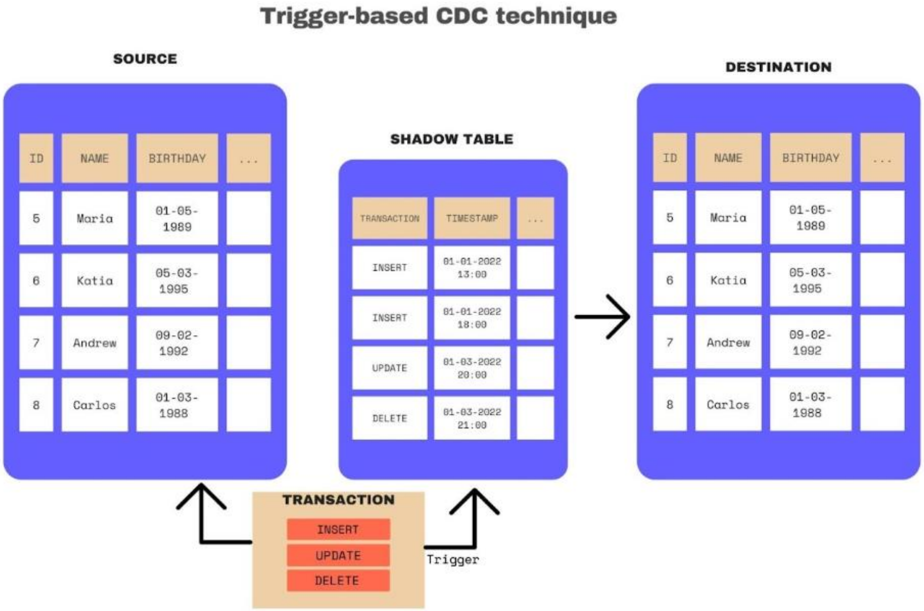
Change Data Capture:

Change Data Capture (CDC) is a technique used in ETL (Extract, Transform, Load) processes to identify and capture changes made to source data since the last ETL run. This allows you to efficiently update the target system with only the changes, reducing the amount of data transferred and processed. Here's a basic overview of implementing CDC in an ETL process:

Example 1:



Trigger-based CDC technique

SOURCE

ID	NAME	BIRTHDAY	...
5	Maria	01-05-1989	
6	Katia	05-03-1995	
7	Andrew	09-02-1992	
8	Carlos	01-03-1988	

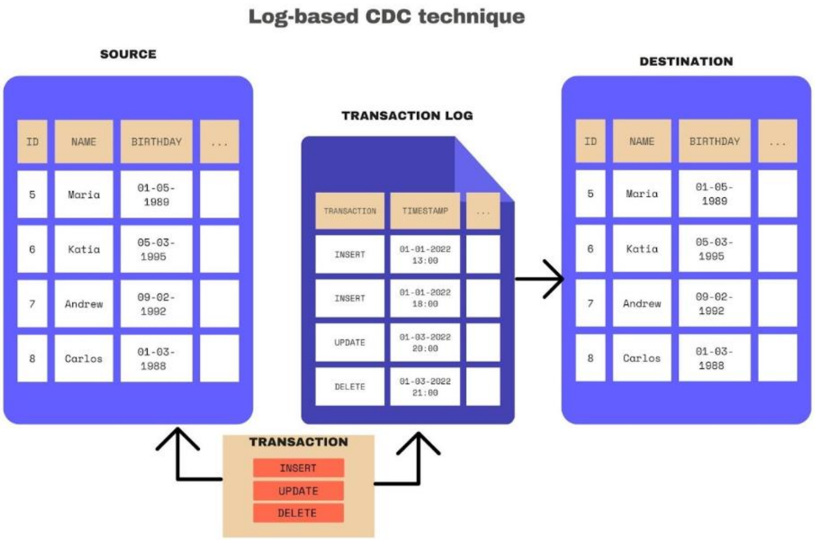
SHADOW TABLE

TRANSACTION	TIMESTAMP	...
INSERT	01-01-2022 13:00	
INSERT	01-01-2022 18:00	
UPDATE	01-03-2022 20:00	
DELETE	01-03-2022 21:00	

DESTINATION

ID	NAME	BIRTHDAY	...
5	Maria	01-05-1989	
6	Katia	05-03-1995	
7	Andrew	09-02-1992	
8	Carlos	01-03-1988	

Example 2:



Log-based CDC technique

SOURCE

ID	NAME	BIRTHDAY	...
5	Maria	01-05-1989	
6	Katia	05-03-1995	
7	Andrew	09-02-1992	
8	Carlos	01-03-1988	

TRANSACTION LOG

TRANSACTION	TIMESTAMP	...
INSERT	01-01-2022 13:00	
INSERT	01-01-2022 18:00	
UPDATE	01-03-2022 20:00	
DELETE	01-03-2022 21:00	

DESTINATION

ID	NAME	BIRTHDAY	...
5	Maria	01-05-1989	
6	Katia	05-03-1995	
7	Andrew	09-02-1992	
8	Carlos	01-03-1988	

Log-based CDC uses the transaction logs that some databases – such as PostgreSQL, MySQL, SQL Server, and Oracle – implement natively as part of their core functionality.

NOTE: Log-based and trigger-based CDC are very similar – both keep a log of changes every time a database operation happens – so the shadow table and the transaction log contain the same information. The difference between log-based and trigger-based CDC is that the first one uses a core functionality of the database (transaction log); meanwhile, the triggers are created and defined by the user.

Steps for Change Data Capture in ETL:

1. **Identify Source System Changes: **

- Determine how you will identify changes in the source system. Common methods include using timestamps, sequence numbers, flags, or a combination of these.

2. **Capture Initial State: **

- During the first ETL run, capture the initial state of the data in the source system. This involves extracting all relevant data and storing it in the target system.

3. **Record Changes: **

- For subsequent ETL runs, identify and record changes that have occurred in the source system since the last run. This is typically done by comparing the current state of the data with the captured initial state.

4. **Use CDC Mechanism: **

- Implement a CDC mechanism to efficiently track changes. This may involve using database features like database triggers, change tables, or utilizing log files.

5. **Extract Changed Data: **

- Extract only the changed data from the source system. This can be done by querying the source system based on the identified changes.

6. **Transform Data: **

- Apply any necessary transformations to the changed data based on business rules or requirements.

7. **Load into Target System: **

- Load the transformed data into the target system. This can be done by updating existing records or inserting new ones, depending on the nature of the change.

8. **Update CDC State: **

- Update the CDC state in the target system to reflect the current state of the data in the source system. This involves updating the captured state to be used in the next ETL run.

Advantages of Change Data Capture:

1. **Efficiency: **

- Reduces the amount of data transferred and processed, leading to more efficient ETL processes.

2. **Real-time or Near-real-time Updates: **

- Allows for real-time or near-real-time updates, ensuring that the target system is kept in sync with the source system.

3. **Reduced Resource Consumption: **

- Minimizes resource consumption by focusing on changes rather than processing the entire dataset.

4. **Improved Data Accuracy: **

- Enhances data accuracy by capturing changes at a granular level, reducing the risk of errors introduced by reprocessing unchanged data.

Note: Implementing Change Data Capture requires careful planning and consideration of the specific requirements and characteristics of your data and systems. The choice of CDC mechanism, such as database triggers, timestamp columns, or log-based CDC, will depend on the technology stack and the nature of the source systems.