

```

import pandas as pd
import pyarrow as pa
import datetime as dt
import os
import duckdb

from deltalake import DeltaTable, write_deltalake
from dotenv import load_dotenv
from pathlib import Path

conn = duckdb.connect()

bucket_name = "forums-analyses-bucket"
object_name = "raw_youtube_videos_comments"
folder_name = ""

DELTA_VIDEOS_COMMENTS_PATH = os.path.join("s3://", bucket_name,
folder_name, object_name).replace("\\", "/")

# load env variables
env_dir = Path('..../').resolve()
load_dotenv(os.path.join(env_dir, '.env'))

# get env variables
aws_creds = {
    "access_key": os.environ.get("AWS_ACCESS_KEY_ID"),
    "secret_key": os.environ.get("AWS_SECRET_ACCESS_KEY"),
    "region": os.environ.get("AWS_REGION_NAME"),
}

```

inspect potential duplicates that may have been scraped in the data extractor scripts

[0, comment, fjOeJssZX_Q, UgxHZSCawcylfpfa5dZ4AaABA, UC4sfyOmp6fQ1TOo_POiRYQQ, UCnLuLSV-Oi0ctqjxGgxFlmg, None, They are good friends, They are good friends, 2025-12-24T07:15:18Z, 2025-12-24T07:15:18Z, 0.0, @chritianjohn9701, <http://www.youtube.com/@chritianjohn9701>, 2025-12-25 02:04:33.671992] [1, comment, fjOeJssZX_Q, UgxHZSCawcylfpfa5dZ4AaABA, UC4sfyOmp6fQ1TOo_POiRYQQ, UCnLuLSV-Oi0ctqjxGgxFlmg, None, They are good friends, They are good friends, 2025-12-24T07:15:18Z, 2025-12-24T07:15:18Z, 0.0, @chritianjohn9701, <http://www.youtube.com/@chritianjohn9701>, 2025-12-25 02:04:33.671992] [2, comment, fjOeJssZX_Q, UgxHZSCawcylfpfa5dZ4AaABA, UC4sfyOmp6fQ1TOo_POiRYQQ, UCnLuLSV-Oi0ctqjxGgxFlmg, None, They are good friends, They are good friends, 2025-12-24T07:15:18Z, 2025-12-24T07:15:18Z, 0.0, @chritianjohn9701, <http://www.youtube.com/@chritianjohn9701>, 2025-12-25 02:04:33.671992]

may have all be potentially scraped as duplicate records and if records like them don't exist in the delta table then delta inserts all these as distinct records even if we define a predicate of

comment_id being the key that delta must use to check for existing records to make necessary updates or inserts.

```
delta_videos_comments_table = DeltaTable(DELTA_VIDEOS_COMMENTS_PATH,  
storage_options=aws_creds)  
  
videos_comments_pa_table =  
delta_videos_comments_table.to_pyarrow_table()  
  
conn.sql("""  
    SELECT * FROM videos_comments_pa_table  
""")
```

level	video_id	comment_id	channel_id_where_comment_was_made	text_display
author_channel_id		channel_id	where_comment_was_made	
parent_comment_id		text_original		text_display
published_at		updated_at		like_count
author_display_name			author_channel_url	
added_at				
varchar	varchar	varchar	varchar	varchar
varchar	varchar	varchar	varchar	varchar
varchar	varchar	int64	varchar	varchar
varchar			timestamp	
comment	fj0eJssZX_Q	UgxHZSCawcylfpfa5dZ4AaABAg		
UC4sfy0mp6fQ1T0o_P0iRYQQ		UCnLuLSV-0i0ctqjxGgxFlmg		NULL
They are good friends		They are good friends		2025-12-24T07:15:18Z
2025-12-24T07:15:18Z			0	@chritianjohn9701
http://www.youtube.com/@chritianjohn9701			2025-12-25	
02:04:33.671992				
comment	fj0eJssZX_Q	UgxHZSCawcylfpfa5dZ4AaABAg		
UC4sfy0mp6fQ1T0o_P0iRYQQ		UCnLuLSV-0i0ctqjxGgxFlmg		NULL
They are good friends		They are good friends		2025-12-24T07:15:18Z
2025-12-24T07:15:18Z			0	@chritianjohn9701
http://www.youtube.com/@chritianjohn9701			2025-12-25	
02:04:33.671992				
comment	fj0eJssZX_Q	UgxHZSCawcylfpfa5dZ4AaABAg		
UC4sfy0mp6fQ1T0o_P0iRYQQ		UCnLuLSV-0i0ctqjxGgxFlmg		NULL
They are good friends		They are good friends		2025-12-24T07:15:18Z
2025-12-24T07:15:18Z			0	@chritianjohn9701

| http://www.youtube.com/@chritianjohn9701 | 2025-12-25
02:04:33.671992 |
| comment | fj0eJssZX_Q | UgxHZSCawcylfpfa5dZ4AaABAgn |
UC4sfy0mp6fQ1T0o_P0iRYQQ | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| They are good friends | They are good friends | 2025-12-
24T07:15:18Z | 2025-12-24T07:15:18Z | 0 | @chritianjohn9701
| http://www.youtube.com/@chritianjohn9701 | 2025-12-25
02:04:33.671992 |
| comment | fj0eJssZX_Q | UgxHZSCawcylfpfa5dZ4AaABAgn |
UC4sfy0mp6fQ1T0o_P0iRYQQ | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| They are good friends | They are good friends | 2025-12-
24T07:15:18Z | 2025-12-24T07:15:18Z | 0 | @chritianjohn9701
| http://www.youtube.com/@chritianjohn9701 | 2025-12-25
02:04:33.671992 |
| comment | fj0eJssZX_Q | UgxHZSCawcylfpfa5dZ4AaABAgn |
UC4sfy0mp6fQ1T0o_P0iRYQQ | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| They are good friends | They are good friends | 2025-12-
24T07:15:18Z | 2025-12-24T07:15:18Z | 0 | @chritianjohn9701
| http://www.youtube.com/@chritianjohn9701 | 2025-12-25
02:04:33.671992 |
| comment | fj0eJssZX_Q | UgxHZSCawcylfpfa5dZ4AaABAgn |
UC4sfy0mp6fQ1T0o_P0iRYQQ | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| They are good friends | They are good friends | 2025-12-
24T07:15:18Z | 2025-12-24T07:15:18Z | 0 | @chritianjohn9701
| http://www.youtube.com/@chritianjohn9701 | 2025-12-25
02:04:33.671992 |
| comment | fj0eJssZX_Q | UgxHZSCawcylfpfa5dZ4AaABAgn |
UC4sfy0mp6fQ1T0o_P0iRYQQ | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| They are good friends | They are good friends | 2025-12-
24T07:15:18Z | 2025-12-24T07:15:18Z | 0 | @chritianjohn9701
| http://www.youtube.com/@chritianjohn9701 | 2025-12-25
02:04:33.671992 |
| comment | fj0eJssZX_Q | UgxHZSCawcylfpfa5dZ4AaABAgn |
UC4sfy0mp6fQ1T0o_P0iRYQQ | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| They are good friends | They are good friends | 2025-12-
24T07:15:18Z | 2025-12-24T07:15:18Z | 0 | @chritianjohn9701
| http://www.youtube.com/@chritianjohn9701 | 2025-12-25
02:04:33.671992 |
| . | . | . | . | . | . |

| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-

```

23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| comment | fj0eJssZX_Q | UgxupW6MBKCmT-f2ANN4AaABAg | UCOj-
NFBzRrlGopoh3HFDdgg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
| Nice good morning and nice | Nice good morning and nice | 2025-12-
23T04:55:14Z | 2025-12-23T04:55:14Z | 0 | @ashalohar998
| http://www.youtube.com/@ashalohar998 | 2025-12-25
02:04:34.343747 |
| ? rows (>9999 rows, 20 shown)
14 columns |

```

So if I had say ff. records:

```

| level | videoid | comment_id | added_at |
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.300
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.300
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.500
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.750
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.750

```

I would want to write a query that would delete the records with 2025-01-01 00:00:00.300, 2025-01-01 00:00:00.500, and 2025-01-01 00:00:00.750 as their added_at timestamp while retaining only one of the most recent record which in this case is 2025-01-01 00:00:00.750

