

Surrogate Keys

Surrogate keys are unique identifiers assigned to each record in a database table. Unlike natural keys, which are based on existing data attributes, surrogate keys are generated specifically for identification purposes. They serve as primary keys in database tables and are often integers or GUIDs (Globally Unique Identifiers).

Surrogate keys offer several advantages:

1. **Uniqueness:** Surrogate keys guarantee uniqueness within a table, simplifying data integrity and consistency.
2. **Stability:** Surrogate keys remain constant even if the underlying data changes, making them reliable for referencing.
3. **Performance:** Surrogate keys, especially integer-based ones, are usually more efficient for indexing and querying compared to composite or large natural keys.
4. **Normalization:** Surrogate keys facilitate database normalization by eliminating the need to rely on complex or composite natural keys.

However, surrogate keys also have some drawbacks:

1. **Additional Storage:** Surrogate keys consume additional storage space in the database.
2. **Complexity:** Introducing surrogate keys can add complexity to the database schema and queries, especially when dealing with joins across multiple tables.
3. **Contextual Understanding:** Surrogate keys lack inherent meaning or context, which might make it challenging for users to interpret the data without referring to other attributes.

Use cases of Surrogate Keys:

Surrogate keys find application in various scenarios across database design and management. Here are some common use cases:

1. **Data Warehousing:** In data warehousing environments, where data from multiple sources is consolidated into a single repository, surrogate keys are often employed to ensure uniqueness and facilitate efficient data integration.
2. **Performance Optimization:** Surrogate keys, especially integer-based ones, can significantly improve database performance, particularly in large tables, by simplifying indexing and speeding up joins and queries.
3. **History Tracking:** Surrogate keys are useful for tracking historical changes to records in a database. By using surrogate keys as primary keys and maintaining effective date ranges or version numbers, it becomes easier to track and analyze changes over time.
4. **Data Migration and Integration:** Surrogate keys play a crucial role in data migration and integration projects, where data from disparate sources needs to be combined. They provide a consistent reference point for mapping and transforming data across different systems.

5. **Distributed Systems:** In distributed databases or systems where data needs to be replicated across multiple nodes or partitions, surrogate keys help ensure consistency and avoid conflicts when merging or synchronizing data.
6. **Security and Privacy:** Surrogate keys can enhance security and privacy by abstracting sensitive information. For example, instead of using a customer's social security number as a primary key, a surrogate key can be employed to represent the customer's identity while keeping the sensitive data separate.
7. **Object-Relational Mapping (ORM):** Surrogate keys are often used in object-relational mapping frameworks, where they provide a simple and consistent way to identify objects in the application code, decoupling the database schema from the application logic.
8. **Sharding:** In sharded database architectures, where data is horizontally partitioned across multiple servers or shards, surrogate keys help ensure that each shard can independently generate unique identifiers without the risk of collisions.

In this process, we're utilizing Azure Data Factory's Mapping Data Flows to generate surrogate keys for dimension tables in a data warehouse environment. The end goal is to ensure that each record in the dimension tables has a unique identifier, facilitating data integration, querying, and analysis. By creating surrogate keys, we enhance data integrity, simplify joins across multiple data sources, and improve overall performance within the data warehouse. This process involves dropping and recreating dimension tables, configuring data flows to generate surrogate keys, validating the changes, and ultimately ensuring that the dimension tables contain accurate and uniquely identifiable data for effective analysis and reporting.

To begin with the Lab:

1. In this lab we will go through how we can build our Surrogate keys in Dimension Tables via the use of Mapping data flows.
2. If you want to have a unique identifier for your rows in your dimension tables, so in case the data in your dimension tables is coming in from different data sources, it's good to have a surrogate key in place for your dimension tables.
3. So, in this case, you could have your product ID, your Product Model ID, and your Product Category ID, as your business keys, right? Because they're coming in from the sources of data. But if you want to uniquely identify the roles within the table, we could go ahead and have the Product SK as the surrogate key.
4. Now we'll actually make use of mapping data flows to generate that value when it comes onto the surrogate key.
5. First we are going to drop our dimension product table then we will create a new dimension product with the Surrogate key.
6. Below you can see that we dropped our table and created a new table.

