package com.twitter.search.common.schema.earlybird;

import java.util.Collection;

import java.util.EnumSet;

import java.util.List;

import java.util.Map;

import java.util.Set;

import javax.annotation.Nullable;

import com.google.common.annotations.VisibleForTesting;

import com.google.common.collect.ImmutableList;

import com.google.common.collect.ImmutableMap;

import com.google.common.collect.ImmutableSet;

import com.google.common.collect.Sets;

import com.twitter.search.common.indexing.thriftjava.ThriftGeoLocationSource;

import com.twitter.search.common.schema.ImmutableSchema;

import com.twitter.search.common.schema.SchemaBuilder;

import com.twitter.search.common.schema.base.FeatureConfiguration;

import com.twitter.search.common.schema.base.FieldNameToIdMapping;

import com.twitter.search.common.schema.thriftjava.ThriftFeatureNormalizationType;

/\*\*

\* Field names, field IDs etc.

\*/

public class EarlybirdFieldConstants extends FieldNameToIdMapping {

@VisibleForTesting

public static final String ENCODED\_TWEET\_FEATURES\_FIELD\_NAME = "encoded\_tweet\_features";

@VisibleForTesting

public static final String EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME =

"extended\_encoded\_tweet\_features";

private enum FlagFeatureFieldType {

NON\_FLAG\_FEATURE\_FIELD,

FLAG\_FEATURE\_FIELD

}

private enum UnusedFeatureFieldType {

USED\_FEATURE\_FIELD,

UNUSED\_FEATURE\_FIELD

}

/\*\*

\* CSF\_NAME\_TO\_MIN\_ENGAGEMENT\_FIELD\_MAP and MIN\_ENGAGEMENT\_FIELD\_TO\_CSF\_NAME\_MAP are used in

\* EarlybirdLuceneQueryVisitor to map the CSFs REPLY\_COUNT, RETWEET\_COUNT, and FAVORITE\_COUNT to

\* their respective min engagement fields, and vice versa.

\*/

public static final ImmutableMap<String, EarlybirdFieldConstant>

CSF\_NAME\_TO\_MIN\_ENGAGEMENT\_FIELD\_MAP = ImmutableMap.<String, EarlybirdFieldConstant>builder()

.put(EarlybirdFieldConstant.REPLY\_COUNT.getFieldName(),

EarlybirdFieldConstant.NORMALIZED\_REPLY\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD)

.put(EarlybirdFieldConstant.RETWEET\_COUNT.getFieldName(),

EarlybirdFieldConstant.NORMALIZED\_RETWEET\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD)

.put(EarlybirdFieldConstant.FAVORITE\_COUNT.getFieldName(),

EarlybirdFieldConstant.NORMALIZED\_FAVORITE\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD)

.build();

public static final ImmutableMap<String, EarlybirdFieldConstant>

MIN\_ENGAGEMENT\_FIELD\_TO\_CSF\_NAME\_MAP = ImmutableMap.<String, EarlybirdFieldConstant>builder()

.put(EarlybirdFieldConstant.NORMALIZED\_REPLY\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD

.getFieldName(),

EarlybirdFieldConstant.REPLY\_COUNT)

.put(EarlybirdFieldConstant.NORMALIZED\_RETWEET\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD

.getFieldName(),

EarlybirdFieldConstant.RETWEET\_COUNT)

.put(EarlybirdFieldConstant.NORMALIZED\_FAVORITE\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD

.getFieldName(),

EarlybirdFieldConstant.FAVORITE\_COUNT)

.build();

/\*\*

\* A list of Earlybird field names and field IDs, and the clusters that need them.

\*/

public enum EarlybirdFieldConstant {

// These enums are grouped by category and sorted alphabetically.

// Next indexed field ID is 76

// Next CSF field ID is 115

// Next encoded\_features CSF field ID is 185

// Next extended\_encoded\_features CSF field ID is 284

// Text searchable fields

// Provides slow ID Mapping from tweet ID to doc ID through TermsEnum.seekExact().

ID\_FIELD("id", 0, EarlybirdCluster.ALL\_CLUSTERS),

RESOLVED\_LINKS\_TEXT\_FIELD("resolved\_links\_text", 1),

TEXT\_FIELD("text", 2),

TOKENIZED\_FROM\_USER\_FIELD("tokenized\_from\_user", 3),

// Other indexed fields

CARD\_TITLE\_FIELD("card\_title", 4),

CARD\_DESCRIPTION\_FIELD("card\_description", 5),

// We require the createdAt field to be set so we can properly filter tweets based on time.

CREATED\_AT\_FIELD("created\_at", 6, EarlybirdCluster.ALL\_CLUSTERS),

// 7 was formerly EVENT\_IDS\_FIELD("event\_ids", 7, EarlybirdCluster.REALTIME)

ENTITY\_ID\_FIELD("entity\_id", 40),

// The screen name of the user that created the tweet. Should be set to the normalized value in

// the com.twitter.gizmoduck.thriftjava.Profile.screen\_name field.

FROM\_USER\_FIELD("from\_user", 8),

// The numeric ID of the user that created the tweet.