There are about 3 ways to approach deleting these duplicate rows with these certain timestamps

1. using a ROW_NUMBER() to be able to retain the record with the maximum added_at timestamp as this is the most recent and will do so regardless if it has duplicates

```
| level | videoid | comment_id | added_at | most_recent_rank
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.300, 5
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.300, 4
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.500, 3
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.750, 2
reply, somevideoid1, somecommentid1, 2025-01-01 00:00:00.750, 1
```

here we want to filter and retain the record with the most_recent_rank of 1 since it is the most recent and we want to keep it in our table. This moreover will not only retain one record of the duplicate record but all records that do not have a duplicate record since they are the only ones and will have a most_recent_rank of 1

once done we can execute an insert overwrite transaction of the whole table.

```
-- 2. Overwrite your original table with the clean data
INSERT OVERWRITE your_delta_table_name
SELECT * FROM clean_comments;
```

1. but the above is much more costly and overwriting 4m rows or more is not a good idea. So another is through a 2 way transaction. First is where we select the duplicate records

identifying comment_ids with duplicates

```
conn.sql("""
    CREATE OR REPLACE TEMPORARY TABLE duplicate_comments AS (
        SELECT
            comment_id,
            COUNT(comment_id) AS n_duplicates
        FROM videos_comments_pa_table
        GROUP BY comment_id
        HAVING COUNT(comment_id) > 1
        ORDER BY n_duplicates DESC
    );
""")
```

```
conn.sql("SELECT * FROM duplicate_comments")
```

comment_id varchar	n_duplicates int64
UgzUwGsxgcpfwQcdVrJ4AaABA	12500
Ugw0mWXyiCv_G4K1ch54AaABA	12500
UgxBz7EV3q2lYeNtbKt4AaABA	12500
UgyQu70_1xaIcwgIuAF4AaABA	12500

UgwCEQ5Acjz78Neh9UR4AaABA	12500
Ugy7pv-I9DEMGfh3uYV4AaABA	12500
UgzIg8wfwHx_AF93omN4AaABA	12500
Ugx_dWL334dp0YZ_8hd4AaABA	12500
UgzZA4w7HiC5f4Qh7bR4AaABA	12500
UgwdaNFtxB86CGyCQJ54AaABA	12500
.	.
.	.
.	.
Ugyfc-e3p5tIQxKuSWB4AaABA	3
UgxnmMIV3q0S8Q0VrqF4AaABA	3
UgwzrRAMItkMmGlmyp4AaABA	3
UgyBZU7lbPZCOpNG9ol4AaABA	3
Ugx0Zyi-v1qgbWhFRl4AaABA	3
UgwjdnmM3Cbg_76k8Mt4AaABA	3
UgwRf9S6cxH5vPeww8l4AaABA	3
UgxSMQ6Uy69Jv_v0HMV4AaABA	3
UgwFSTm4P3RCdA1JPgl4AaABA	3
Ugxb58411DSukVjNkW94AaABA	3

4944 rows (20 shown) 2 columns

extracting strictly only one among records with duplicates that are the most recent

```
conn.sql("""
    CREATE OR REPLACE TEMPORARY TABLE ranked_comments AS (
        SELECT
            *,
            ROW_NUMBER() OVER(
                PARTITION BY comment_id
                ORDER BY added_at DESC
            ) AS most_recent_rank
        FROM videos_comments_pa_table
        WHERE comment_id IN (SELECT comment_id FROM
duplicate_comments)
    );
""")

conn.sql("""
    SELECT
        *
    FROM ranked_comments
    WHERE comment_id LIKE 'UgyzlVOY0j9Uy3Fr04l4AaABA'
    ORDER BY most_recent_rank ASC;
""")
```


http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:50.438585 |
241 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:50.117556 |
242 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:49.921235 |
243 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:49.722527 |
244 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:49.465147 |
245 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:49.164713 |
246 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:48.705347 |
247 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:48.343736 |
248 |
| comment | 8y-ViyUk7Dk | Ugyz1V0Y0j9Uy3Fr04l4AaABAg |
UCx97Zhd9vRq93FGJ0Fn8vRA | UC8-Th83bH_thdKZDJCrn88g | NULL
| So sweet | So sweet | 2025-12-24T18:15:24Z | 2025-12-
24T18:15:24Z | 0 | @chefvickyyy |
http://www.youtube.com/@chefvickyyy | 2025-12-25 02:23:47.946027 |
249 |

comment	8y-ViyUk7Dk	Ugyz1V0Y0j9Uy3Fr04l4AaABAg		
UCx97Zhd9vRq93FGJ0Fn8vRA	UC8-Th83bH_thdKZDJCrn88g			NULL
So sweet	So sweet	2025-12-24T18:15:24Z	2025-12-	
24T18:15:24Z	0	@chefvickyy		
http://www.youtube.com/@chefvickyy	2025-12-25	02:23:47.415837		
250				

250 rows (20 shown)
15 columns

```
conn.sql("""
    CREATE OR REPLACE TEMPORARY TABLE most_recent_dups AS (
        SELECT
            * EXCLUDE(most_recent_rank)
        FROM ranked_comments
        WHERE most_recent_rank = 1
    );
""")
```

```
conn.sql("SELECT * FROM most_recent_dups")
```

level	video_id	comment_id
author_channel_id	channel_id_where_comment_was_made	
parent_comment_id		
text_original		
text_display		
published_at	updated_at	like_count
author_display_name		
author_channel_url		
added_at		
varchar	varchar	varchar
	varchar	varchar

varchar				varchar	int64	varchar
varchar	varchar		varchar	varchar		varchar
	timestamp					
comment GE4S_68R0cg UgzX7Yjm-1Bqy8-9YLF4AaABA UCULXWZalE7KRT4-UESAHDpg UCGapJkN0gRQhzWj6wFKd9ew						NULL
Mystery is big fat because he always takes food from zoey😊😊						
Mystery is big fat because he always takes food from zoey😊😊						
2025-12-08T06:52:31Z 2025-12-08T06:52:31Z 1						
@pitrisetia6554 http://www.youtube.com/@pitrisetia6554						
2025-12-25 02:43:08.438308						
comment BS1fQWpCFco UgzXFfG4kqcPhbjd2kd4AaABA UCPxLLUJSyFiJ088b3c2Np-g UCX0D3opIyp2M9nd3B28sk6A						NULL
Un saludo desde argentina,y literalmente						
perfectas❤️❤️❤️❤️❤️❤️❤️ Un saludo desde argentina,y						
literalmente perfectas❤️❤️❤️❤️❤️❤️ 2025-12-21T18:17:07Z						
2025-12-21T18:17:07Z 1 @alejandroolivera3882						
http://www.youtube.com/@alejandroolivera3882						
2025-12-25 02:12:34.065923						
comment iqxq16AzJDw UgzYW00hdncYAZxKA7Z4AaABA UC_dosPZ0r0fm2Fbd-NkNG_w UCVeGs02tj9YewfNi7xxabqA						NULL
So cute and amazing guys ❤️						
So cute and amazing guys ❤️						
2025-12-05T16:21:53Z 2025-12-05T16:21:53Z 0						
@dreammodeon131 http://www.youtube.com/@dreammodeon131						
2025-12-25 02:40:35.418459						
comment 5iA6bYz5FjM UgzZHR3es74IAP8BIzB4AaABA UCusVpnP0vYDZlTXo3AIa8ww UCdgERiTAAvvQmhvSvo_09WA						NULL
♥️♥️♥️♥️♥️♥️♥️♥️♥️♥️♥️♥️rumi						
♥️♥️♥️♥️♥️♥️♥️♥️♥️♥️♥️♥️rumi						
2025-11-24T16:27:22Z 2025-11-24T16:27:22Z 0						
@PaoloTufano-h7y http://www.youtube.com/@PaoloTufano-h7y						
2025-12-25 02:17:37.175164						
comment TbMEMCvFbZk UgzaQtngvLMqtTmJQct4AaABA UC7gEFRSZ4IQqfLo3fAxWlPw UCnLuLSV-0i0ctqjxGgxFlmg						NULL
██████████						
██████████						
2025-12-22T11:00:09Z 2025-12-22T11:00:09Z 0						
@IsmaelLlovido-g5e http://www.youtube.com/@IsmaelLlovido-g5e						