SQLQuery1.sql - dat... (sqladminuser (0)) * X Surrogate Keys Dimension tables

```
DROP TABLE [dbo].[DimProduct]

CREATE TABLE [dbo].[DimProduct](
    [ProductSK] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [ProductModelID] [int] NOT NULL,
    [ProductcategoryID] [int] NOT NULL,
    [ProductName] varchar(50) NOT NULL,
    [ProductModelName] varchar(50) NULL,
    [ProductCategoryName] varchar(50) NULL
)
WITH
(
    DISTRIBUTION = REPLICATE
)

SELECT * FROM [dbo].[DimProduct]
```

150 %

Results Messages

| ProductSK | ProductID | ProductModelID | ProductcategoryID | ProductName | ProductModelName | ProductCategoryName |
|-----------|-----------|----------------|-------------------|-------------|------------------|---------------------|
|-----------|-----------|----------------|-------------------|-------------|------------------|---------------------|

- After that you need to delete the data from your customer table too.

```
DELETE FROM [dbo].[DimCustomer]
```

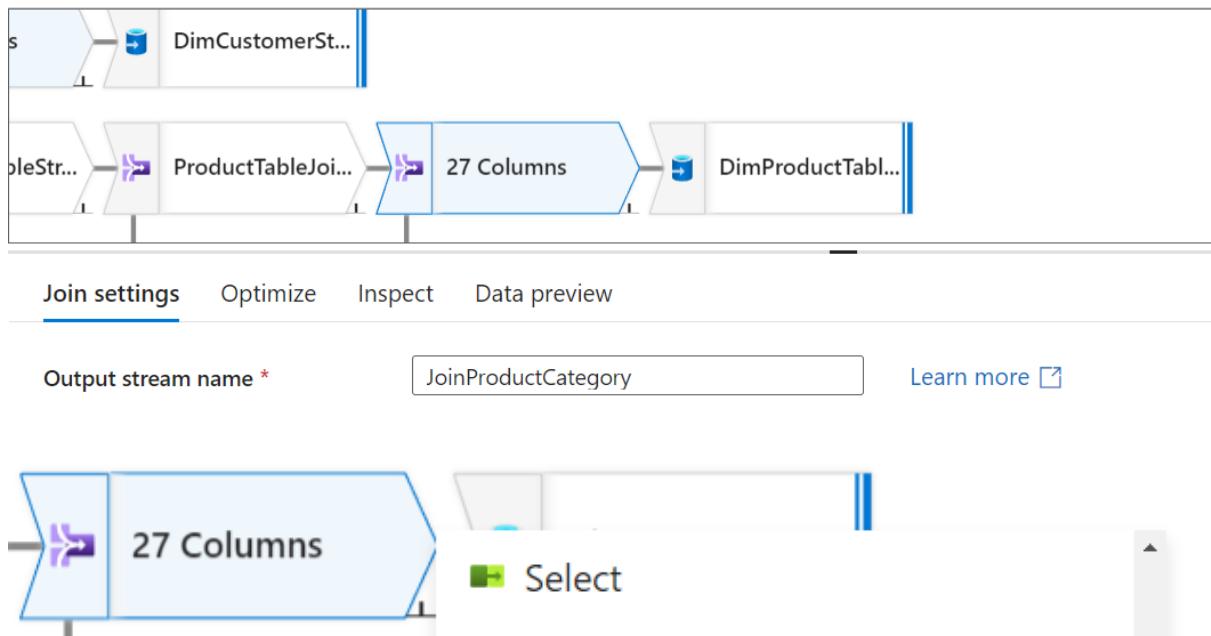
150 %

Messages

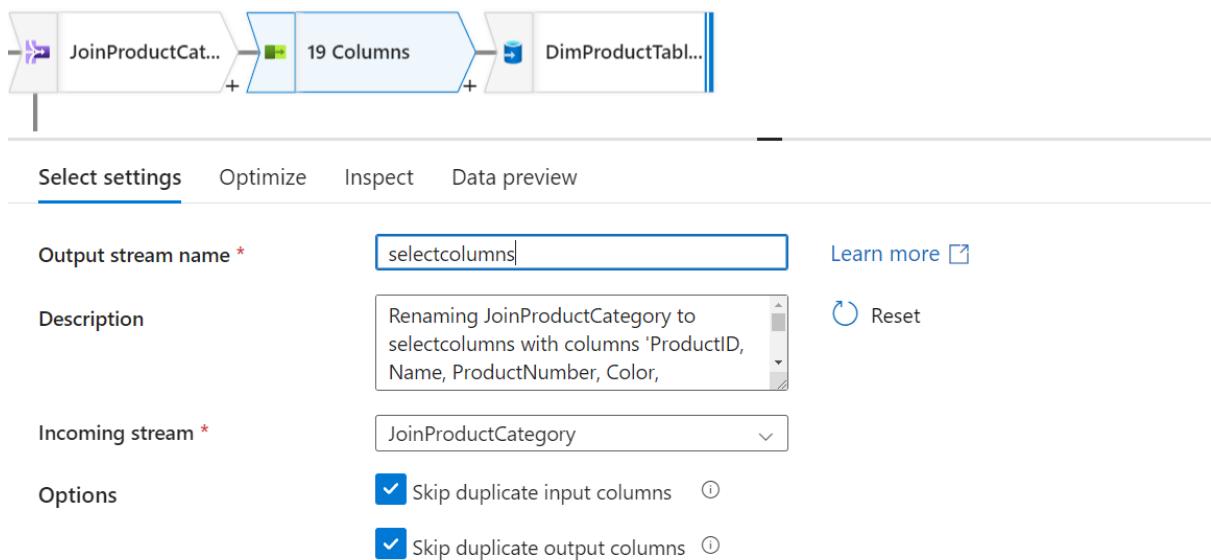
```
(847 rows affected)
```

```
Completion time: 2024-05-01T14:25:40.7713052+05:30
```

- Now we need to go Data Factory wizard and select our Data Flow for dimension tables.