FROM\_USER\_ID\_FIELD("from\_user\_id", 9, EarlybirdCluster.ALL\_CLUSTERS),

CARD\_DOMAIN\_FIELD("card\_domain", 11),

CARD\_NAME\_FIELD("card\_name", 12),

GEO\_HASH\_FIELD("geo\_hash", 13),

HASHTAGS\_FIELD("hashtags", 14),

HF\_PHRASE\_PAIRS\_FIELD(ImmutableSchema.HF\_PHRASE\_PAIRS\_FIELD, 15),

HF\_TERM\_PAIRS\_FIELD(ImmutableSchema.HF\_TERM\_PAIRS\_FIELD, 16),

IMAGE\_LINKS\_FIELD("image\_links", 17),

IN\_REPLY\_TO\_TWEET\_ID\_FIELD("in\_reply\_to\_tweet\_id", 59),

IN\_REPLY\_TO\_USER\_ID\_FIELD("in\_reply\_to\_user\_id", 38),

// The internal field is used for many purposes:

// 1. to store facet skiplists

// 2. to power the filter operator, by storing posting list for terms like \_\_filter\_twimg

// 3. to store posting lists for positive and negative smileys

// 4. to store geo location types.

// etc.

INTERNAL\_FIELD("internal", 18, EarlybirdCluster.ALL\_CLUSTERS),

ISO\_LANGUAGE\_FIELD("iso\_lang", 19),

LINK\_CATEGORY\_FIELD("link\_category", 36),

LINKS\_FIELD("links", 21),

MENTIONS\_FIELD("mentions", 22),

// Field 23 used to be NAMED\_ENTITIES\_FIELD

NEWS\_LINKS\_FIELD("news\_links", 24),

NORMALIZED\_SOURCE\_FIELD("norm\_source", 25),

PLACE\_FIELD("place", 26),

// Field 37 used to be PUBLICLY\_INFERRED\_USER\_LOCATION\_PLACE\_ID\_FIELD

// The ID of the source tweet. Set for retweets only.

RETWEET\_SOURCE\_TWEET\_ID\_FIELD("retweet\_source\_tweet\_id", 60,

EarlybirdCluster.ALL\_CLUSTERS),

// The ID of the source tweet's author. Set for retweets only.

RETWEET\_SOURCE\_USER\_ID\_FIELD("retweet\_source\_user\_id", 39),

SOURCE\_FIELD("source", 29),

STOCKS\_FIELD("stocks", 30),

// The screen name of the user that a tweet was directed at.

TO\_USER\_FIELD("to\_user", 32),

// Field 33 used to be TOPIC\_IDS\_FIELD and is now unused. It can be reused later.

TWIMG\_LINKS\_FIELD("twimg\_links", 34),

VIDEO\_LINKS\_FIELD("video\_links", 35),

CAMELCASE\_USER\_HANDLE\_FIELD("camelcase\_tokenized\_from\_user", 41),

// This field should be set to the the tokenized and normalized value in the

// com.twitter.gizmoduck.thriftjava.Profile.name field.

TOKENIZED\_USER\_NAME\_FIELD("tokenized\_from\_user\_display\_name", 42),

CONVERSATION\_ID\_FIELD("conversation\_id", 43),

PLACE\_ID\_FIELD("place\_id", 44),

PLACE\_FULL\_NAME\_FIELD("place\_full\_name", 45),

PLACE\_COUNTRY\_CODE\_FIELD("place\_country\_code", 46),

PROFILE\_GEO\_COUNTRY\_CODE\_FIELD("profile\_geo\_country\_code", 47),

PROFILE\_GEO\_REGION\_FIELD("profile\_geo\_region", 48),

PROFILE\_GEO\_LOCALITY\_FIELD("profile\_geo\_locality", 49),

LIKED\_BY\_USER\_ID\_FIELD("liked\_by\_user\_id", 50, EarlybirdCluster.REALTIME),

NORMALIZED\_REPLY\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD(

"normalized\_reply\_count\_greater\_than\_or\_equal\_to", 51, EarlybirdCluster.FULL\_ARCHIVE),

NORMALIZED\_RETWEET\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD(

"normalized\_retweet\_count\_greater\_than\_or\_equal\_to", 52, EarlybirdCluster.FULL\_ARCHIVE),

NORMALIZED\_FAVORITE\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD(

"normalized\_favorite\_count\_greater\_than\_or\_equal\_to", 53, EarlybirdCluster.FULL\_ARCHIVE),

COMPOSER\_SOURCE("composer\_source", 54),

QUOTED\_TWEET\_ID\_FIELD("quoted\_tweet\_id", 55),

QUOTED\_USER\_ID\_FIELD("quoted\_user\_id", 56),

RETWEETED\_BY\_USER\_ID("retweeted\_by\_user\_id", 57, EarlybirdCluster.REALTIME),

REPLIED\_TO\_BY\_USER\_ID("replied\_to\_by\_user\_id", 58, EarlybirdCluster.REALTIME),

CARD\_LANG("card\_lang", 61),

// SEARCH-27823: Field ID 62 used to be named\_entity, which was the combination of all

// named\_entity\* fields below. We need to leave 62 unused for backwards compatibility.

