Enabling Change Data Capture (CDC) in Azure SQL Database

# Step 1: Check for any Enabled Triggers

SELECT name, object\_id   
FROM sys.triggers   
WHERE parent\_class\_desc = 'DATABASE'   
AND is\_disabled = 0;

Purpose: Identify active database-level triggers (especially DDL triggers that may block CDC enablement).

# Step 2: View Trigger Definition – Pass object id from above script

SELECT OBJECT\_DEFINITION(object\_id) AS trigger\_definition;

Purpose: Review the trigger name and logic for understanding its impact.

# Step 3: Disable the Trigger – Get trigger name from above script and disable

DISABLE TRIGGER **trg\_ddl\_table\_changes** ON DATABASE;

Why: Some DDL triggers can prevent enabling CDC — disable temporarily.

# Step 4: Enable CDC on Database

EXEC sys.sp\_cdc\_enable\_db;

Goal: Enables CDC at the database level.

# Step 5: Enable CDC on Table

EXEC sys.sp\_cdc\_enable\_table   
 @source\_schema = N'dbo',   
 @source\_name = N'account',   
 @role\_name = NULL;

Effect: Starts tracking changes on the dbo.account table.

# Step 6: Re-Enable Trigger

ENABLE TRIGGER trg\_ddl\_table\_changes ON DATABASE;

Note: Restore the original database trigger after CDC is enabled.

# Summary & Best Practices

- Always verify trigger impact before CDC operations.  
- Use Business Critical or supported tiers in Azure SQL.  
- Monitor CDC performance and clean up old change data.

Samples :

DISABLE TRIGGER trg\_ddl\_table\_changes ON DATABASE;

EXEC sys.sp\_cdc\_enable\_db;

EXEC sys.sp\_cdc\_enable\_table @source\_schema = N'dbo', @source\_name = N'organizations\_people', @role\_name = NULL;

EXEC sys.sp\_cdc\_enable\_table @source\_schema = N'dbo', @source\_name = N'policy', @role\_name = NULL;

ENABLE TRIGGER trg\_ddl\_table\_changes ON DATABASE;