WITH

vars AS (

SELECT

TIMESTAMP("{START\_TIME}") AS start\_date,

TIMESTAMP("{END\_TIME}") AS end\_date,

),

-- Get raw user-tweet interaction events from UUA (We will use fav engagements here)

raw\_engagements AS (

SELECT

userIdentifier.userId AS userId,

eventMetadata.sourceTimestampMs AS tsMillis,

CASE

WHEN actionType IN ({CONTRIBUTING\_ACTION\_TYPES\_STR}) THEN {CONTRIBUTING\_ACTION\_TWEET\_ID\_COLUMN}

WHEN actionType IN ({UNDO\_ACTION\_TYPES\_STR}) THEN {UNDO\_ACTION\_TWEET\_ID\_COLUMN}

END AS tweetId,

CASE

WHEN actionType IN ({CONTRIBUTING\_ACTION\_TYPES\_STR}) THEN 1

WHEN actionType IN ({UNDO\_ACTION\_TYPES\_STR}) THEN -1

END AS doOrUndo

FROM `twttr-bql-unified-prod.unified\_user\_actions\_engagements.streaming\_unified\_user\_actions\_engagements`, vars

WHERE (DATE(dateHour) >= DATE(vars.start\_date) AND DATE(dateHour) <= DATE(vars.end\_date))

AND eventMetadata.sourceTimestampMs >= UNIX\_MILLIS(vars.start\_date)

AND eventMetadata.sourceTimestampMs <= UNIX\_MILLIS(vars.end\_date)

AND (actionType IN ({CONTRIBUTING\_ACTION\_TYPES\_STR})

OR actionType IN ({UNDO\_ACTION\_TYPES\_STR}))

),

-- Get video tweet ids

video\_tweet\_ids AS (

WITH vars AS (

SELECT

TIMESTAMP("{START\_TIME}") AS start\_date,

TIMESTAMP("{END\_TIME}") AS end\_date

),

-- Get raw user-tweet interaction events from UUA

video\_view\_engagements AS (

SELECT item.tweetInfo.actionTweetId AS tweetId

FROM `twttr-bql-unified-prod.unified\_user\_actions\_engagements.streaming\_unified\_user\_actions\_engagements`, vars

WHERE (DATE(dateHour) >= DATE(vars.start\_date) AND DATE(dateHour) <= DATE(vars.end\_date))

AND eventMetadata.sourceTimestampMs >= UNIX\_MILLIS(start\_date)

AND eventMetadata.sourceTimestampMs <= UNIX\_MILLIS(end\_date)

AND (actionType IN ("ClientTweetVideoPlayback50")

OR actionType IN ("ClientTweetVideoPlayback95"))

)

SELECT DISTINCT(tweetId)

FROM video\_view\_engagements

),

-- Join video tweet ids

video\_tweets\_engagements AS (

SELECT raw\_engagements.\*

FROM raw\_engagements JOIN video\_tweet\_ids USING(tweetId)

),

-- Group by userId and tweetId

user\_tweet\_engagement\_pairs AS (

SELECT userId, tweetId, ARRAY\_AGG(STRUCT(doOrUndo, tsMillis) ORDER BY tsMillis DESC LIMIT 1) AS details, COUNT(\*) AS cnt

FROM video\_tweets\_engagements

GROUP BY userId, tweetId

)

-- Remove undo events

SELECT userId, tweetId, CAST(dt.tsMillis AS FLOAT64) AS tsMillis

FROM user\_tweet\_engagement\_pairs, vars

CROSS JOIN UNNEST(details) AS dt

WHERE dt.doOrUndo = 1