reply | xorPL4rr2x8 |
Ugwm0FB1tuIH9hRfNyJ4AaABAgnANEw1ZwXmouAPxQNj_I9sP |
UCwsDGRJQdejr0NnnnG9vsdQ | UC2sxxXRBL5SgY0fI58Br0CA
Ugwm0FB1tuIH9hRfNyJ4AaABAgn | هدوج وتهج
-11-2025 | هدوج وتهج
25T14:36:54Z | 2025-11-25T14:36:54Z | 0 | @6@ملها حساس-و
| http://www.youtube.com/@%D9%85%D9%84%D9%83%D9%87%D8%A7%D8%AD
%D8%B3%D8%A7%D8%B3-%D9%886%D8%AB | 2025-12-25 02:47:08.402369 |
comment | QGsevnBItdU | UgwpRhtgges_VRIDQ714AaABAgn
UCDSw5-rGdd4-LtiMGMU70hg | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
I love 1:34
I love <a href="https://www.youtube.com/watch?
v=QGsevnBItdU&t=94">1:34 | 2025-12-23T14:24:14Z | 0 | 2025-12-
23T14:24:14Z | 2025-12-23T14:24:14Z | 0 | @louj388
http://www.youtube.com/@louj388
2025-12-24 15:36:03.794422 |
comment | 3JTVQTK36R8 | UgwphZW7havq-cZMR7x4AaABAgn
UCR1Zv6dxQeMQ8bFCwynYJA | UCnLuLSV-0i0ctqjxGgxFlmg | NULL
uhull
uhull
2025-12-24T22:33:00Z | 2025-12-24T22:33:00Z | 0 |
@helologuedes | http://www.youtube.com/@helologuedes
2025-12-25 02:23:46.901319 |
comment | 9e9t3VdAHLw | UgwqQFneuf3bBbXMntR4AaABAgn
UCEJvmDYVdMwOoHp1f0tpc7A | UCroNr00068n25IqSNapMK8w | NULL
La voix de Rumi est superbe ☺♥♥
La voix de Rumi est superbe ☺♥♥
2025-12-24T16:57:54Z | 2025-12-24T16:57:54Z | 1 |
@JudithMbahn | http://www.youtube.com/@JudithMbahn
2025-12-25 02:38:00.298138 |
comment | nmt594aKD9U | UgwrIGfueo97es9eutx4AaABAgn
UCa-ZInvN0VKLzkFb805pvfQ | UC5ZiUaIJ2b5dYBYGf5iEUrA | NULL
Sovaco
Sovaco
2025-12-23T21:06:41Z | 2025-12-23T21:06:41Z | 0 |
@RoxanaRiquelmevenegas | http://www.youtube.com/@RoxanaRiquelmevenegas
2025-12-24 15:36:17.715631 |
comment | ybNrUQ1JMGE | UgwrJJWuwc6UBJKNYj14AaABAgn
UC1SiYNaNMyACnVUa2HeVX9g | UCAQ0f_uPOJ6rX0_k9RwfK3A | NULL
Didn't see that coming
Didn't see that coming

2025-12-20T19:56:51Z	2025-12-20T19:56:51Z	0	
@HIBAMughal-d5t	http://www.youtube.com/@HIBAMughal-d5t		
2025-12-24 15:16:23.814373			
comment 8y-ViyUk7Dk UgwsoNN3xDyc5S9uBmN4AaABAq			
UCj8kz4t_34NWXLQ3_b113YQ UC8-Th83bH_thdKZDJCrn88g			NULL
Lagu teh meni enak kieu Duh Nuhun Tetehe Love U all Pokok na			
mah Lagu teh meni enak kieu Duh Nuhun Tetehe			
tetehe Love U all Pokok na mah 2025-12-			
23T10:39:48Z 2025-12-23T10:39:48Z 1 @AriIfin-w7f			
http://www.youtube.com/@AriIfin-w7f			
2025-12-25 02:25:07.619227			
comment xorpL4rr2x8 UgwtiCHTLsNy8l2hDwB4AaABAq			
UCutoUYlaUyyvmR0ts-hSaQg UC2sxxxXRBL5SgY0fI58Br0CA			NULL
eu adoro esta música ヽ(・▽・)ノ♥			
eu adoro esta música ヽ(・▽・)ノ♥			
2025-10-12T21:57:51Z 2025-10-12T21:57:51Z 1			
@VitóriaNdami http://www.youtube.com/@Vit%C3%B3riaNdami			
2025-12-25 02:47:08.402264			
comment 3JTVQTK36R8 UgwwGd26Z066bxQ_QNp4AaABAq			
UCi25DsJmHQLxg1K-aIAoXKA UCnLuLSV-0i0ctqjxGgxFlmg			NULL
its December			
its December			
2025-12-18T23:24:17Z 2025-12-18T23:24:17Z 0			
@MAPSofSTARS23 http://www.youtube.com/@MAPSofSTARS23			
2025-12-25 02:23:46.901458			
comment cWppAbqm9I8 UgwyOrP6EUcQPZXdn2t4AaABAq			
UCXsGSRZaAJ4P2zu6LDnAEzg UCnLuLSV-0i0ctqjxGgxFlmg			NULL
130 m			
130 m			
2025-12-24T06:23:08Z 2025-12-24T06:23:08Z 0			
@liamosorio3819 http://www.youtube.com/@liamosorio3819			
2025-12-25 02:33:50.510682			

4944 rows (20 shown)
14 columns

because we have now extracted both the set of duplicate records and these duplicate records most recent record versions, we will now commence with the first stage transaction: the deletion of the duplicate records first.

And rather than an overwrite operation we instead delete the duplicate records in our original delta table. To do this we need to convert the `duplicate_comments` temp table to a pyarrow table that delta can utilize

```
duplicate_comments_pa_table = conn.sql("SELECT * FROM
duplicate_comments").to_arrow_table()
duplicate_comments_pa_table

pyarrow.Table
comment_id: string
n_duplicates: int64
-----
comment_id:
[[{"comment_id": "UgzUwGsxgcpfwQcdVrJ4AaABA"}, {"comment_id": "Ugw0mWXyiCv_G4K1ch54AaABA"}, {"comment_id": "UgxBz7EV3q2lYeNtbKt4AaABA"}, {"comment_id": "UgyQu70_1xaIcwgIuAF4AaABA"}, {"comment_id": "UgwCEQ5Acjz78Neh9UR4AaABA"}, {"comment_id": "UgwjDnmM3Cbg_76k8Mt4AaABA"}, {"comment_id": "UgwRf9S6cxH5vPeww8l4AaABA"}, {"comment_id": "UgxSMQ6Uy69Jv_v0HMV4AaABA"}, {"comment_id": "UgwFSTm4P3RCdA1JPgl4AaABA"}, {"comment_id": "Ugxb58411DSukVjNkW94AaABA"}]
n_duplicates: [[12500, 12500, 12500, 12500, 12500, ..., 3, 3, 3, 3, 3]]

delta_videos_comments_table.merge(
    duplicate_comments_pa_table,
    predicate="source.comment_id = target.comment_id",
    source_alias="source",
    target_alias="target",
    merge_schema=True
).when_matched_delete(
).execute()

# this transaction is equivalent to
# ``
# MERGE INTO delta_videos_comments_table AS source
```

```
# USING duplicate_comments_pa_table AS target
# ON source.comment_id = target.comment_id
# WHEN MATCHED THEN
#     DELETE;
# ```

{'num_source_rows': 4944,
 'num_target_rows_inserted': 0,
 'num_target_rows_updated': 0,
 'num_target_rows_deleted': 4086841,
 'num_target_rows_copied': 417,
 'num_output_rows': 417,
 'num_target_files_scanned': 2,
 'num_target_files_skipped_during_scan': 0,
 'num_target_files_added': 1,
 'num_target_files_removed': 1,
 'execution_time_ms': 9358,
 'scan_time_ms': 9192,
 'rewrite_time_ms': 5}
```

```
videos_comments_pa_table_new =  
delta_videos_comments_table.to_pyarrow_table()
```

```
conn.sql("""
    SELECT * FROM videos_comments_pa_table_new
""")