NAMED\_ENTITY\_FROM\_URL\_FIELD("named\_entity\_from\_url", 63),

NAMED\_ENTITY\_FROM\_TEXT\_FIELD("named\_entity\_from\_text", 64),

NAMED\_ENTITY\_WITH\_TYPE\_FROM\_URL\_FIELD("named\_entity\_with\_type\_from\_url", 65),

NAMED\_ENTITY\_WITH\_TYPE\_FROM\_TEXT\_FIELD("named\_entity\_with\_type\_from\_text", 66),

DIRECTED\_AT\_USER\_ID\_FIELD("directed\_at\_user\_id", 67),

SPACE\_ID\_FIELD("space\_id", 68,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_GENERAL\_PURPOSE\_CLUSTERS),

SPACE\_TITLE\_FIELD("space\_title", 69,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_GENERAL\_PURPOSE\_CLUSTERS),

// Detailed description of the space admin fields can be found at go/earlybirdfields.

SPACE\_ADMIN\_FIELD("space\_admin", 70,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_GENERAL\_PURPOSE\_CLUSTERS),

TOKENIZED\_SPACE\_ADMIN\_FIELD("tokenized\_space\_admin", 71,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_GENERAL\_PURPOSE\_CLUSTERS),

CAMELCASE\_TOKENIZED\_SPACE\_ADMIN\_FIELD("camelcase\_tokenized\_space\_admin", 72,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_GENERAL\_PURPOSE\_CLUSTERS),

TOKENIZED\_SPACE\_ADMIN\_DISPLAY\_NAME\_FIELD("tokenized\_space\_admin\_display\_name", 73,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_GENERAL\_PURPOSE\_CLUSTERS),

URL\_DESCRIPTION\_FIELD("url\_description", 74),

URL\_TITLE\_FIELD("url\_title", 75),

// CSF

CARD\_TYPE\_CSF\_FIELD("card\_type\_csf", 100),

ENCODED\_TWEET\_FEATURES\_FIELD(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, 102,

EarlybirdCluster.ALL\_CLUSTERS),

// Provides the doc ID -> original tweet ID mapping for retweets.

SHARED\_STATUS\_ID\_CSF("shared\_status\_id\_csf", 106, EarlybirdCluster.ALL\_CLUSTERS),

// Provides the doc ID -> tweet author's user ID mapping.

FROM\_USER\_ID\_CSF("from\_user\_id\_csf", 103, EarlybirdCluster.ALL\_CLUSTERS),

CREATED\_AT\_CSF\_FIELD("created\_at\_csf", 101, EarlybirdCluster.ARCHIVE\_CLUSTERS),

// Provides the doc ID -> tweet ID mapping.

ID\_CSF\_FIELD("id\_csf", 104, EarlybirdCluster.ARCHIVE\_CLUSTERS),

LAT\_LON\_CSF\_FIELD("latlon\_csf", 105),

CONVERSATION\_ID\_CSF("conversation\_id\_csf", 107, EarlybirdCluster.ALL\_CLUSTERS),

QUOTED\_TWEET\_ID\_CSF("quoted\_tweet\_id\_csf", 108),

QUOTED\_USER\_ID\_CSF("quoted\_user\_id\_csf", 109),

CARD\_LANG\_CSF("card\_lang\_csf", 110),

DIRECTED\_AT\_USER\_ID\_CSF("directed\_at\_user\_id\_csf", 111),

REFERENCE\_AUTHOR\_ID\_CSF("reference\_author\_id\_csf", 112),

EXCLUSIVE\_CONVERSATION\_AUTHOR\_ID\_CSF("exclusive\_conversation\_author\_id\_csf", 113),

CARD\_URI\_CSF("card\_uri\_csf", 114),

// CSF Views on top of ENCODED\_TWEET\_FEATURES\_FIELD

IS\_RETWEET\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_RETWEET\_FLAG", 150,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

IS\_OFFENSIVE\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_OFFENSIVE\_FLAG", 151,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_LINK\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_LINK\_FLAG", 152,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_TREND\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_TREND\_FLAG", 153,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

IS\_REPLY\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_REPLY\_FLAG", 154,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

IS\_SENSITIVE\_CONTENT(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_SENSITIVE\_CONTENT", 155,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_MULTIPLE\_HASHTAGS\_OR\_TRENDS\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"HAS\_MULTIPLE\_HASHTAGS\_OR\_TRENDS\_FLAG", 156, FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.ALL\_CLUSTERS),

FROM\_VERIFIED\_ACCOUNT\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "FROM\_VERIFIED\_ACCOUNT\_FLAG",

157,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

TEXT\_SCORE(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "TEXT\_SCORE", 158,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

LANGUAGE(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "LANGUAGE", 159,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

LINK\_LANGUAGE(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "LINK\_LANGUAGE", 160,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_IMAGE\_URL\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_IMAGE\_URL\_FLAG", 161,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_VIDEO\_URL\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_VIDEO\_URL\_FLAG", 162,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_NEWS\_URL\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_NEWS\_URL\_FLAG", 163,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_EXPANDO\_CARD\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_EXPANDO\_CARD\_FLAG", 164,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_MULTIPLE\_MEDIA\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_MULTIPLE\_MEDIA\_FLAG", 165,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

PROFILE\_IS\_EGG\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "PROFILE\_IS\_EGG\_FLAG", 166,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

NUM\_MENTIONS(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "NUM\_MENTIONS", 167,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