Then we are going to choose Join Product Category Stream.
- Now we need to click on plus icon from Join product category stream and choose to select to select our rows of interest.



10. First we need to give it a name then scroll down and you will see mappings.



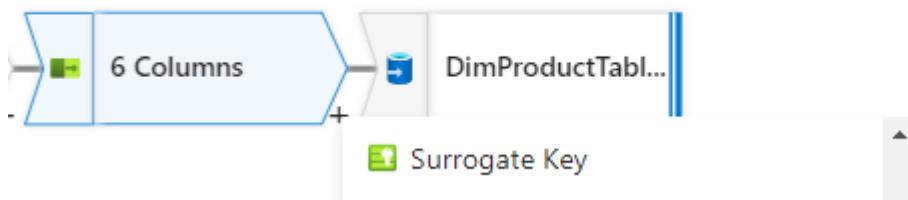
11. Now in the mappings we just need the below mapping and we need to remove the other mappings.

| JoinProductCategory's column | Name as |
|--|-------------------|
| 123 ProductID | ProductID |
| abc ProductTableStream@Name | Name |
| 123 ProductTableStream@ProductCategoryID | ProductCategoryID |
| 123 ProductTableStream@ProductModelID | ProductModelID |
| abc ProductModelStream@Name | Name |
| abc ProductCategoryStream@Name | Name |

12. After that we need to change the names of the highlighted mappings.

| JoinProductCategory's column | Name as |
|--|---------------------|
| 123 ProductID | ProductID |
| abc ProductTableStream@Name | ProductName |
| 123 ProductTableStream@ProductCategoryID | ProductCategoryID |
| 123 ProductTableStream@ProductModelID | ProductModelID |
| abc ProductModelStream@Name | ProductmodelName |
| abc ProductCategoryStream@Name | ProductCategoryName |