```

level	video_id	comment_id
author_channel_id		channel_id_where_comment_was_made
parent_comment_id		
text_original		
text_display		
published_at		updated_at
author_display_name		like_count
added_at		author_channel_url
varchar	varchar	varchar
	varchar	varchar
		varchar
		varchar
		varchar

comment | f1faUvBlpQA | UgycTzCig6Uwif5Bepx4AaABAg
UC2qqr865f-6W1-L9cM-ggXQ | UCE7KTbfpkNqhzeB_OL6Ycbg | NULL
Ai,is,getting,so,realistic,we're,cookedchat😊♥
Ai,is,getting,so,realistic,we're,cookedchat😊♥
2025-12-25T01:56:11Z | 2025-12-25T01:56:11Z | 0 |
@SrianjarWulan | http://www.youtube.com/@SrianjarWulan
2025-12-25 02:52:05.581063 |
comment | f1faUvBlpQA | UgzoSyjK3n2iHJ00SpF4AaABAg
UCiKjDmnf70x4JwWtVfRVRdA | UCE7KTbfpkNqhzeB_OL6Ycbg | NULL
OMG I HATE AI
OMG I HATE AI
2025-12-24T01:57:48Z | 2025-12-24T01:57:48Z | 0 |
@PranshuJani | http://www.youtube.com/@PranshuJani
2025-12-25 02:52:05.581099 |
comment | f1faUvBlpQA | UgxXsZXdfnFTyXXYbah4AaABAg
UCYzty6yCY8VByb0QSzYqDHw | UCE7KTbfpkNqhzeB_OL6Ycbg | NULL
Ty
Ty
2025-12-23T10:21:42Z | 2025-12-23T10:21:42Z | 0 |
@Yauda_Si3 | http://www.youtube.com/@Yauda_Si3
2025-12-25 02:52:05.58112 |
comment | f1faUvBlpQA | Ugyx2mnEXRUr-82maeF4AaABAg
UCcl4UBilMtL5Mz_xqIeTXVQ | UCE7KTbfpkNqhzeB_OL6Ycbg | NULL
rakokamoca♥♥♥♥
rakokamoca♥♥♥♥
2025-12-23T07:06:55Z | 2025-12-23T07:06:55Z | 0 |
@vistarizkianita1081 |
http://www.youtube.com/@vistarizkianita1081 | 2025-12-25
02:52:05.581122 |
comment | f1faUvBlpQA | Ugzw7whTuk0ABQtWJ6x4AaABAg
UCHCrQqhX0anzRci3zfudFNw | UCE7KTbfpkNqhzeB_OL6Ycbg | NULL
Das ist krass😊😊😊😊😊😊😊
Das ist krass😊😊😊😊😊😊😊


```
| 418 rows (20 shown)
14 columns
```

```
conn.sql("""
    SELECT
        comment_id,
        COUNT(comment_id) AS n_duplicates
    FROM videos_comments_pa_table_new
    GROUP BY comment_id
    HAVING COUNT(comment_id) > 1
    ORDER BY n_duplicates DESC;
""")
```

comment_id	n_duplicates
varchar	int64

0 rows

Once we finish the delete transaction and check if there are still duplicate comments, right after we will now get to the second stage of our transaction

which is reinserting the most recent version/record of those removed duplicate records we have saved as a temporary table earlier and which we now need to convert to a pyarrow table

```
most_recent_dups_pa_table = conn.sql("SELECT * FROM
most_recent_dups").to_arrow_table()
most_recent_dups_pa_table
```



```

perfectas𠂇𠂇𠂇𠂇 ( ... 40 chars omitted)", "So cute and amazing
guys ❤", "❤️❤️❤️❤️❤️❤️❤️rumi", "𠂇𠂇𠂇", ..., "Didnt see that
coming", "Lagu teh meni enak kieu Duh Nuhun Teteh teteh Love U all
Pokok na mah", "eu adoro esta música 𠂇𠂇𠂇", "its December", "130
m"]]
published_at: [[{"2025-12-08T06:52:31Z", "2025-12-21T18:17:07Z", "2025-
12-05T16:21:53Z", "2025-11-24T16:27:22Z", "2025-12-
22T11:00:09Z", ..., "2025-12-20T19:56:51Z", "2025-12-23T10:39:48Z", "2025-
10-12T21:57:51Z", "2025-12-18T23:24:17Z", "2025-12-24T06:23:08Z"]]
updated_at: [[{"2025-12-08T06:52:31Z", "2025-12-21T18:17:07Z", "2025-12-
05T16:21:53Z", "2025-11-24T16:27:22Z", "2025-12-22T11:00:09Z", ..., "2025-
12-20T19:56:51Z", "2025-12-23T10:39:48Z", "2025-10-12T21:57:51Z", "2025-
12-18T23:24:17Z", "2025-12-24T06:23:08Z"]]
...
delta_videos_comments_table.merge(
    most_recent_dups_pa_table,
    predicate="target.comment_id = source.comment_id",
    source_alias="source",
    target_alias="target",
    merge_schema=True
).when_matched_update(
    updates={
        # these are not included as these are the composite keys
        # that are not by good practice supposed to be updated
        "level" : "source.level",
        # "video_id" : "source.video_id",
        # "comment_id" : "source.comment_id",
        "author_channel_id" : "source.author_channel_id",
        "channel_id_where_comment_was_made" :
"source.channel_id_where_comment_was_made",
        "parent_comment_id" : "source.parent_comment_id",
        "text_original" : "source.text_original",
        "text_display" : "source.text_display",
        # "published_at" : "source.published_at",
        "updated_at" : "source.updated_at",
        "like_count" : "source.like_count",
        "author_display_name" : "source.author_display_name",
        "author_channel_url" : "source.author_channel_url",
        "added_at": "source.added_at",
    },
    # this tells delta to only update a record if the new record
    # does indeed have changed its column values when compared to the
    # current record
    predicate="""
        (source.level IS DISTINCT FROM target.level) OR
        (source.author_channel_id IS DISTINCT FROM
target.author_channel_id) OR
        (source.channel_id_where_comment_was_made IS DISTINCT FROM

```

```

target.channel_id_where_comment_was_made) OR
    (source.parent_comment_id IS DISTINCT FROM
target.parent_comment_id) OR
    (source.text_original IS DISTINCT FROM target.text_original)
OR
    (source.text_display IS DISTINCT FROM target.text_display) OR
    (source.updated_at > target.updated_at) OR
    (source.like_count IS DISTINCT FROM target.like_count) OR
    (source.author_display_name IS DISTINCT FROM
target.author_display_name) OR
    (source.author_channel_url IS DISTINCT FROM
target.author_channel_url) OR

    (source.added_at IS DISTINCT FROM target.added_at)
"""

# (source.video_id IS DISTINCT FROM target.video_id) OR
# (source.comment_id IS DISTINCT FROM target.comment_id) OR
# (source.published_at IS DISTINCT FROM target.published_at) OR
).when_not_matched_insert_all()\n
.execute()

{'num_source_rows': 4944,
 'num_target_rows_inserted': 4944,
 'num_target_rows_updated': 0,
 'num_target_rows_deleted': 0,
 'num_target_rows_copied': 0,
 'num_output_rows': 4944,
 'num_target_files_scanned': 2,
 'num_target_files_skipped_during_scan': 0,
 'num_target_files_added': 1,
 'num_target_files_removed': 0,
 'execution_time_ms': 3707,
 'scan_time_ms': 3687,
 'rewrite_time_ms': 3}

videos_comments_pa_table_new =
delta_videos_comments_table.to_pyarrow_table()

conn.sql("SELECT * FROM videos_comments_pa_table_new")