NUM\_HASHTAGS(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "NUM\_HASHTAGS", 168,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_CARD\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_CARD\_FLAG", 169,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_VISIBLE\_LINK\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_VISIBLE\_LINK\_FLAG", 170,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

USER\_REPUTATION(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "USER\_REPUTATION", 171,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

IS\_USER\_SPAM\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_USER\_SPAM\_FLAG", 172,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

IS\_USER\_NSFW\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_USER\_NSFW\_FLAG", 173,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

IS\_USER\_BOT\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_USER\_BOT\_FLAG", 174,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

IS\_USER\_NEW\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_USER\_NEW\_FLAG", 175,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

PREV\_USER\_TWEET\_ENGAGEMENT(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "PREV\_USER\_TWEET\_ENGAGEMENT",

176,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

COMPOSER\_SOURCE\_IS\_CAMERA\_FLAG(

ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"COMPOSER\_SOURCE\_IS\_CAMERA\_FLAG",

177,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.ALL\_CLUSTERS),

RETWEET\_COUNT(

ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"RETWEET\_COUNT",

178,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.ALL\_CLUSTERS,

ThriftFeatureNormalizationType.LEGACY\_BYTE\_NORMALIZER\_WITH\_LOG2),

FAVORITE\_COUNT(

ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"FAVORITE\_COUNT",

179,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.ALL\_CLUSTERS,

ThriftFeatureNormalizationType.LEGACY\_BYTE\_NORMALIZER\_WITH\_LOG2),

REPLY\_COUNT(

ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"REPLY\_COUNT",

180,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.ALL\_CLUSTERS,

ThriftFeatureNormalizationType.LEGACY\_BYTE\_NORMALIZER\_WITH\_LOG2),

PARUS\_SCORE(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "PARUS\_SCORE", 181,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

/\*\*

\* This is the rough percentage of the nth token at 140 divided by num tokens

\* and is basically n / num tokens where n is the token starting before 140 characters

\*/

VISIBLE\_TOKEN\_RATIO(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "VISIBLE\_TOKEN\_RATIO", 182,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_QUOTE\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_QUOTE\_FLAG", 183,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

FROM\_BLUE\_VERIFIED\_ACCOUNT\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"FROM\_BLUE\_VERIFIED\_ACCOUNT\_FLAG",

184,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

TWEET\_SIGNATURE(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "TWEET\_SIGNATURE", 188,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

// MEDIA TYPES

HAS\_CONSUMER\_VIDEO\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_CONSUMER\_VIDEO\_FLAG", 189,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_PRO\_VIDEO\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_PRO\_VIDEO\_FLAG", 190,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_VINE\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_VINE\_FLAG", 191,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_PERISCOPE\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_PERISCOPE\_FLAG", 192,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

HAS\_NATIVE\_IMAGE\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "HAS\_NATIVE\_IMAGE\_FLAG", 193,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

// NOTE: if possible, please reserve field ID 194 to 196 for future media types (SEARCH-9131)

IS\_NULLCAST\_FLAG(ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, "IS\_NULLCAST\_FLAG", 197,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD, EarlybirdCluster.ALL\_CLUSTERS),

// EXTENDED ENCODED TWEET FEATURES that's not available on archive clusters

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME, 200,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

EMBEDS\_IMPRESSION\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EMBEDS\_IMPRESSION\_COUNT",

221,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.LEGACY\_BYTE\_NORMALIZER),

EMBEDS\_URL\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EMBEDS\_URL\_COUNT",

222,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.LEGACY\_BYTE\_NORMALIZER),

VIDEO\_VIEW\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"VIDEO\_VIEW\_COUNT",

223,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.LEGACY\_BYTE\_NORMALIZER),

// empty bits in integer 0 (starting bit 24, 8 bits)

EXTENDED\_FEATURE\_UNUSED\_BITS\_0\_24\_8(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS\_0\_24\_8", 244,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

// SEARCH-8564 - Reference Tweet Author ID

REFERENCE\_AUTHOR\_ID\_LEAST\_SIGNIFICANT\_INT(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"REFERENCE\_AUTHOR\_ID\_LEAST\_SIGNIFICANT\_INT", 202,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

REFERENCE\_AUTHOR\_ID\_MOST\_SIGNIFICANT\_INT(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"REFERENCE\_AUTHOR\_ID\_MOST\_SIGNIFICANT\_INT", 203,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

// SEARCHQUAL-8130: engagement counters v2

// Integer 3

RETWEET\_COUNT\_V2(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"RETWEET\_COUNT\_V2", 225,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

FAVORITE\_COUNT\_V2(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"FAVORITE\_COUNT\_V2", 226,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

REPLY\_COUNT\_V2(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"REPLY\_COUNT\_V2", 227,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

EMBEDS\_IMPRESSION\_COUNT\_V2(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EMBEDS\_IMPRESSION\_COUNT\_V2",

228,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

// Integer 4

EMBEDS\_URL\_COUNT\_V2(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EMBEDS\_URL\_COUNT\_V2",

229,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

VIDEO\_VIEW\_COUNT\_V2(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"VIDEO\_VIEW\_COUNT\_V2",

230,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

QUOTE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"QUOTE\_COUNT",

231,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

// Tweet Safety Labels

LABEL\_ABUSIVE\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LABEL\_ABUSIVE\_FLAG", 232,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

