**Surrogate Key**

Surrogate keys are widely used and accepted design standard in data warehouses. It is sequentially generated unique number attached with each and every record in a Dimension table in any Data Warehouse. It joins the fact and dimension tables and is necessary to handle changes in dimension table attributes.

**Features**

Surrogate Key (SK) is sequentially generated meaningless unique number attached with each and every record in a table in any Data Warehouse (DW).

* It is unique since it is sequentially generated integer for each record being inserted in the table.
* It is meaningless since it does not carry any business meaning regarding the record it is attached to in any table.
* It is sequential since it is assigned in sequential order as and when new records are created in the table, starting with one and going up to the highest number that is needed.

**Why Should We Use Surrogate Key**

* Basically, it’s an artificial key that is used as a substitute for a Natural Key (NK).
* We should have defined natural key in our tables as per the business requirement and that might be able to uniquely identify any record.
* But surrogate key is just an Integer attached to a record for the purpose of joining different tables in a Star or Snowflake schema-based Data Warehouse.
* Surrogate key is much needed when we have very long natural key or the datatype of the natural key is not suitable for Indexing.
* If we replace the natural key with a single Integer, it should be able to save a substantial amount of storage space.
* The surrogate key of different Dimensions would be stored as Foreign Keys (FK) in the Fact tables to maintain Referential Integrity (RI), and here instead of storing of those big or huge natural key, storing of concise surrogate key would result in less amount of space needed.

**Example of a Surrogate Key**

**Without a Surrogate Key (Using Natural Key)**

CREATE TABLE Customers (  
 customer\_email VARCHAR(255) PRIMARY KEY,   
 name VARCHAR(100),  
 phone VARCHAR(20)  
);

**Problem:** If the email changes, we must update it in all related tables.

**With a Surrogate Key (Auto-increment ID)**

CREATE TABLE Customers (  
 customer\_id INT IDENTITY(1,1) PRIMARY KEY,   
 customer\_email VARCHAR(255) UNIQUE,  
 name VARCHAR(100),  
 phone VARCHAR(20)  
);

**Solution:** Even if customer\_email changes, customer\_id remains **unchanged**.