```


comment | f1faUvBlpQA | UgycTZCig6Uwif5Bepx4AaABAgn | UC2qqr865f-6W1-L9cM-ggXQ | UCE7KTbfpkNqhzeB_0L6Ycbg | NULL
Ai, is, getting, so, realistic, we're, cookedchat😊
Ai, is, getting, so, realistic, we're, cookedchat😊
2025-12-25T01:56:11Z | 2025-12-25T01:56:11Z | 0 |
@SrianjarWulan | http://www.youtube.com/@SrianjarWulan
2025-12-25 02:52:05.581063 |
comment | f1faUvBlpQA | UgzoSyjK3n2iHJ00SpF4AaABAgn |
UCiKjDmnf70x4JwWtVfRVRdA | UCE7KTbfpkNqhzeB_0L6Ycbg | NULL
OMG I HATE AI
OMG I HATE AI
2025-12-24T01:57:48Z | 2025-12-24T01:57:48Z | 0 |
@PranshuJani | http://www.youtube.com/@PranshuJani
2025-12-25 02:52:05.581099 |
comment | f1faUvBlpQA | UgxXsZXdfnFTyXXYbah4AaABAgn |
UCYzty6yCY8VByb0QSzYqDHw | UCE7KTbfpkNqhzeB_0L6Ycbg | NULL
Ty
Ty
2025-12-23T10:21:42Z | 2025-12-23T10:21:42Z | 0 |
@Yauda_Si3 | http://www.youtube.com/@Yauda_Si3
2025-12-25 02:52:05.58112 |
comment | f1faUvBlpQA | Ugyx2mnEXRUr-82maeF4AaABAgn |
UCcl4UBilMtL5Mz_xqIeTXVQ | UCE7KTbfpkNqhzeB_0L6Ycbg | NULL
rakokamoca♥♥♥♥
rakokamoca♥♥♥♥
2025-12-23T07:06:55Z | 2025-12-23T07:06:55Z | 0 |
@vistarizkianita1081 | http://www.youtube.com/@vistarizkianita1081
2025-12-25 02:52:05.581122 |
comment | f1faUvBlpQA | Ugzw7whTuk0ABQtWJ6x4AaABAgn |
UCHCrQqhX0anzRci3zfudFNw | UCE7KTbfpkNqhzeB_0L6Ycbg | NULL
Das ist krass😊😊😊😊😊😊
Das ist krass😊😊😊😊😊😊
2025-12-20T10:31:22Z | 2025-12-20T10:31:22Z | 1 |
@SimoneGrätz-f7f | http://www.youtube.com/@SimoneGr%C3%A4tz-f7f

2025-12-25 02:52:05.581206			
comment f1faUvBlpQA Ugw8q-u-iigf0zeSqA94AaABAgn	UCBJptTsmNGoeP6v6Y_rUzaA UCE7KTbfpkNqhzeB_0L6Ycbg	NULL	
That's AI			
That's AI			
2025-12-19T23:55:05Z 2025-12-19T23:55:05Z 0			
@AngelBernalPérez-i3p8d http://www.youtube.com/@AngelBernalP	%C3%A9rez-i3p8d 2025-12-25 02:52:05.581214		
comment yebNIHKAC4A UgxncH0HLfZktW2j3lZ4AaABAgn	UCvhrmXnbiWKJ5CDVdmrk0Sg UCnLuLSV-0i0ctqjxGgxFlmg	NULL	
I like golden ♥♥♥			
I like golden ♥♥♥			
2025-12-24T08:01:45Z 2025-12-24T08:01:45Z 0			
@shafiniaziniasri2630 http://www.youtube.com/@shafiniaziniasri2630	2025-12-24 16:13:39.52169		
comment yebNIHKAC4A UgyMcxevjLEn6a-et054AaABAgn	UC9HaGELaQJsCMgNg78IZziA UCnLuLSV-0i0ctqjxGgxFlmg	NULL	
한국말 가사가 더 매력.			
한국말 가사가 더 매력.			
2025-12-24T07:47:44Z 2025-12-24T07:47:44Z 0 @귀욤-n5v			
http://www.youtube.com/@%EA%B7%80%EC%9A%A4-n5v 2025-12-24	16:13:39.52169		
comment yebNIHKAC4A Ugxfkp1UXZ5wluJIqpR4AaABAgn	UCqmbMamGwvhtdrdmDnqxCUA UCnLuLSV-0i0ctqjxGgxFlmg	NULL	
Yoo I made an extra credit edit... I didn't get the extra credit :			
(https://www.youtube.com/watch?v=Mvza3J0rRyg&t=11s Yoo I made an			
extra credit edit... I didn't get the extra credit :(https://			
www.youtube.com/watch?v=Mvza3J0rRyg&t=11s 2025-12-			
24T01:46:41Z 2025-12-24T01:46:41Z 1 @ComedyDistracts101			
http://www.youtube.com/@ComedyDistracts101 2025-12-24	14:47:50.215823		
comment -8AHcNRfERI NULL NULL	NULL	NULL	
NULL NULL			
NULL NULL			
NULL NULL			
15:14:43.404799			2025-12-24
5362 rows (20 shown)			
14 columns			

```
conn.sql("""  
    SELECT  
        comment_id,  
        COUNT(comment_id) AS n_duplicates  
    FROM videos_comments_pa_table_new  
    GROUP BY comment_id  
    HAVING COUNT(comment_id) > 1  
    ORDER BY n_duplicates DESC;  
""")
```

comment_id	n_duplicates
varchar	int64
0 rows	

Reddit posts

```
DELTA_POSTS_PATH = "../data/raw_reddit_posts"  
delta_posts_table = DeltaTable(DELTA_POSTS_PATH)  
  
df1 = delta_posts_table.to_pandas()  
  
df1
```

post_id \	post_title	post_score
0 Kpop Demon Hunters is nominated for Best Anima...	RAH THEY LOOK PERFECT	608
1p6gyco	Ami Thompson appreciation post	3104
1or05al	Omg look at her ☺	1820
2los3fiy	I drew Mira! ☺	715
3los8xri	K-Pop Demon Hunters Nominated for 5 Grammys	579
4los7kjs		1247

```
loqywhk
```

```
          post_url    post_name \
0      https://i.redd.it/4zyoezehjf3g1.jpeg t3_1p6gyco
1      https://i.redd.it/rtlgsauzavzf1.jpeg t3_lor05al
2  https://www.reddit.com/gallery/los3fiy t3_los3fiy
3      https://i.redd.it/6vkapl1ki50g1.jpeg t3_los8xri
4  https://www.reddit.com/gallery/los7kjs t3_los7kjs
5 https://www.reddit.com/r/KpopDemonhunters/comm... t3_loqywhk

          post_author_name
post_body \
0   Maximum-Ask-1745

1           TLOU_1

2   Trillex_PL_Zykov  There won't be enough of a praise for all
the ...
3 Drawingandstuff2000

4       MayutArts

5       StrategicCarry * Song of the Year for Golden (EJAE and Mark
S...

          post_created_at    post_edited_at        added_at
0 2025-11-25 16:28:02 1970-01-01 00:00:00 2025-11-26 08:38:45.407172
1 2025-11-08 01:11:53 1970-01-01 08:00:00 2025-11-09 20:33:25.623074
2 2025-11-09 07:10:17 1970-01-01 08:00:00 2025-11-09 20:33:25.634669
3 2025-11-09 11:32:59 1970-01-01 08:00:00 2025-11-09 20:33:25.634669
4 2025-11-09 10:24:06 1970-01-01 08:00:00 2025-11-09 20:33:25.639084
5 2025-11-08 00:25:16 2025-11-08 00:27:11 2025-11-09 20:26:56.682074

test_posts_table = pa.table({
    "post_title": ["new title", "new title too", "another new title"],
    "post_score": [9999, 10000, 24548],
    "post_id": ["1p6gyco", "mira", "zoey"],
    "post_url": ["https://i.redd.it/a0ecmeswtizf1.jpeg",
    "https://some-url.com", "https://some-url.com"],
    "post_name": ["t3_1p6gyco", "t3_mira", "t3_zoey"],
    "post_author_name": ["DemiFiendRSA", "Aristodemus", "leonidas"],
    "post_author_fullname": ["t2_DemiFiendRSA", "t2_Aristodemus",
    "t2_leonidas"],
    "post_body": ["test1", "test2", "test3"],
    "post_created_at": [dt.datetime.strptime("2025-11-06 03:44:31",
    "%Y-%m-%d %H:%M:%S"), dt.datetime.now(), dt.datetime.now()],
    "post_edited_at": [dt.datetime.now(), dt.datetime.strptime("1970-
    01-01 08:00:00", "%Y-%m-%d %H:%M:%S"), dt.datetime.strptime("1970-01-
    01 08:00:00", "%Y-%m-%d %H:%M:%S")],
    "added_at": [dt.datetime.now(), dt.datetime.now()],
```

```

dt.datetime.now()]
})
test_posts_table

pyarrow.Table
post_title: string
post_score: int64
post_id: string
post_url: string
post_name: string
post_author_name: string
post_author_fullname: string
post_body: string
post_created_at: timestamp[us]
post_edited_at: timestamp[us]
added_at: timestamp[us]
----
post_title: [["new title","new title too","another new title"]]
post_score: [[9999,10000,24548]]
post_id: [["1p6gyco","mira","zoey"]]
post_url: [["https://i.redd.it/a0ecmeswtizf1.jpeg","https://some-
url.com","https://some-url.com"]]
post_name: [["t3_1p6gyco","t3_mira","t3_zoey"]]
post_author_name: [["DemiFiendRSA","Aristodemus","leonidas"]]
post_author_fullname:
[[["t2_DemiFiendRSA","t2_Aristodemus","t2_leonidas"]]]
post_body: [["test1","test2","test3"]]
post_created_at: [[2025-11-06 03:44:31.000000,2025-11-26
17:26:40.330449,2025-11-26 17:26:40.330449]]
post_edited_at: [[2025-11-26 17:26:40.330449,1970-01-01
08:00:00.000000,1970-01-01 08:00:00.000000]]
...
delta_posts_table.merge(
    test_posts_table,
    predicate="target.post_id = source.post_id",
    source_alias="source",
    target_alias="target",
    merge_schema=True
).when_matched_update(
    updates={
        # these are not included as these are the composite keys
        # that are not by good practice supposed to be updated
        # "post_id": "source.post_id",
        "post_title": "source.post_title",
        "post_score": "source.post_score",
        "post_url": "source.post_url",
        "post_name": "source.post_name",
        "post_author_name": "source.post_author_name",
        "post_author_fullname": "source.post_author_fullname",