LABEL\_ABUSIVE\_HI\_RCL\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LABEL\_ABUSIVE\_HI\_RCL\_FLAG", 233,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

LABEL\_DUP\_CONTENT\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LABEL\_DUP\_CONTENT\_FLAG", 234,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

LABEL\_NSFW\_HI\_PRC\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LABEL\_NSFW\_HI\_PRC\_FLAG", 235,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

LABEL\_NSFW\_HI\_RCL\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LABEL\_NSFW\_HI\_RCL\_FLAG", 236,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

LABEL\_SPAM\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LABEL\_SPAM\_FLAG", 237,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

LABEL\_SPAM\_HI\_RCL\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LABEL\_SPAM\_HI\_RCL\_FLAG", 238,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

// please save this bit for other safety labels

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_4\_31\_1(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS\_4\_31\_1", 239,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

// Integer 5

WEIGHTED\_RETWEET\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"WEIGHTED\_RETWEET\_COUNT",

240,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

WEIGHTED\_REPLY\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"WEIGHTED\_REPLY\_COUNT",

241,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

WEIGHTED\_FAVORITE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"WEIGHTED\_FAVORITE\_COUNT",

242,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

WEIGHTED\_QUOTE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"WEIGHTED\_QUOTE\_COUNT",

243,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

// Integer 6

// Periscope features

PERISCOPE\_EXISTS(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"PERISCOPE\_EXISTS", 245,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

PERISCOPE\_HAS\_BEEN\_FEATURED(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"PERISCOPE\_HAS\_BEEN\_FEATURED", 246,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

PERISCOPE\_IS\_CURRENTLY\_FEATURED(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"PERISCOPE\_IS\_CURRENTLY\_FEATURED", 247,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

PERISCOPE\_IS\_FROM\_QUALITY\_SOURCE(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"PERISCOPE\_IS\_FROM\_QUALITY\_SOURCE", 248,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

PERISCOPE\_IS\_LIVE(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"PERISCOPE\_IS\_LIVE", 249,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

IS\_TRENDING\_NOW\_FLAG(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"IS\_TRENDING\_NOW\_FLAG", 292,

FlagFeatureFieldType.FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

// remaining bits for integer 6 (starting bit 6, 26 remaining bits)

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_7\_6\_26(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS\_7\_6\_26", 250,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

// Decaying engagement counters

// Integer 7

DECAYED\_RETWEET\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"DECAYED\_RETWEET\_COUNT",

251,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

DECAYED\_REPLY\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"DECAYED\_REPLY\_COUNT",

252,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

DECAYED\_FAVORITE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"DECAYED\_FAVORITE\_COUNT",

253,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

DECAYED\_QUOTE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"DECAYED\_QUOTE\_COUNT",

254,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

// Fake engagement counters. The fake here is in the sense of spam, not in the sense of testing.

// Refer to [JIRA SEARCHQUAL-10736 Remove Fake Engagements in Search] for more details.

// Integer 8

FAKE\_RETWEET\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"FAKE\_RETWEET\_COUNT", 269,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

FAKE\_REPLY\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"FAKE\_REPLY\_COUNT", 270,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

FAKE\_FAVORITE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"FAKE\_FAVORITE\_COUNT", 271,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

FAKE\_QUOTE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"FAKE\_QUOTE\_COUNT", 272,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

// Last engagement timestamps. These features use the Tweet's creation time as base and

// are incremented every 1 hour

// Integer 9

LAST\_RETWEET\_SINCE\_CREATION\_HRS(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LAST\_RETWEET\_SINCE\_CREATION\_HRS",

273,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.NONE),

LAST\_REPLY\_SINCE\_CREATION\_HRS(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LAST\_REPLY\_SINCE\_CREATION\_HRS",

274,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.NONE),

LAST\_FAVORITE\_SINCE\_CREATION\_HRS(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LAST\_FAVORITE\_SINCE\_CREATION\_HRS",

275,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.NONE),

LAST\_QUOTE\_SINCE\_CREATION\_HRS(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"LAST\_QUOTE\_SINCE\_CREATION\_HRS",

276,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.NONE),

// 4 bits hashtag count, mention count and stock count (SEARCH-24336)

// Integer 10

NUM\_HASHTAGS\_V2(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"NUM\_HASHTAGS\_V2",

277,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.NONE

),

NUM\_MENTIONS\_V2(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"NUM\_MENTIONS\_V2",

278,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.NONE

),

NUM\_STOCKS(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"NUM\_STOCKS",

279,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.NONE

),

// Integer 11

// Blink engagement counters

BLINK\_RETWEET\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"BLINK\_RETWEET\_COUNT",

280,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

BLINK\_REPLY\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"BLINK\_REPLY\_COUNT",

281,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

BLINK\_FAVORITE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"BLINK\_FAVORITE\_COUNT",

282,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

BLINK\_QUOTE\_COUNT(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"BLINK\_QUOTE\_COUNT",

283,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.SMART\_INTEGER\_NORMALIZER),

// Integer 10 (remaining)

// Production Toxicity and PBlock score from HML (go/toxicity, go/pblock)

TOXICITY\_SCORE(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"TOXICITY\_SCORE", 284,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

PBLOCK\_SCORE(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"PBLOCK\_SCORE", 285,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

// Integer 12

// Experimental health model scores from HML

EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_1(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_1", 286,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_2(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_2", 287,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_3(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_3", 288,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

// remaining bits for index 12 (unused\_bits\_12)

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_12\_30\_2(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS\_12\_30\_2", 289,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

// Integer 13

// Experimental health model scores from HML (cont.)

EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_4(

EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_4", 290,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

// Production pSpammyTweet score from HML (go/pspammytweet)

P\_SPAMMY\_TWEET\_SCORE(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"P\_SPAMMY\_TWEET\_SCORE", 291,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

// Production pReportedTweet score from HML (go/preportedtweet)

P\_REPORTED\_TWEET\_SCORE(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"P\_REPORTED\_TWEET\_SCORE", 293,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

// remaining bits for index 13 (unused\_bits\_13)

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_13\_30\_2(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS\_13\_30\_2", 294,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS

),

// Integer 14

// Health model scores from HML (cont.)

// Prod Spammy Tweet Content model score from Platform Manipulation (go/spammy-tweet-content)

SPAMMY\_TWEET\_CONTENT\_SCORE(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"SPAMMY\_TWEET\_CONTENT\_SCORE", 295,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS,

ThriftFeatureNormalizationType.PREDICTION\_SCORE\_NORMALIZER

),

// remaining bits for index 14 (unused\_bits\_14)

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_14\_10\_22(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS\_14\_10\_22", 296,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS

),

// Note that the integer block index i in the names UNUSED\_BITS{i}" below is 1-based, but the

// index j in UNUSED\_BITS\_{j}\_x\_y above is 0-based.

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_16(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS16", 216,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_17(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS17", 217,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_18(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS18", 218,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_19(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS19", 219,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS),

EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_20(EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

"UNUSED\_BITS20", 220,

FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD,

EarlybirdCluster.TWITTER\_IN\_MEMORY\_INDEX\_FORMAT\_ALL\_CLUSTERS);

// Filter field terms. These end up as terms in the "internal" field (id=18). So for example

// you can have a doc with field(internal) = "\_\_filter\_nullcast", "\_\_filter\_vine" and that will

// be a nullcast tweet with a vine link in it.

public static final String NULLCAST\_FILTER\_TERM = "nullcast";

public static final String VERIFIED\_FILTER\_TERM = "verified";

public static final String BLUE\_VERIFIED\_FILTER\_TERM = "blue\_verified";

public static final String NATIVE\_RETWEETS\_FILTER\_TERM = "nativeretweets";

public static final String QUOTE\_FILTER\_TERM = "quote";

public static final String REPLIES\_FILTER\_TERM = "replies";

public static final String CONSUMER\_VIDEO\_FILTER\_TERM = "consumer\_video";

public static final String PRO\_VIDEO\_FILTER\_TERM = "pro\_video";

public static final String VINE\_FILTER\_TERM = "vine";

public static final String PERISCOPE\_FILTER\_TERM = "periscope";

public static final String PROFILE\_GEO\_FILTER\_TERM = "profile\_geo";

public static final String SELF\_THREAD\_FILTER\_TERM = "self\_threads";

public static final String DIRECTED\_AT\_FILTER\_TERM = "directed\_at";

public static final String EXCLUSIVE\_FILTER\_TERM = "exclusive";

// Reserved terms for the internal field.

public static final String HAS\_POSITIVE\_SMILEY = "\_\_has\_positive\_smiley";

public static final String HAS\_NEGATIVE\_SMILEY = "\_\_has\_negative\_smiley";

public static final String IS\_OFFENSIVE = "\_\_is\_offensive";

// Facet fields

public static final String MENTIONS\_FACET = "mentions";

public static final String HASHTAGS\_FACET = "hashtags";

public static final String STOCKS\_FACET = "stocks";

public static final String VIDEOS\_FACET = "videos";

public static final String IMAGES\_FACET = "images";

public static final String NEWS\_FACET = "news";

public static final String LANGUAGES\_FACET = "languages";

public static final String SOURCES\_FACET = "sources";

public static final String TWIMG\_FACET = "twimg";

public static final String FROM\_USER\_ID\_FACET = "user\_id";

public static final String RETWEETS\_FACET = "retweets";

public static final String LINKS\_FACET = "links";

public static final String SPACES\_FACET = "spaces";

/\*\*

\* Used by the query parser to check that the operator of a [filter X] query is valid.

\* Also used by blender, though it probably shouldn't be.

\*/

public static final ImmutableSet<String> FACETS = ImmutableSet.<String>builder()

.add(MENTIONS\_FACET)

.add(HASHTAGS\_FACET)

.add(STOCKS\_FACET)

.add(VIDEOS\_FACET)

.add(IMAGES\_FACET)

.add(NEWS\_FACET)

.add(LINKS\_FACET)

.add(LANGUAGES\_FACET)

.add(SOURCES\_FACET)

.add(TWIMG\_FACET)

.add(SPACES\_FACET)

.build();

/\*\*

\* Used by blender to convert facet names to field names. We should find a way to get the

\* information we need in blender without needing this map.

