package com.twitter.simclusters\_v2.scio.bq\_generation.simclusters\_index\_generation

object Config {

// Common Root Path

val RootMHPath: String = "manhattan\_sequence\_files/simclusters\_to\_tweet\_index/"

val RootThriftPath: String = "processed/simclusters\_to\_tweet\_index/"

val AdhocRootPath = "adhoc/simclusters\_to\_tweet\_index/"

// cluster-to-tweet KeyVal Dataset Output Path

val FavBasedClusterToTweetIndexOutputPath = "fav\_based\_index"

val FavBasedEvergreenContentClusterToTweetIndexOutputPath = "fav\_based\_evergreen\_index"

val FavBasedVideoClusterToTweetIndexOutputPath = "fav\_based\_video\_index"

val VideoViewBasedClusterToTweetIndexOutputPath = "video\_view\_based\_index"

val RetweetBasedClusterToTweetIndexOutputPath = "retweet\_based\_index"

val ReplyBasedClusterToTweetIndexOutputPath = "reply\_based\_index"

val PushOpenBasedClusterToTweetIndexOutputPath = "push\_open\_based\_index"

val AdsFavBasedClusterToTweetIndexOutputPath = "ads\_fav\_based\_index"

val AdsFavClickBasedClusterToTweetIndexOutputPath = "ads\_fav\_click\_based\_index"

// SQL file path

val simclustersEngagementBasedIndexGenerationSQLPath =

s"/com/twitter/simclusters\_v2/scio/bq\_generation/sql/engagement\_based\_index\_generation.sql"

val unifiedUserTweetActionPairGenerationSQLPath =

s"/com/twitter/simclusters\_v2/scio/bq\_generation/sql/unified\_user\_tweet\_action\_pair\_generation.sql"

val combinedUserTweetActionPairGenerationSQLPath =

s"/com/twitter/simclusters\_v2/scio/bq\_generation/sql/combined\_user\_tweet\_action\_pair\_generation.sql"

val adsUserTweetActionPairGenerationSQLPath =

s"/com/twitter/simclusters\_v2/scio/bq\_generation/sql/ads\_user\_tweet\_action\_pair\_generation.sql"

val evergreenContentUserTweetActionPairGenerationSQLPath =

s"/com/twitter/simclusters\_v2/scio/bq\_generation/sql/evergreen\_content\_user\_tweet\_action\_pair\_generation.sql"

val favBasedVideoTweetActionPairGenerationSQLPath =

s"/com/twitter/simclusters\_v2/scio/bq\_generation/sql/user\_video\_tweet\_fav\_engagement\_generation.sql"

// Table name for server/client engagements

val clientEngagementTableName: String = "twttr-bq-iesource-prod.user.client\_engagements"

val serverEngagementTableName: String = "twttr-bq-iesource-prod.user.server\_engagements"

// Tweet id column names from UUA

val actionTweetIdColumn: String = "item.tweetInfo.actionTweetId"

val retweetTweetIdColumn: String = "item.tweetInfo.retweetedTweetId"

val replyTweetIdColumn: String = "item.tweetInfo.inReplyToTweetId"

val pushTweetIdColumn: String = "item.notificationInfo.content.tweetNotification.tweetId"

// Do not enable health or video filters by default

val enableHealthAndVideoFilters: Boolean = false

// Do not enable top k tweets per cluster intersection with fav-based clusters

val enableIntersectionWithFavBasedClusterTopKTweetsIndex: Boolean = false

// Min fav/interaction threshold

val minInteractionCount: Int = 50

val minFavCount: Int = 50

// Tweet Embeddings configs

val tweetEmbeddingsLength: Int = 50

val tweetEmbeddingsHalfLife: Int = 28800000

// Cluster-to-tweet index configs

val clusterTopKTweets: Int = 2000

val maxTweetAgeHours: Int = 24

val minEngagementPerCluster: Int = 0

// Placeholder action type for interactions that don't have undo events (e.g. video views)

val PlaceholderActionType: String = "PLACEHOLDER\_ACTION\_TYPE"

// Ads event engagement type ids

val AdsFavEngagementTypeIds = Seq(8) // Fav promoted tweet

val AdsClickEngagementTypeIds = Seq(

1, //URL

42, // CARD\_URL\_CLICK

53, // WEBSITE\_CARD\_CONTAINER\_CLICK

54, // WEBSITE\_CARD\_BUTTON\_CLICK

55, // WEBSITE\_CARD\_IMAGE\_CLICK

56, // WEBSITE\_CARD\_TITLE\_CLICK

69, // BUYNOW\_CARD\_CLICK

70, // BUYNOW\_PURCHASE\_SUCCESS

72, // VIDEO\_CTA\_URL\_CLICK

76, // VIDEO\_AD\_CTA\_URL\_CLICK

80, // VIDEO\_CONTENT\_CTA\_URL\_CLICK

84, // CL\_OFFER\_CARD\_CLICK

)

}