```

```

        "post_body": "source.post_body",
        "post_created_at": "source.post_created_at",
        "post_edited_at": "source.post_edited_at",
        "added_at": "source.added_at",
    },
    # this tells delta to only update a record if the new record
    # does indeed have changed its column values when compared to
the
    # current record
    predicate="source.post_title IS DISTINCT FROM
target.post_title OR" \
        "source.post_score IS DISTINCT FROM target.post_score
OR" \
        "source.post_url IS DISTINCT FROM target.post_url OR" \
        "source.post_name IS DISTINCT FROM target.post_name OR" \
        "source.post_author_name IS DISTINCT FROM
target.post_author_name OR" \
        "source.post_author_fullname IS DISTINCT FROM
target.post_author_fullname" \
        "source.post_body IS DISTINCT FROM target.post_body OR" \
        "# "source.post_created_at IS DISTINCT FROM
target.post_created_at OR" \
        "source.post_edited_at > target.post_edited_at OR" \
        "source.added_at IS DISTINCT FROM target.added_at OR"
).when_not_matched_insert(
    updates={
        "post_id": "source.post_id",
        "post_title": "source.post_title",
        "post_score": "source.post_score",
        "post_url": "source.post_url",
        "post_name": "source.post_name",
        "post_author_name": "source.post_author_name",
        "post_author_fullname": "source.post_author_fullname",
        "post_body": "source.post_body",
        "post_created_at": "source.post_created_at",
        "post_edited_at": "source.post_edited_at",
        "added_at": "source.added_at",
    }
)\\
.execute()

```


Exception last)	Traceback (most recent call Cell In[23], line 51 1 delta_posts_table.merge(2 test_posts_table, 3 predicate="target.post_id = source.post_id", 4 source_alias="source",
--------------------	--

```

5      target_alias="target",
6      merge_schema=True
7  ) .when_matched_update(
8      updates={
9          # these are not included as these are the
composite keys
10         # that are not by good practice supposed to be
updated
11         # "post_id": "source.post_id",
12         "post_title": "source.post_title",
13         "post_score": "source.post_score",
14         "post_url": "source.post_url",
15         "post_name": "source.post_name",
16         "post_author_name": "source.post_author_name",
17         "post_author_fullname":
"source.post_author_fullname",
18         "post_body": "source.post_body",
19         "post_created_at": "source.post_created_at",
20         "post_edited_at": "source.post_edited_at",
21         "added_at": "source.added_at",
22     },
23     # this tells delta to only update a record if the new
record
24     # does indeed have changed its column values when
compared to the
25     # current record
26     predicate="source.post_title IS DISTINCT FROM
target.post_title OR" \
27         "source.post_score IS DISTINCT FROM
target.post_score OR" \
28         "source.post_url IS DISTINCT FROM target.post_url
OR" \
29         "source.post_name IS DISTINCT FROM
target.post_name OR" \
30         "source.post_author_name IS DISTINCT FROM
target.post_author_name OR" \
31         "source.post_author_fullname IS DISTINCT FROM
target.post_author_fullname" \
32         "source.post_body IS DISTINCT FROM
target.post_body OR" \
33         "# "source.post_created_at IS DISTINCT FROM
target.post_created_at OR" \
34         "source.post_edited_at > target.post_edited_at OR"
\
35         "source.added_at IS DISTINCT FROM target.added_at
OR"
36     ) .when_not_matched_insert(
37         updates={
38             "post_id": "source.post_id",

```

```

39         "post_title": "source.post_title",
40         "post_score": "source.post_score",
41         "post_url": "source.post_url",
42         "post_name": "source.post_name",
43         "post_author_name": "source.post_author_name",
44         "post_author_fullname": "source.post_author_fullname",
45         "post_body": "source.post_body",
46         "post_created_at": "source.post_created_at",
47         "post_edited_at": "source.post_edited_at",
48         "added_at": "source.added_at",
49     }
50 )
---> 51 .execute()

File c:\Users\LARRY\anaconda3\envs\tech-interview\Lib\site-packages\deltalake\table.py:1685, in TableMerger.execute(self)
1679 def execute(self) -> dict[str, Any]:
1680     """Executes `MERGE` with the previously provided settings
in Rust with Apache Datafusion query engine.
1681
1682     Returns:
1683         Dict: metrics
1684     """
-> 1685     metrics = self._table.merge_execute(self._builder)
1686     return json.loads(metrics)

```

Exception: External error: Schema error: Duplicate field name:
post_author_fullname

```
DELTA_COMMENTS_PATH = "../data/raw_reddit_posts_comments"
delta_comments_table = DeltaTable(DELTA_COMMENTS_PATH)
```

```
df = delta_comments_table.to_pandas()
```

```
df.head()
```

	post_title	post_score	post_id	post_url	post_name	post_author_name
0	None	NaN	loqywhk	None	t3_loqywhk	None
1	None	NaN	loqywhk	None	t3_loqywhk	None
2	None	NaN	loqywhk	None	t3_loqywhk	None
3	None	NaN	loqywhk	None	t3_loqywhk	None
4	None	NaN	loqywhk	None	t3_loqywhk	None

```

      level comment_id comment_name  comment_upvotes  comment_downvotes
\ 0  comment    nnmd63c   t1_nnmd63c           414            0
1  reply     nnnjit2   t1_nnnjit2           76            0
2  reply     nnmwj7m   t1_nnmwj7m           40            0
3  reply     nnmjoc7   t1_nnmjoc7           28            0
4  reply     nnn2wsq   t1_nnn2wsq           13            0

      comment_created_at  comment_edited_at  comment_author_name \
0  2025-11-08 00:38:48  1970-01-01 08:00:00          Wannabe_Pear
1  2025-11-08 01:09:51  1970-01-01 08:00:00          SkullMan140
2  2025-11-08 02:14:24  1970-01-01 08:00:00  Zimpiest_of_them_all
3  2025-11-08 01:10:37  1970-01-01 08:00:00        Enough-Ad-3111
4  2025-11-08 02:45:48  1970-01-01 08:00:00  Motor-Source8711

      comment_author_fullname comment_parent_id \
0             t2_1rcscd9n8f       t3_loqywhk
1             t2_10px8y        t1_nnmd63c
2             t2_1qnyyqmguk       t1_nnmd63c
3             t2_4r8syd06        t1_nnmd63c
4             t2_x4md3w9pn       t1_nnmd63c

      comment_body
0  The fact everyone thought this was gonna be a ...
1  Who would have thought the story of the golden...
2  I always love to imagine the faces of the peop...
3  Not to mention it has some potential for even ...
4  when you add in total views of covers, views o...

df.tail()

      post_title  post_score
post_id \
37          Huntrix x Powerpuff girls         494
lopikw3
38  EJAE and Mark Sonnenblick reveal an early iter...         154
lopqnxz
39  EJAE and Mark Sonnenblick reveal an early iter...         154
lopqnxz
40          Rumi and EJAE are so similar!         319
loplblfw
41          Rumi and EJAE are so similar!         319
loplblfw

      post_url  post_name
post_author_name \