\*/

public static final ImmutableMap<String, String> FACET\_TO\_FIELD\_MAP =

ImmutableMap.<String, String>builder()

.put(MENTIONS\_FACET, MENTIONS\_FIELD.getFieldName())

.put(HASHTAGS\_FACET, HASHTAGS\_FIELD.getFieldName())

.put(STOCKS\_FACET, STOCKS\_FIELD.getFieldName())

.put(VIDEOS\_FACET, VIDEO\_LINKS\_FIELD.getFieldName())

.put(IMAGES\_FACET, IMAGE\_LINKS\_FIELD.getFieldName())

.put(NEWS\_FACET, NEWS\_LINKS\_FIELD.getFieldName())

.put(LANGUAGES\_FACET, ISO\_LANGUAGE\_FIELD.getFieldName())

.put(SOURCES\_FACET, SOURCE\_FIELD.getFieldName())

.put(TWIMG\_FACET, TWIMG\_LINKS\_FIELD.getFieldName())

.put(LINKS\_FACET, LINKS\_FIELD.getFieldName())

.put(SPACES\_FACET, SPACE\_ID\_FIELD.getFieldName())

.build();

public static String getFacetSkipFieldName(String fieldName) {

return "\_\_has\_" + fieldName;

}

private final String fieldName;

private final int fieldId;

private final EnumSet<EarlybirdCluster> clusters;

private final FlagFeatureFieldType flagFeatureField;

private final UnusedFeatureFieldType unusedField;

// Only set for feature fields.

@Nullable

private final FeatureConfiguration featureConfiguration;

// Only set for feature fields.

private final ThriftFeatureNormalizationType featureNormalizationType;

// To simplify field configurations and reduce duplicate code, we give clusters a default value

EarlybirdFieldConstant(String fieldName, int fieldId) {

this(fieldName, fieldId, EarlybirdCluster.GENERAL\_PURPOSE\_CLUSTERS, null);

}

EarlybirdFieldConstant(String fieldName, int fieldId, Set<EarlybirdCluster> clusters) {

this(fieldName, fieldId, clusters, null);

}

EarlybirdFieldConstant(String fieldName, int fieldId, EarlybirdCluster cluster) {

this(fieldName, fieldId, ImmutableSet.<EarlybirdCluster>of(cluster), null);

}

/\*\*

\* Base field name is needed here in order to construct the full

\* name of the feature. Our convention is that a feature should be named

\* as: baseFieldName.featureName. For example: encoded\_tweet\_features.retweet\_count.

\*/

EarlybirdFieldConstant(

String baseName,

String fieldName,

int fieldId,

FlagFeatureFieldType flagFeatureField,

Set<EarlybirdCluster> clusters) {

this((baseName + SchemaBuilder.CSF\_VIEW\_NAME\_SEPARATOR + fieldName).toLowerCase(),

fieldId, clusters, flagFeatureField, null);

}

EarlybirdFieldConstant(

String baseName,

String fieldName,

int fieldId,

FlagFeatureFieldType flagFeatureField,

UnusedFeatureFieldType unusedField,

Set<EarlybirdCluster> clusters) {

this((baseName + SchemaBuilder.CSF\_VIEW\_NAME\_SEPARATOR + fieldName).toLowerCase(),

fieldId, clusters, flagFeatureField, unusedField, null);

}

EarlybirdFieldConstant(

String baseName,

String fieldName,

int fieldId,

FlagFeatureFieldType flagFeatureField,

Set<EarlybirdCluster> clusters,

ThriftFeatureNormalizationType featureNormalizationType) {

this((baseName + SchemaBuilder.CSF\_VIEW\_NAME\_SEPARATOR + fieldName).toLowerCase(),

fieldId, clusters, flagFeatureField, UnusedFeatureFieldType.USED\_FEATURE\_FIELD,

featureNormalizationType, null);

}

/\*\*

\* Constructor.

\*/

EarlybirdFieldConstant(String fieldName, int fieldId, Set<EarlybirdCluster> clusters,

@Nullable FeatureConfiguration featureConfiguration) {

this(fieldName, fieldId, clusters, FlagFeatureFieldType.NON\_FLAG\_FEATURE\_FIELD,

featureConfiguration);

}

/\*\*

\* Constructor.

\*/

EarlybirdFieldConstant(String fieldName,

int fieldId,

Set<EarlybirdCluster> clusters,

FlagFeatureFieldType flagFeatureField,

@Nullable FeatureConfiguration featureConfiguration) {

this(fieldName, fieldId, clusters, flagFeatureField,

UnusedFeatureFieldType.USED\_FEATURE\_FIELD, featureConfiguration);

}

/\*\*

\* Constructor.

\*/

EarlybirdFieldConstant(String fieldName,

int fieldId,

Set<EarlybirdCluster> clusters,

FlagFeatureFieldType flagFeatureField,

UnusedFeatureFieldType unusedField,

@Nullable FeatureConfiguration featureConfiguration) {

this(fieldName, fieldId, clusters, flagFeatureField, unusedField, null, featureConfiguration);

}

/\*\*

\* Constructor.

