date as new columns. In order to find the most recent values we need to browse all of the rows.

Can it be implemented using the current dimensions design? No, We need to add at least two more columns in this table.

Type3: Add new column

Pro: both historical and new values stored, storage space efficient, easier to query using new column value;

Cons: need to modify the table structure and design every time there’s a change; not suitable for unpredictable changes;

Can it be implemented using the current dimensions design: No need to add new columns when there are changes of values.

What is the best way?

Most of the time we use type2;

For some special predictable changes, we use type 3;

Type 1 is used with other types or can used for replacing typos or mistakes;

1. Assume that the organization has decided to model all the changes to employee dimension as type 2 SCD. At the next quarterly update of the data warehouse, the ETL analyst informs you that EmployeeNo 123456789 has had both a change in SalesDistrict and change in SalesEmployeePosition. Should these changes be added to the dimension table as one new row that captures both changes, or two new rows with one row for Sales Position Change and one row for district change?

If these two changes would be made on the same, so the beginning date /effective date will be the same, we can make the two changes in one row;

If these two changes would be effective on two different days depending on ETL, the two changes will be made in two rows.

If its not effective dates and its effective seconds, as long as these two-time stamps are different the changes should be made to two rows.

3. Certain departments, such as HR, expect frequent changes to records within their systems. For example, an employee’s designation can change multiple times a year. In these cases, companies generally want to be able to maintain the history of data while still being able to quickly query it based on the most current attribute value.

In DWB, you could choose to deal with this situation by setting up the Job Title as an SCD6 field. Given the original table below, design a type 6 dimension table for employee dimension.

|  |  |  |  |
| --- | --- | --- | --- |
| Employee ID | Employee Code | Employee Name | Job Title |
| 30032 | 303 | Frodo Baggins | Sales Manager |

Change job title to “sales advisor”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Employee ID | Employee Code | Employee Name | Historical  Job Title | Current  Job Title | Row Effective Date | Row End date | Current Indicator |
| 30032 | 303 | Frodo Baggins | Sales Manager | Sales advisor | 2021-01-01 | 2023-10-01 | no |
| 30033 | 303 | Frodo Baggins | Sales advisor | Sales advisor | 2023-10-02 | 9999-12-31 | Yes |