```

```

37   https://i.redd.it/a0ecmeswtizf1.jpeg  t3_lopikw3
Necessary_Board_9775
38   https://v.redd.it/1txclqfcpkzf1  t3_lopqnxz
escarzador
39   https://v.redd.it/1txclqfcpkzf1  t3_lopqnxz
escarzador
40   https://www.reddit.com/gallery/1oplbfw  t3_loplbfw
Medical_Dimension919
41   https://www.reddit.com/gallery/1oplbfw  t3_loplbfw
Medical_Dimension919

      level comment_id comment_name  comment_upvotes
comment_downvotes \
37  comment    nnbzzgf  t1_nnbzzgf          30
0
38  comment    nnditt4  t1_nnditt4          32
0
39  reply     nndljsjn  t1_nndljsjn         12
0
40  comment    nncldfv  t1_nncldfv          39
0
41  reply     nncm6cz  t1_nncm6cz           7
0

      comment_created_at  comment_edited_at  comment_author_name \
37 2025-11-06 08:04:19 1970-01-01 08:00:00      Nerdy-Everyday
38 2025-11-06 14:03:06 1970-01-01 08:00:00      mid-lev
39 2025-11-06 14:29:00 1970-01-01 08:00:00      PinkJenni
40 2025-11-06 10:10:16 1970-01-01 08:00:00      Monkejedi
41 2025-11-06 10:15:05 1970-01-01 08:00:00 Medical_Dimension919

      comment_author_fullname comment_parent_id \
37            t2_buc5n1jm  t3_lopikw3
38            t2_lfznrwzse  t3_lopqnxz
39            t2_1zdx8rrbv  t1_nnditt4
40            t2_surt50njj  t3_loplbfw
41            t2_8f9ixwa4q  t1_nncldfv

      comment_body
37 Their personalities even line up.\n\nRumi / Bl...
38 That woman in the background looks like she's ...
39 She is taking it all in. EJAE's voice is so he...
40 Don't forget Remi Ami!\n\nhttps://preview.redd...
41 Of course! can't forget our queen!

```

<https://delta-docs-incubator.netlify.app/delta-update/#slowly-changing-data-scd-type-2-operation-into-delta-tables>

Upsert into a table using merge You can upsert data from a source table, view, or DataFrame into a target Delta table by using the MERGE SQL operation. Delta Lake supports inserts,

updates and deletes in MERGE, and it supports extended syntax beyond the SQL standards to facilitate advanced use cases.

Suppose you have a source table named people10mupdates or a source path at /tmp/delta/people-10m-updates that contains new data for a target table named people10m or a target path at /tmp/delta/people-10m. Some of these new records may already be present in the target data. To merge the new data, you want to update rows where the person's id is already present and insert the new rows where no matching id is present. You can run the following:

```
MERGE INTO people10m
USING people10mupdates
ON people10m.id = people10mupdates.id
WHEN MATCHED THEN
  UPDATE SET
    id = people10mupdates.id,
    firstName = people10mupdates.firstName,
    middleName = people10mupdates.middleName,
    lastName = people10mupdates.lastName,
    gender = people10mupdates.gender,
    birthDate = people10mupdates.birthDate,
    ssn = people10mupdates.ssn,
    salary = people10mupdates.salary
WHEN NOT MATCHED
  THEN INSERT (
    id,
    firstName,
    middleName,
    lastName,
    gender,
    birthDate,
    ssn,
    salary
  )
VALUES (
  people10mupdates.id,
  people10mupdates.firstName,
  people10mupdates.middleName,
  people10mupdates.lastName,
  people10mupdates.gender,
  people10mupdates.birthDate,
  people10mupdates.ssn,
  people10mupdates.salary
)
```

Lets test making a dummy updated record and upsert it into the existing delta lake table

```
created_at = dt.datetime.strptime("2025-11-06 08:04:19", "%Y-%m-%d %H:%M:%S")  
  
datetime.datetime(2025, 11, 6, 8, 4, 19)  
  
dt.datetime.now()  
  
datetime.datetime(2025, 11, 8, 9, 18, 15, 317213)  
  
test_table = pa.table({  
    "post_title": ["new title", "new title too", "another new title"],  
    "post_score": [9999, 10000, 24548],  
    "post_id": ["lopcev8", "mira", "zoey"],  
    "post_url": ["https://i.redd.it/a0ecmeswtizf1.jpeg",  
    "https://some-url.com", "https://some-url.com"],  
    "post_name": ["t3_lopcev8", "t3_mira", "t3_zoey"],  
    "post_author_name": ["DemiFiendRSA", "Aristodemus", "leonidas"],  
    "level": ["comment", "comment", "reply"],  
    "comment_upvotes": [1000, 2000, 3000],  
    "comment_downvotes": [6, 0, 0],  
    "comment_name": ["t1_nnakkjzc", "t1_someguy", "t1_anotherguy"],  
    "comment_id": ["nnakkjzc", "someguy", "anotherguy"],  
    "comment_created_at": [dt.datetime.strptime("2025-11-06 03:44:31",  
    "%Y-%m-%d %H:%M:%S"), dt.datetime.now(), dt.datetime.now()],  
    "comment_edited_at": [dt.datetime.now(),  
    dt.datetime.strptime("1970-01-01 08:00:00", "%Y-%m-%d %H:%M:%S"),  
    dt.datetime.strptime("1970-01-01 08:00:00", "%Y-%m-%d %H:%M:%S")],  
    "comment_author_name": ["Cluelessbigirl", "SomeGuy123",  
    "anotherguy123"],  
    "comment_author_fullname": ["t2_duy9z7kw", "t2_4234982398",  
    "t2_0feie23"],  
    "comment_parent_id": ["t3_lopcev8", "t3_mira", "t3_zoey"],  
    "comment_body": ["We're going up up up, it's our moment!", "fit  
    check for my napalm eraaa", "need to beat my face make it cute and  
    savage"],  
})  
test_table  
  
pyarrow.Table  
post_title: string  
post_score: int64  
post_id: string  
post_url: string  
post_name: string  
post_author_name: string  
level: string
```

```

comment_upvotes: int64
comment_downvotes: int64
comment_name: string
comment_id: string
comment_created_at: timestamp[us]
comment_edited_at: timestamp[us]
comment_author_name: string
comment_author_fullname: string
comment_parent_id: string
comment_body: string
---
post_title: [["new title","new title too","another new title"]]
post_score: [[9999,10000,24548]]
post_id: [["lopcev8","mira","zoey"]]
post_url: [["https://i.redd.it/a0ecmeswtizf1.jpeg","https://some-
url.com","https://some-url.com"]]
post_name: [["t3_lopcev8","t3_mira","t3_zoey"]]
post_author_name: [["DemiFiendRSA","Aristodemus","leonidas"]]
level: [["comment","comment","reply"]]
comment_upvotes: [[1000,2000,3000]]
comment_downvotes: [[6,0,0]]
comment_name: [["t1_nnakzjc","t1_someguy","t1_anotherguy"]]
...

