DSS030 - Cannot insert the value NULL into column 'X', table 'Y'; column does not allow nulls

Last updated by | Vitor Tomaz | Aug 16, 2021 at 9:43 AM PDT

Contents

Issue

The error "Cannot insert the value NULL into column 'X', table Y'; column does not allow nulls."

Note: if error is "Cannot insert the value NULL into column provision_timestamp, table DataSync.provision_marker_dss" check DSS034

Investigation/Analysis

Possible causes:

- A table doesn't have a primary key. To fix this issue, add a primary key to all the tables that you're syncing.

- There's a WHERE clause in your CREATE INDEX statement. Data Sync doesn't handle this condition. To fix this issue, remove the WHERE clause or manually make the changes to all databases.

- Check Data Sync Health Checker results for warnings like: WARNING: [dbo].[TableX].[ColumnY] is not included in the sync group but is NOT NULLABLE, not a computed column or has a default value!

  If column is not included in the sync group, Data Sync will not have data for that column. If column is not nullable a value must be set, but Data Sync has no data for this column so inserts will fail as expected.

  Customer should decide what to do, mitigation options include:

    - Include the column in the sync group
    - Set column as nullable in the destination database
    - Add a default value in the destination database
    - Remove the column from the destination database
- In case this is not a basic schema issues, it can be caused by invalid tracking records on the source (tracking record that thinks the row exist but the row does not actually exist in the table).

  Use DSS004 - Data Sync Health Checker to detect invalid tracking records.

  It can validate tracking records for some or all tables, in hub and/or member.

It will show the delete statement that needs to be run in case it detects invalid records.

You can control this using:

```
## Tracking Record Validations
# Set as "All" to validate all tables
# or pick the tables you need using '[dbo].[TableName1]','[dbo].[TableName2]'
ExtendedValidationsTableFilter = @('All')
ExtendedValidationsEnabledForHub = $true  # Attention, this may cause high I/O impact
ExtendedValidationsEnabledForMember = $true  # Attention, this may cause high I/O impact
ExtendedValidationsCommandTimeout = 900 # seconds
```

Output example:

```
WARNING: Tracking Records for Table [dbo].[InvalidTrackingRecordsTable] may have 1 invalid records!
DELETE DataSync.[InvalidTrackingRecordsTable_dss_tracking]
FROM DataSync.[InvalidTrackingRecordsTable_dss_tracking] t
WHERE sync_row_is_tombstone=0 AND NOT EXISTS (SELECT * FROM [dbo].[InvalidTrackingRecordsTable] s WHERE t
s.Id)
```

In case invalid tracking records are detected, Data Sync Health Checker will also show a delete statement that can be used to remove them. Please note that invalid tracking records are usually generated by customers updating primary key values, this is not supported as listed in the General requirements ↗

Classification

Root cause Tree - DataSync/User issue/error/SyncFail

How good have you found this content?

🙂 🙁