namespace java com.twitter.simclusters\_v2.thriftjava

namespace py gen.twitter.simclusters\_v2.graph

#@namespace scala com.twitter.simclusters\_v2.thriftscala

#@namespace strato com.twitter.simclusters\_v2

struct DecayedSums {

// last time the decayed sum was updated, in millis.

1: required i64 lastUpdatedTimestamp

// a map from half life (specified in days) to the decayed sum

2: required map<i32, double> halfLifeInDaysToDecayedSums

}(persisted = 'true', hasPersonalData = 'false')

struct EdgeWithDecayedWeights {

1: required i64 sourceId(personalDataType = 'UserId')

2: required i64 destinationId(personalDataType = 'UserId')

3: required DecayedSums weights

}(persisted="true", hasPersonalData = "true")

struct NeighborWithWeights {

1: required i64 neighborId(personalDataType = 'UserId')

2: optional bool isFollowed(personalDataType = 'Follow')

3: optional double followScoreNormalizedByNeighborFollowersL2(personalDataType = 'EngagementsPublic')

4: optional double favScoreHalfLife100Days(personalDataType = 'EngagementsPublic')

5: optional double favScoreHalfLife100DaysNormalizedByNeighborFaversL2(personalDataType = 'EngagementsPublic')

// log(favScoreHalfLife100Days + 1)

6: optional double logFavScore(personalDataType = 'EngagementsPublic')

// log(favScoreHalfLife100Days + 1) normalized so that a user's incoming weights have unit l2 norm

7: optional double logFavScoreL2Normalized(personalDataType = 'EngagementsPublic')

}(persisted = 'true', hasPersonalData = 'true')

struct UserAndNeighbors {

1: required i64 userId(personalDataType = 'UserId')

2: required list<NeighborWithWeights> neighbors

}(persisted="true", hasPersonalData = 'true')

struct NormsAndCounts {

1: required i64 userId(personalDataType = 'UserId')

2: optional double followerL2Norm(personalDataType = 'CountOfFollowersAndFollowees')

3: optional double faverL2Norm(personalDataType = 'EngagementsPublic')

4: optional i64 followerCount(personalDataType = 'CountOfFollowersAndFollowees')

5: optional i64 faverCount(personalDataType = 'EngagementsPublic')

// sum of the weights on the incoming edges where someone fav'ed this producer

6: optional double favWeightsOnFavEdgesSum(personalDataType = 'EngagementsPublic')

// sum of the fav weights on all the followers of this producer

7: optional double favWeightsOnFollowEdgesSum(personalDataType = 'EngagementsPublic')

// log(favScore + 1)

8: optional double logFavL2Norm(personalDataType = 'EngagementsPublic')

// sum of log(favScore + 1) on the incoming edges where someone fav'ed this producer

9: optional double logFavWeightsOnFavEdgesSum(personalDataType = 'EngagementsPublic')

// sum of log(favScore + 1) on all the followers of this producer

10: optional double logFavWeightsOnFollowEdgesSum(personalDataType = 'EngagementsPublic')

}(persisted="true", hasPersonalData = 'true')