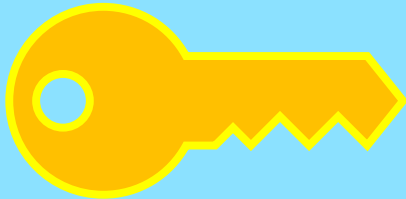


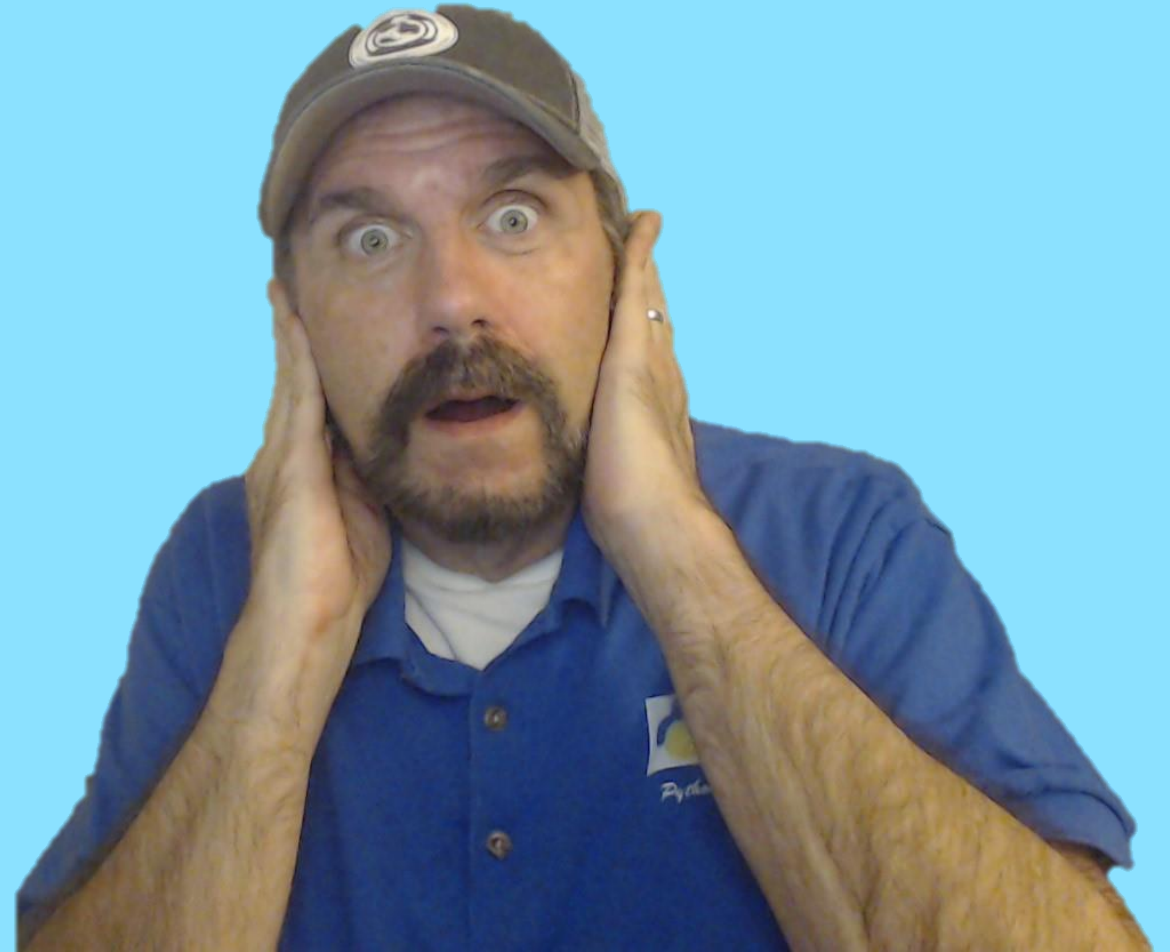
How to Use Delta Table Identity Columns



databricks



Bryan Cafferky, YouTube Channel Creator



Where Are We Going?

- **What is an Identity Column?**
- **Use Cases**
- **How to Create an Identity Column**
- **Challenges to Identity Columns on Scaled Out Platforms**
- **Limitations**

What is an Identity Column?

- **Is an Automatically Incrementing Value: 1, 2, 3...**
- **Assign the Identity Type to a table column.**
- **Goal is to get a unique column value for every row.**
- **Popular in SQL databases especially SQL Server.**
- **Similar to a Sequence except an Identity is Bound to a Column.**

Use Cases

- **Provides a Guaranteed Unique Primary Key.**
- **Because they use integers, they perform well.**
- **You Do NOT Provide a Value on an Identity Column Insert.**
- **Popular Way to Create Surrogate Keys.**

How Identity Columns Work

```
CREATE OR REPLACE TABLE demo (  
  id BIGINT GENERATED ALWAYS AS IDENTITY,  
  product_type STRING,  
  sales BIGINT  
);
```

