package com.twitter.search.common.schema;

import com.google.common.base.Preconditions;

import org.apache.lucene.index.DocValuesType;

import org.apache.lucene.index.IndexOptions;

import org.apache.lucene.util.BytesRef;

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

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

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

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

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

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

import com.twitter.search.common.util.analysis.IntTermAttributeImpl;

import com.twitter.search.common.util.analysis.LongTermAttributeImpl;

import com.twitter.search.common.util.analysis.SortableLongTermAttributeImpl;

public final class SchemaUtil {

private SchemaUtil() {

}

/\*\*

\* Get the a fixed CSF field's number of values per doc.

\* @param schema the Schema for the index

\* @param fieldId the field id the CSF field - the field must be of binary integer type and

\* in fixed size

\* @return the number of values per doc

\*/

public static int getCSFFieldFixedLength(ImmutableSchemaInterface schema, int fieldId) {

final Schema.FieldInfo fieldInfo = Preconditions.checkNotNull(schema.getFieldInfo(fieldId));

return getCSFFieldFixedLength(fieldInfo);

}

/\*\*

\* Get the a fixed CSF field's number of values per doc.

\* @param schema the Schema for the index

\* @param fieldName the field name of the CSF field - the field must be of binary integer type

\* and in fixed size

\* @return the number of values per doc

\*/

public static int getCSFFieldFixedLength(ImmutableSchemaInterface schema, String fieldName) {

final Schema.FieldInfo fieldInfo = Preconditions.checkNotNull(schema.getFieldInfo(fieldName));

return getCSFFieldFixedLength(fieldInfo);

}

/\*\*

\* Get the a fixed CSF field's number of values per doc.

\* @param fieldInfo the field of the CSF field - the field must be of binary integer type

\* and in fixed size

\* @return the number of values per doc

\*/

public static int getCSFFieldFixedLength(Schema.FieldInfo fieldInfo) {

final EarlybirdFieldType fieldType = fieldInfo.getFieldType();

Preconditions.checkState(fieldType.docValuesType() == DocValuesType.BINARY

&& fieldType.getCsfType() == ThriftCSFType.INT);

return fieldType.getCsfFixedLengthNumValuesPerDoc();

}

/\*\* Converts the given value to a BytesRef instance, according to the type of the given field. \*/

public static BytesRef toBytesRef(Schema.FieldInfo fieldInfo, String value) {

EarlybirdFieldType fieldType = fieldInfo.getFieldType();

Preconditions.checkArgument(fieldType.indexOptions() != IndexOptions.NONE);

IndexedNumericFieldSettings numericSetting = fieldType.getNumericFieldSettings();

if (numericSetting != null) {

if (!numericSetting.isUseTwitterFormat()) {

throw new UnsupportedOperationException(

"Numeric field not using Twitter format: cannot drill down.");

}

ThriftNumericType numericType = numericSetting.getNumericType();

switch (numericType) {

case INT:

try {

return IntTermAttributeImpl.copyIntoNewBytesRef(Integer.parseInt(value));

} catch (NumberFormatException e) {

throw new UnsupportedOperationException(

String.format("Cannot parse value for int field %s: %s",

fieldInfo.getName(), value),

e);

}

case LONG:

try {

return numericSetting.isUseSortableEncoding()

? SortableLongTermAttributeImpl.copyIntoNewBytesRef(Long.parseLong(value))

: LongTermAttributeImpl.copyIntoNewBytesRef(Long.parseLong(value));

} catch (NumberFormatException e) {

throw new UnsupportedOperationException(

String.format("Cannot parse value for long field %s: %s",

fieldInfo.getName(), value),

e);

}

default:

throw new UnsupportedOperationException(

String.format("Unsupported numeric type for field %s: %s",

fieldInfo.getName(), numericType));

}

}

return new BytesRef(value);

}

}