```

now we merge the updated records and newly inserted records into the existing delta lake table based on the unique ids such as `post_name/post_id`, `comment_name/comment_id`, and `comment_parent_id` as these composite keys make the records unique and prevent duplication

what if a target delta lake table has a composite key that exists but has been deleted when scraped, I guess in this case as we don't need to already since we are doing this on a daily basis, and we only want to get the information left by users in a forum purely, doesn't matter if their accounts get deleted, we don't need to reflect this change in our data lake house table

which is basically the equivalent of

```

MERGE INTO delta_table
USING test_table
ON delta_table.post_id = source_table.post_id AND
target.comment_id = source.comment_id AND
target.comment_parent_id = source.comment_parent_id
WHEN MATCHED AND (
    -- The Conditional Update Predicate (Your Change Check)
    source.post_title IS DISTINCT FROM target.post_title OR
    source.post_score IS DISTINCT FROM target.post_score OR
    source.post_url IS DISTINCT FROM target.post_url OR
    source.post_name IS DISTINCT FROM target.post_name OR
    source.post_author_name IS DISTINCT FROM target.post_author_name
OR
    source.level IS DISTINCT FROM target.level OR
    source.comment_upvotes IS DISTINCT FROM target.comment_upvotes OR
    source.comment_downvotes IS DISTINCT FROM target.comment_downvotes
OR
    source.comment_name IS DISTINCT FROM target.comment_name OR
    source.comment_edited_at > target.comment_edited_at OR
    source.comment_author_name IS DISTINCT FROM
target.comment_author_name OR
    source.comment_author_fullname IS DISTINCT FROM
target.comment_author_fullname OR
    source.comment_body IS DISTINCT FROM target.comment_body
) THEN
    UPDATE SET
        target.post_title = source.post_title,
        target.post_score = source.post_score,
        target.post_url = source.post_url,
        target.post_name = source.post_name,
        target.post_author_name = source.post_author_name,
        target.level = source.level,
        target.comment_upvotes = source.comment_upvotes,
        target.comment_downvotes = source.comment_downvotes,
        target.comment_name = source.comment_name,
        target.comment_edited_at = source.comment_edited_at, -- Must
update the column that triggers the change!
        target.comment_author_name = source.comment_author_name,
        target.comment_author_fullname =
source.comment_author_fullname,
        target.comment_body = source.comment_body
WHEN NOT MATCHED THEN
    INSERT (
        post_id
        comment_id
        comment_parent_id
        post_title
        post_score
        post_url
        post_name

```

```

    post_author_name
    level
    comment_upvotes
    comment_downvotes
    comment_name
    comment_created_at
    comment_edited_at
    comment_author_name
    comment_author_fullname
    comment_body
)
VALUES (
    source.post_id
    source.comment_id
    source.comment_parent_id
    source.post_title
    source.post_score
    source.post_url
    source.post_name
    source.post_author_name
    source.level
    source.comment_upvotes
    source.comment_downvotes
    source.comment_name
    source.comment_created_at
    source.comment_edited_at
    source.comment_author_name
    source.comment_author_fullname
    source.comment_body
)
delta_table.merge(
    test_table,
    predicate="target.post_id = source.post_id AND \
        target.comment_id = source.comment_id AND \
        target.comment_parent_id = source.comment_parent_id",
    source_alias="source",
    target_alias="target"
).when_matched_update(
    updates={
        # these are not included as these are the composite keys
        # that are not by good practice supposed to be updated
        # "post_id": "source.post_id",
        # "comment_id": "source.comment_id",
        # "comment_parent_id": "source.comment_parent_id",
        "post_title": "source.post_title",
        "post_score": "source.post_score",
        "post_url": "source.post_url",
        "post_name": "source.post_name",
        "post_author_name": "source.post_author_name",

```

```

    "level": "source.level",
    "comment_upvotes": "source.comment_upvotes",
    "comment_downvotes": "source.comment_downvotes",
    "comment_name": "source.comment_name",
    # "comment_created_at": "source.comment_created_at",
    "comment_edited_at": "source.comment_edited_at",
    "comment_author_name": "source.comment_author_name",
    "comment_author_fullname": "source.comment_author_fullname",
    "comment_body": "source.comment_body"
},
# this tells delta to only update a record if the new record
# does indeed have changed its column values when compared to the
# current record
predicate="source.post_title IS DISTINCT FROM target.post_title
OR" \
    "source.post_score IS DISTINCT FROM target.post_score OR" \
    "source.post_url IS DISTINCT FROM target.post_url OR" \
    "source.post_name IS DISTINCT FROM target.post_name OR" \
    "source.post_author_name IS DISTINCT FROM
target.post_author_name OR" \
    "source.level IS DISTINCT FROM target.level OR" \
    "source.comment_upvotes IS DISTINCT FROM
target.comment_upvotes OR" \
    "source.comment_downvotes IS DISTINCT FROM
target.comment_downvotes OR" \
    "source.comment_name IS DISTINCT FROM target.comment_name
OR" \
    # "source.comment_created_at IS DISTINCT FROM
target.comment_created_at OR" \
    "source.comment_edited_at > target.comment_edited_at OR" \
    "source.comment_author_name IS DISTINCT FROM
target.comment_author_name OR" \
    "source.comment_author_fullname IS DISTINCT FROM
target.comment_author_fullname OR" \
    "source.comment_body IS DISTINCT FROM target.comment_body" \
).when_not_matched_insert_all()\.
.execute()

{'num_source_rows': 3,
 'num_target_rows_inserted': 1,
 'num_target_rows_updated': 0,
 'num_target_rows_deleted': 0,
 'num_target_rows_copied': 0,
 'num_output_rows': 1,
 'num_target_files_scanned': 1,
 'num_target_files_skipped_during_scan': 0,
 'num_target_files_added': 1,
 'num_target_files_removed': 0,
 'execution_time_ms': 82,
}

```

```
'scan_time_ms': 40,  
'rewrite_time_ms': 0}
```

after update and insertion of these 2 records we will check the delta table again

```
new_df = delta_table.to_pandas()  
new_df.head()
```

		post_title	post_score
post_id	\		
0		another new title	24548
zoey			
1		new title too	10000
mira			
2		new title	9999
lopcev8			
3	'KPop Demon Hunters 2' Aims for 2029 Release o...		2121
lopcev8			
4	'KPop Demon Hunters 2' Aims for 2029 Release o...		2121
lopcev8			

		post_url	post_name	\
0		https://some-url.com	t3_zoey	
1		https://some-url.com	t3_mira	
2		https://i.redd.it/a0ecmeswtizf1.jpeg	t3_lopcev8	
3		https://variety.com/2025/film/news/kpop-demon-...	t3_lopcev8	
4		https://variety.com/2025/film/news/kpop-demon-...	t3_lopcev8	

	post_author_name	level	comment_id	comment_name
comment_upvotes	\			
0	leonidas	reply	anotherguy	t1_anotherguy
3000				
1	Aristodemus	comment	someguy	t1_someguy
2000				
2	DemiFiendRSA	comment	nnakzjc	t1_nnakzjc
1000				
3	DemiFiendRSA	reply	nnaold6	t1_nnaold6
205				
4	DemiFiendRSA	reply	nnb3wf6	t1_nnb3wf6
79				

	comment_downvotes	comment_created_at
comment_edited_at	\	
0	0	2025-11-08 13:13:10.800989 1970-01-01
08:00:00.000000		
1	0	2025-11-08 09:29:41.414516 1970-01-01

```
08:00:00.000000
2                               6 2025-11-06 03:44:31.000000 2025-11-08
09:29:41.414516
3                               0 2025-11-06 04:01:42.000000 1970-01-01
08:00:00.000000
4                               0 2025-11-06 05:15:41.000000 1970-01-01
08:00:00.000000

comment_author_name comment_author_fullname comment_parent_id \
0      anotherguy123           t2_0feie23      t3_zoey
1      SomeGuy123            t2_4234982398   t3_mira
2      Cluelessbigirl        t2_duy9z7kw     t3_1opcev8
3      Hero_of_the_toons     t2_64w7o0ao     t1_nnakzjc
4      CelineShotFirst       t2_1rqf9pzis9    t1_nnakzjc

comment_body
0      need to beat my face make it cute and savage
1          fit check for my napalm eraaa
2      We're going up up up, it's our moment!
3  Wouldn't want a Kpdh equivalent of Sonic 06 if...
4  The original film took 7 years from pitch to r...
```