What are the benefits and drawbacks of using surrogate keys in a data warehouse?

Powered by AI and the LinkedIn community

A data warehouse is a centralized repository of integrated data from multiple sources that supports analytical queries and decision making. A surrogate key is a unique identifier that is generated and assigned to each row in a data warehouse table, regardless of the source data. Surrogate keys are often used to link dimension tables to fact tables, which store the measures of interest. But what are the benefits and drawbacks of using surrogate keys in a data warehouse? In this article, we will explore some of the main advantages and disadvantages of this approach.

1 Benefits of surrogate keys

One of the main benefits of using surrogate keys in a data warehouse is that they simplify the data integration process. Surrogate keys are independent of the source data, which may have different formats, values, or keys. This means that you do not have to worry about matching or transforming the source keys, or dealing with duplicates or nulls. Surrogate keys also allow you to preserve the history of changes in the source data, by creating new rows with new keys when the source data is updated or deleted. This enables you to track the changes over time and perform historical analysis.

Another benefit of using surrogate keys in a data warehouse is that they improve the performance and scalability of the system. Surrogate keys are usually numeric and sequential, which makes them easy to index and join. Surrogate keys also reduce the size of the data warehouse tables, as they are shorter and more compact than the source keys. This means that you can store more data in less space, and speed up the query execution and data loading. Surrogate keys also enable you to partition the data warehouse tables based on the key ranges, which can enhance the parallelism and availability of the system.

2 Drawbacks of surrogate keys

One of the main drawbacks of using surrogate keys in a data warehouse is that they add complexity and overhead to the data modeling and management process. Surrogate keys are artificial and meaningless, which means that they do not convey any information about the data or the business logic. This can make it harder to understand and document the data warehouse schema, and to trace the data lineage and quality. Surrogate keys also require additional steps and tools to generate and maintain them, such as sequences, triggers, or ETL processes. This can increase the development and maintenance costs and introduce potential errors or inconsistencies.

Another drawback of using surrogate keys in a data warehouse is that they limit the flexibility and interoperability of the system. Surrogate keys are specific to the data warehouse, which means that they do not match the keys used in the source systems or in other external systems. This can make it difficult to integrate or synchronize the data warehouse with other systems, or to migrate or share the data across different platforms. Surrogate keys also restrict the ability to query or analyze the data using the natural or business keys, which may be more meaningful or relevant for some purposes. For example, you may want to query the data by customer name, product code, or order date, rather than by surrogate key.

3 Here's what else to consider

This is a space to share examples, stories, or insights that don't fit into any of the previous sections. What else would you like to add?

DBMS + Follow

Rate this article

We created this article with the help of AI. What do you think of it?

It's great It's not so great

Report this article

More articles on DBMS

How do ER diagrams help with database normalization and optimization in DBMS?
4 contributions

How can you implemer transactions in a distri
5 contributions

See all

Top experts in this article

Experts who add quality contributions will have a chance to be featured. **Learn more**

Earn a Community Top Voice badge

Add to collaborative articles to get recognized for your expertise on your profile. **Learn more**

Start a contribution

See what others are saying

Add your perspective

Add your perspective

Add your perspective

More relevant reading

Database Development
What is the best way to implement a denormalized data warehouse?

Data Architecture
What are the differences between the Kimball and Inmon data warehouse development methodologies?

Data Warehousing
How can you handle null values in a star schema?

Data Warehousing
What are the disadvantages of using Slowly Changing Dimension Type 2?