13. After that from the select columns, click on plus icon and choose Surrogate key.



14. Now give your stream a name then your incoming column is select columns and the key column name is ProductSK.

Surrogate key settings Optimize Inspect Data preview

| | | |
|----------------------|--|------------------------------|
| Output stream name * | ProductSKStream | Learn more □ |
| Description | Adding new key ProductSK starting from 1 with step 1 | Reset |
| Incoming stream * | selectcolumns | |
| Key column * | ProductSK | |
| Start value * | 1 | |
| Step value | 1 | |

15. Then you need to open the Dimension product table stream and click on open.

Sink Settings Errors Mapping Optimize Inspect Data preview

Output stream name * DimProductTableStream [Learn more](#)

Description Export data to DimProduct_table [Reset](#)

Incoming stream * ProductSKStream

Sink type *

| | | |
|--|--|--|
| | | |
|--|--|--|

Dataset * DimProduct_table [Test connection](#) [Open](#) [New](#)

Options

- Allow schema drift ⓘ
- Validate schema ⓘ

16. Now move to schema click on import schema to get the Product SK column

| Connection | Schema | Parameters |
|--------------------------------|---|------------|
| | Import schema Clear | |
| Column name Type | | |
| ProductSK | int | |
| ProductID | int | |
| ProductModelID | int | |
| ProductcategoryID | int | |
| ProductName | varchar | |
| ProductmodelName | varchar | |
| ProductCategoryName | varchar | |

17. After that go to mappings and click on reset.

Sink Settings Errors **Mapping** Optimize Inspect Data preview

⚠️ At least one incoming column is mapped to a column in the sink dataset schema with a conflicting type, which can cause NULL values or runti

Options

- Skip duplicate input columns
- Skip duplicate output columns
- Auto mapping

Output format

| Input columns | Output columns |
|-------------------------|-------------------------|
| 12l ProductSK | 123 ProductSK |
| 123 ProductID | 123 ProductID |
| 123 ProductModelID | 123 ProductModelID |
| 123 ProductCategoryID | 123 ProductcategoryID |
| abc ProductName | abc ProductName |
| abc ProductModelName | abc ProductModelName |
| abc ProductCategoryName | abc ProductCategoryName |

18. Now click on Validate all and then click on Publish all.
19. Then you need to go back to your pipeline and click on Trigger to run your pipeline.
20. After some time, you will see that your pipeline run was successful.

Run Succeeded

Successfully ran 05_mapping_dataflow_dimtable (Pipeline).
[View pipeline run](#)

a few seconds ago

All pipeline runs > **05_mapping_dataflow_dimtable - Activity runs**

Rerun **Cancel** **Refresh** **Update pipeline** **List** **Gantt**

| Activity runs | | | | | | | | |
|---|--|---------------|----------------------|----------|------------------------|-----------------|--------------------------------------|-----|
| Pipeline run ID: b9f03e6b-63bc-4c1d-914a-8876acd6c4c3 | | | | | | | | |
| All status | | | | | | | | |
| Showing 1 - 1 items | | | | | | | | |
| Activity name | Activity status | Activity type | Run start | Duration | Integration runtime | User properties | Activity run ID | Log |
| Dataflow_DimTable | Succeeded | Data flow | 5/1/2024, 2:30:37 PM | 3m 31s | AutoResolveIntegration | | 7d4c6773-a1a1-4adc-8269-94d6c53853bd | |

21. Now come back to SSMS run the select statement for the product table and you will see the data in place with the Surrogate key.

```
SELECT * FROM [dbo].[DimProduct]
```