No Id column
value provided!

```
INSERT INTO demo (product_type, sales)  
VALUES ("Batteries", 150000);  
  
INSERT INTO demo (product_type,  
sales) VALUES ("Bread", 100000);  
  
INSERT INTO demo (product_type,  
sales) VALUES ("Gum", 25000);
```

id	product	sales
1	Batteries	150000
2	Bread	100000
3	Gum	25000

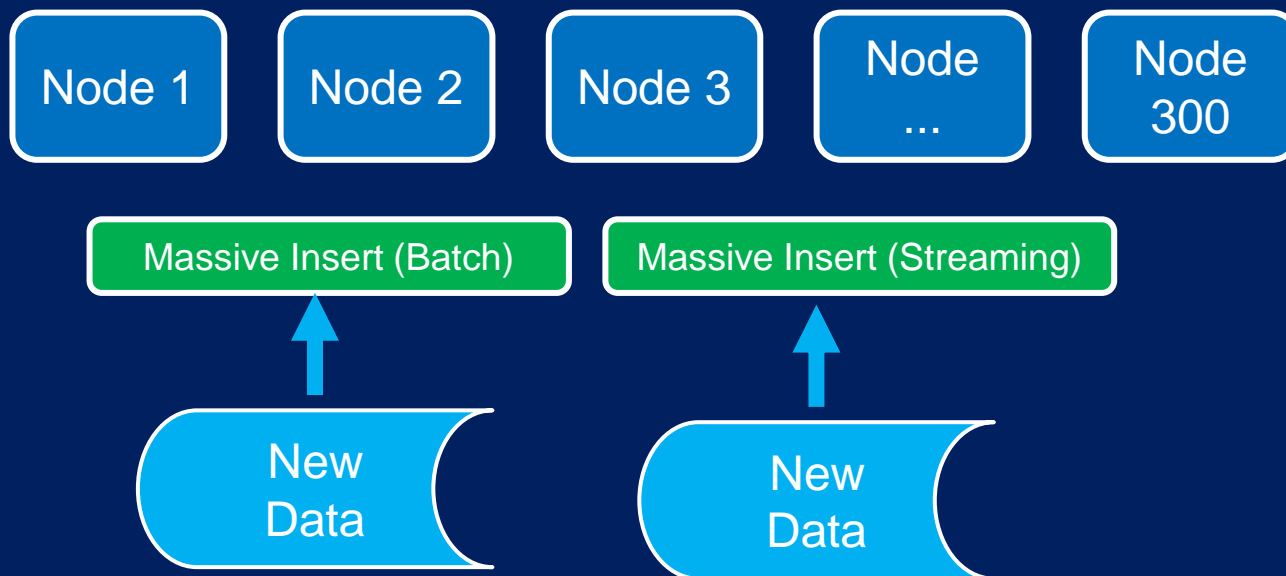
Blog Post:

<https://www.databricks.com/blog/2022/08/08/identity-columns-to-generate-surrogate-keys-are-now-available-in-a-lakehouse-near-you.html>

Challenges on a Scaled-Out Platform

id	product	sales
1	Batteries	150000
2	Bread	100000
3	Gum	25000
...
999999999999

- How does Spark get the next Identity Value?
- How to Avoid Impact Spark Performance?
- Potential for Table Locks.
 - Note: In relational database data warehouses, maintenance is usually done when the data is not being used.
- What About the Delta Logs?



Identity Column Syntax & Limitations

```
GENERATED { ALWAYS | BY DEFAULT } AS IDENTITY [ ( [ START WITH start ] [ INCREMENT BY step ] ) ]
```

Applies to:  Databricks SQL  Databricks Runtime 10.3 and above

Defines an identity column. When you write to the table, and do not provide values for the identity column, it will be automatically assigned a unique and statistically increasing (or decreasing if `step` is negative) value.

This clause is only supported for Delta Lake tables. This clause can only be used for columns with BIGINT data type.

The automatically assigned values start with `start` and increment by `step`. Assigned values are unique but are not guaranteed to be contiguous. Both parameters are optional, and the default value is 1. `step` cannot be 0.

If the automatically assigned values are beyond the range of the identity column type, the query will fail.

When `ALWAYS` is used, you cannot provide your own values for the identity column.

The following operations are not supported:

- `PARTITIONED BY` an identity column
- `UPDATE` an identity column


Note

Declaring an identity column on a Delta table disables concurrent transactions. Only use identity columns in use cases where concurrent writes to the target table are not required.

Spoiler Alert: How Identity Column Values Are Maintained

```
{'metaData': {'id': 'ebea5bfc-dfd8-484f-aa61-16fc05df5fc0', 'description': 'Logs all data lake ETL job executions', 'format': {'provider': 'parquet',  
'options': {}}, 'schemaString': '{"type": "struct", "fields": [{"name": "JobExecutionID", "type": "long", "nullable": true, "metadata": {"delta.identity.start":  
1, "delta.identity.step": 1, "delta.identity.highWaterMark": 1, "delta.identity.allowExplicitInsert": false}}, {"name": "AppID", "type": "integer", "nullable": false,  
"metadata": {}}, {"name": "JobID", "type": "integer", "nullable": false, "metadata": {}}, {"name": "Audit_Pattern", "type": "string", "nullable": false, "metadata":  
{}}, {"name": "StartDateTime", "type": "timestamp", "nullable": false, "metadata": {}}, {"name": "EndDateTime", "type": "timestamp", "nullable": true, "metadata": {}},  
{}, {"name": "SuccessInd", "type": "boolean", "nullable": true, "metadata": {}}, {"name": "JobStatus", "type": "string", "nullable": true, "metadata": {}}, {"name": "Message", "type": "string", "nullable": true, "metadata": {}}, {"name": "EntityName", "type": "string", "nullable": true, "metadata": {}}, {"name": "InsertConflictsCount", "type": "integer", "nullable": true, "metadata": {}}, {"name": "UpdateConflictsCount", "type": "integer", "nullable": true, "metadata": {}}]'}, 'partitionColumns': [],  
'configuration': {}, 'createdTime': 1696693889778}}
```


Wrapping Up

- **What is an Identity Column?**
- **Use Cases**
- **How**  **Column**
- **Challenges to Identity Columns on Scaled Out Platforms**
- **Limitations**

Creating Delta Tables Documentation:

<https://docs.databricks.com/en/sql/language-manual/sql-ref-syntax-ddl-create-table-using.html>