\*/

EarlybirdFieldConstant(String fieldName,

int fieldId,

Set<EarlybirdCluster> clusters,

FlagFeatureFieldType flagFeatureField,

UnusedFeatureFieldType unusedField,

@Nullable ThriftFeatureNormalizationType featureNormalizationType,

@Nullable FeatureConfiguration featureConfiguration) {

this.fieldId = fieldId;

this.fieldName = fieldName;

this.clusters = EnumSet.copyOf(clusters);

this.flagFeatureField = flagFeatureField;

this.unusedField = unusedField;

this.featureNormalizationType = featureNormalizationType;

this.featureConfiguration = featureConfiguration;

}

// Override toString to make replacing StatusConstant Easier.

@Override

public String toString() {

return fieldName;

}

public boolean isValidFieldInCluster(EarlybirdCluster cluster) {

return clusters.contains(cluster);

}

public String getFieldName() {

return fieldName;

}

public int getFieldId() {

return fieldId;

}

public FlagFeatureFieldType getFlagFeatureField() {

return flagFeatureField;

}

public boolean isFlagFeatureField() {

return flagFeatureField == FlagFeatureFieldType.FLAG\_FEATURE\_FIELD;

}

public boolean isUnusedField() {

return unusedField == UnusedFeatureFieldType.UNUSED\_FEATURE\_FIELD;

}

@Nullable

public FeatureConfiguration getFeatureConfiguration() {

return featureConfiguration;

}

@Nullable

public ThriftFeatureNormalizationType getFeatureNormalizationType() {

return featureNormalizationType;

}

}

private static final Map<String, EarlybirdFieldConstant> NAME\_TO\_ID\_MAP;

private static final Map<Integer, EarlybirdFieldConstant> ID\_TO\_FIELD\_MAP;

static {

ImmutableMap.Builder<String, EarlybirdFieldConstant> nameToIdMapBuilder =

ImmutableMap.builder();

ImmutableMap.Builder<Integer, EarlybirdFieldConstant> idToFieldMapBuilder =

ImmutableMap.builder();

Set<String> fieldNameDupDetector = Sets.newHashSet();

Set<Integer> fieldIdDupDetector = Sets.newHashSet();

for (EarlybirdFieldConstant fc : EarlybirdFieldConstant.values()) {

if (fieldNameDupDetector.contains(fc.getFieldName())) {

throw new IllegalStateException("detected fields sharing field name: " + fc.getFieldName());

}

if (fieldIdDupDetector.contains(fc.getFieldId())) {

throw new IllegalStateException("detected fields sharing field id: " + fc.getFieldId());

}

fieldNameDupDetector.add(fc.getFieldName());

fieldIdDupDetector.add(fc.getFieldId());

nameToIdMapBuilder.put(fc.getFieldName(), fc);

idToFieldMapBuilder.put(fc.getFieldId(), fc);

}

NAME\_TO\_ID\_MAP = nameToIdMapBuilder.build();

ID\_TO\_FIELD\_MAP = idToFieldMapBuilder.build();

}

// This define the list of boolean features, but the name does not have "flag" inside. This

// definition is only for double checking purpose to prevent code change mistakes. The setting

// of the flag feature is based on FlagFeatureFieldType.FLAG\_FEATURE\_FIELD.

public static final Set<EarlybirdFieldConstants.EarlybirdFieldConstant> EXTRA\_FLAG\_FIELDS =

Sets.newHashSet(EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_SENSITIVE\_CONTENT);

public static final String FLAG\_STRING = "flag";

private static final List<EarlybirdFieldConstant> FLAG\_FEATURE\_FIELDS;

static {

ImmutableList.Builder<EarlybirdFieldConstant> flagFieldBuilder = ImmutableList.builder();

for (EarlybirdFieldConstant fc : EarlybirdFieldConstant.values()) {

if (fc.getFlagFeatureField() == FlagFeatureFieldType.FLAG\_FEATURE\_FIELD

&& !fc.isUnusedField()) {

flagFieldBuilder.add(fc);

}

}

FLAG\_FEATURE\_FIELDS = flagFieldBuilder.build();

}

/\*\*

\* Get all the flag features meaning that they are boolean features with only 1 bit in the packed

\* feature encoding.

\*/

public static Collection<EarlybirdFieldConstant> getFlagFeatureFields() {

return FLAG\_FEATURE\_FIELDS;

}

/\*\*

\* Get the EarlybirdFieldConstant for the specified field.

\*/

public static EarlybirdFieldConstant getFieldConstant(String fieldName) {

EarlybirdFieldConstant field = NAME\_TO\_ID\_MAP.get(fieldName);

if (field == null) {

throw new IllegalArgumentException("Unknown field: " + fieldName);

}

return field;

}

/\*\*

\* Get the EarlybirdFieldConstant for the specified field.

\*/

public static EarlybirdFieldConstant getFieldConstant(int fieldId) {

EarlybirdFieldConstant field = ID\_TO\_FIELD\_MAP.get(fieldId);

if (field == null) {

throw new IllegalArgumentException("Unknown field: " + fieldId);

}

return field;

}

/\*\*

\* Determines if there's a field with the given ID.

\*/

public static boolean hasFieldConstant(int fieldId) {

return ID\_TO\_FIELD\_MAP.keySet().contains(fieldId);

}

@Override

public final int getFieldID(String fieldName) {

return getFieldConstant(fieldName).getFieldId();

}

public static final String formatGeoType(ThriftGeoLocationSource source) {

return "\_\_geo\_location\_type\_" + source.name().toLowerCase();

}

}