150 %

Results Messages

| | ProductSK | ProductID | ProductModelID | ProductcategoryID | ProductName | ProductModelName | ProductCategoryName |
|----|-----------|-----------|----------------|-------------------|-------------------------------|---------------------------|---------------------|
| 1 | 2 | 878 | 121 | 34 | Fender Set - Mountain | Fender Set - Mountain | Fenders |
| 2 | 187 | 946 | 47 | 8 | LL Touring Handlebars | LL Touring Handlebars | Handlebars |
| 3 | 195 | 712 | 2 | 23 | AWC Logo Cap | Cycling Cap | Caps |
| 4 | 168 | 807 | 61 | 15 | HL Headset | HL Headset | Headsets |
| 5 | 174 | 995 | 96 | 9 | ML Bottom Bracket | ML Bottom Bracket | Bottom Brackets |
| 6 | 295 | 880 | 107 | 36 | Hydration Pack - 70 oz. | Hydration Pack | Hydration Packs |
| 7 | 148 | 915 | 65 | 19 | ML Touring Seat/Saddle | ML Touring Seat/Saddle | Saddles |
| 8 | 261 | 876 | 118 | 30 | Hitch Rack - 4-Bike | Hitch Rack - 4-Bike | Bike Racks |
| 9 | 188 | 808 | 52 | 8 | LL Mountain Handlebars | LL Mountain Handlebars | Handlebars |
| 10 | 196 | 842 | 120 | 39 | Touring-Panniers, Large | Touring-Panniers | Panniers |
| 11 | 169 | 805 | 59 | 15 | LL Headset | LL Headset | Headsets |
| 12 | 175 | 996 | 97 | 9 | HL Bottom Bracket | HL Bottom Bracket | Bottom Brackets |
| 13 | 184 | 707 | 33 | 35 | Sport-100 Helmet, Red | Sport-100 | Helmets |
| 14 | 149 | 910 | 81 | 19 | HL Mountain Seat/Saddle | HL Mountain Seat/Saddle 2 | Saddles |
| 15 | 258 | 803 | 105 | 14 | ML Fork | ML Fork | Forks |
| 16 | 189 | 812 | 57 | 8 | ML Road Handlebars | ML Road Handlebars | Handlebars |
| 17 | 20 | 855 | 12 | 22 | Men's Bib-Shorts, S | Men's Bib-Shorts | Bib-Shorts |
| 18 | 170 | 806 | 60 | 15 | ML Headset | ML Headset | Headsets |
| 19 | 176 | 994 | 95 | 9 | LL Bottom Bracket | LL Bottom Bracket | Bottom Brackets |
| 20 | 185 | 708 | 33 | 35 | Sport-100 Helmet, Black | Sport-100 | Helmets |
| 21 | 150 | 913 | 84 | 19 | HL Road Seat/Saddle | HL Road Seat/Saddle 2 | Saddles |
| 22 | 259 | 804 | 106 | 14 | HL Fork | HL Fork | Forks |
| 23 | 190 | 809 | 54 | 8 | ML Mountain Handlebars | ML Mountain Handlebars | Handlebars |
| 24 | 21 | 856 | 12 | 22 | Men's Bib-Shorts, M | Men's Bib-Shorts | Bib-Shorts |
| 25 | 96 | 885 | 7 | 20 | HL Touring Frame - Yellow, 60 | HL Touring Frame | Touring Frames |
| 26 | 177 | 941 | 53 | 17 | Touring Pedal | Touring Pedal | Pedals |
| 27 | 186 | 711 | 33 | 35 | Sport-100 Helmet, Blue | Sport-100 | Helmets |
| 28 | 151 | 911 | 82 | 19 | LL Road Seat/Saddle | LL Road Seat/Saddle 1 | Saddles |
| 29 | 260 | 802 | 104 | 14 | LL Fork | LL Fork | Forks |
| 30 | 191 | 947 | 48 | 8 | HL Touring Handlebars | HL Touring Handlebars | Handlebars |
| 31 | 22 | 857 | 12 | 22 | Men's Bib-Shorts, L | Men's Bib-Shorts | Bib-Shorts |
| 32 | 97 | 887 | 7 | 20 | HL Touring Frame - Yellow, 46 | HL Touring Frame | Touring Frames |
| 33 | 178 | 937 | 64 | 17 | HL Mountain Pedal | HL Mountain Pedal | Pedals |

✓ Query executed successfully.

