KIP-955: Add stream-table join on foreign key

Status

Current state: Under Discussion

Discussion thread: here

JIRA: *KAFKA-15299*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Reduce the gap between the semantics of relational databases and data streams.

When business entity is required to be stored in relational database the data model for this entity is typically normalized. This data normalization in general results in using several tables to store each data entity type with several types of relations between tables. If foreign key relationship is used between database tables then it is easy to assemble the business entity data from corresponding DB tables when necessary using SQL SELECT statements with foreign key table join.

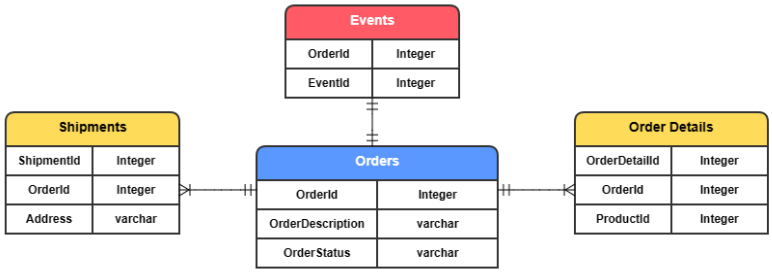
Also reading the data entities from the application database usually associated with some specific "business" event that could be represented as a stream of business events resulting from the business activity. For example Order data entity shown on the diagram below could be required to be extracted for the order processing as the result of "ORDER COMPLETED EVENT".

So if we create KStream from the stream of business events and create a KTable(s) from all database tables that store business entity data (for example using Change Data Capture) we can assemble complete Business Entity using stream-table joins and then aggregating on event key.

For the Order Completion example above the sequence of order extraction from Kafka topics is as follows:

- As the first step Business event stream could be joined with business entity parent table using stream-table join on PK. