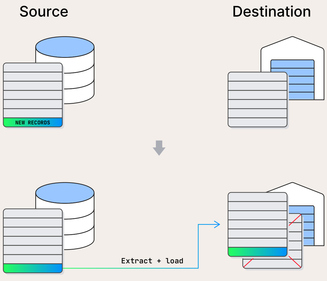
Fivetran

How incremental updates work

Incremental updates are a key feature of an efficient, high-performing data pipeline. Instead of extracting and loading the entire data source with every sync, incremental updates involve detecting new or modified records and reproducing the changes at the destination.

Full syncs

Copy and replace the entire data set



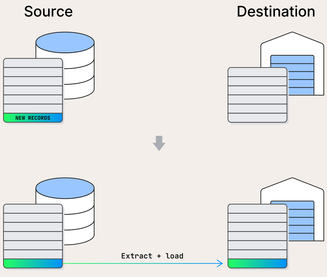
You must use full syncs to capture the full data set during an initial sync. They are also occasionally necessary to fix corrupted records or resolve other data integrity issues.

However, full syncs have the following drawbacks:

1. They often take a long time, especially for large data sources. This makes full syncs incapable of supporting real-time updates.
2. The large volume of data moved can bog down both the data source and the destination, consuming resources otherwise needed for operations and analytics.
3. They consume excessive network bandwidth and compute expense.

Incremental updates

Only add new records



Incremental updates require the ability to identify changes made to a previous state of the source. This practice is often referred to as change **data capture** (CDC). There are several methods of change data capture, but one method commonly used to sync data from databases is the use of logs. At very small increments or batch sizes, CDC enables real-time or streaming updates.

Walking through log-based incremental updates

The sync workflow looks like so:

1. Import a full table using a SELECT* query. This initial sync provides a snapshot of the table's entire contents, primary keys and all, at a particular moment in time. Make note of the primary key column; this will later be essential for designating the correct updates and preventing duplicate or conflicting records. Suppose we import a table (right) at 2022-07-30 13:23:04.

id	username	status
1	“D_artagnan”	“inactive”
34	“Goeman”	“active”
45	“Ichabod_Crane”	“active”
94	“jon_rico”	“inactive”
101	“Drosselmeyer”	“active”

2. After an interval of time has elapsed, query the change log and identify all the changes that occurred since your import. Our update corresponds with all transactions before transaction_id = 3, so let's start from there.

transaction_id	transaction	timestamp_completed	table	row_id	column	value
1	INSERT	2022-07-30 13:23:02	user	45	status	“active”
2	UPDATE	2022-07-30 13:23:03	user	45	username	“Ichabod_Crane”
3	UPDATE	2022-07-30 17:45:30	user	1	status	“active”
4	DELETE	2022-07-30 17:59:03	user	34	NULL	NULL
5	DELETE	2022-07-30 18:12:59	user	94	NULL	NULL
6	INSERT	2022-07-30 18:23:03	user	13	status	“inactive”
7	INSERT	2022-07-30 18:23:03	user	13	username	“capt_ahab”

3. Start reading the change log from transaction_id = 3. As you progress through the change log, based on unique row identifiers:
   - Merge updates with the corresponding records in the destination
   - Add new records as new rows with new primary keys
   - Remove deleted records from the destination

id	username	status
1	“D_artagnan”	“active”
13	“capt_ahab”	“inactive”
101	“Drosselmeyer”	“active”
45	“Ichabod_Crane”	“inactive”

Incremental data syncs depend on idempotence, which you can learn about in another infographic. Aside from log-based updates, another common method of incremental data syncs involves using timestamps.

Incremental updating is a key pillar of how Fivetran implements data movement.