22. And if we do an order by to Product SK column then it is also arranging in proper order.

```
SELECT * FROM [dbo].[DimProduct]
ORDER BY [ProductSK]
```

150 %

Results Messages

| | ProductSK | ProductID | ProductModelID | ProductcategoryID | ProductName | ProductModelName | ProductCategoryName |
|----|-----------|-----------|----------------|-------------------|----------------------------|-------------------------|---------------------|
| 1 | 1 | 879 | 122 | 31 | All-Purpose Bike Stand | All-Purpose Bike Stand | Bike Stands |
| 2 | 2 | 878 | 121 | 34 | Fender Set - Mountain | Fender Set - Mountain | Fenders |
| 3 | 3 | 852 | 38 | 28 | Women's Tights, S | Women's Tights | Tights |
| 4 | 4 | 853 | 38 | 28 | Women's Tights, M | Women's Tights | Tights |
| 5 | 5 | 854 | 38 | 28 | Women's Tights, L | Women's Tights | Tights |
| 6 | 6 | 841 | 13 | 26 | Men's Sports Shorts, S | Men's Sports Shorts | Shorts |
| 7 | 7 | 849 | 13 | 26 | Men's Sports Shorts, M | Men's Sports Shorts | Shorts |
| 8 | 8 | 850 | 13 | 26 | Men's Sports Shorts, L | Men's Sports Shorts | Shorts |
| 9 | 9 | 851 | 13 | 26 | Men's Sports Shorts, XL | Men's Sports Shorts | Shorts |
| 10 | 10 | 867 | 37 | 26 | Women's Mountain Shorts, S | Women's Mountain Shorts | Shorts |
| 11 | 11 | 868 | 37 | 26 | Women's Mountain Shorts, M | Women's Mountain Shorts | Shorts |
| 12 | 12 | 869 | 37 | 26 | Women's Mountain Shorts, L | Women's Mountain Shorts | Shorts |
| 13 | 13 | 874 | 24 | 27 | Racing Socks, M | Racing Socks | Socks |
| 14 | 14 | 875 | 24 | 27 | Racing Socks, L | Racing Socks | Socks |
| 15 | 15 | 709 | 18 | 27 | Mountain Bike Socks, M | Mountain Bike Socks | Socks |
| 16 | 16 | 710 | 18 | 27 | Mountain Bike Socks, L | Mountain Bike Socks | Socks |
| 17 | 17 | 951 | 101 | 12 | HL Crankset | HL Crankset | Cranksets |
| 18 | 18 | 950 | 100 | 12 | ML Crankset | ML Crankset | Cranksets |
| 19 | 19 | 949 | 99 | 12 | LL Crankset | LL Crankset | Cranksets |
| 20 | 20 | 855 | 12 | 22 | Men's Bib-Shorts, S | Men's Bib-Shorts | Bib-Shorts |
| 21 | 21 | 856 | 12 | 22 | Men's Bib-Shorts, M | Men's Bib-Shorts | Bib-Shorts |
| 22 | 22 | 857 | 12 | 22 | Men's Bib-Shorts, L | Men's Bib-Shorts | Bib-Shorts |
| 23 | 23 | 945 | 103 | 13 | Front Derailleur | Front Derailleur | Derailleurs |
| 24 | 24 | 894 | 127 | 13 | Rear Derailleur | Rear Derailleur | Derailleurs |
| 25 | 25 | 977 | 31 | 6 | Road-750 Black, 58 | Road-750 | Road Bikes |
| 26 | 26 | 997 | 31 | 6 | Road-750 Black, 44 | Road-750 | Road Bikes |
| 27 | 27 | 998 | 31 | 6 | Road-750 Black, 48 | Road-750 | Road Bikes |
| 28 | 28 | 999 | 31 | 6 | Road-750 Black, 52 | Road-750 | Road Bikes |
| 29 | 29 | 754 | 28 | 6 | Road-450 Red, 58 | Road-450 | Road Bikes |
| 30 | 30 | 755 | 28 | 6 | Road-450 Red, 60 | Road-450 | Road Bikes |
| 31 | 31 | 756 | 28 | 6 | Road-450 Red, 44 | Road-450 | Road Bikes |
| 32 | 32 | 757 | 28 | 6 | Road-450 Red, 48 | Road-450 | Road Bikes |
| 33 | 33 | 758 | 28 | 6 | Road-450 Red, 52 | Road-450 | Road Bikes |

Query executed successfully.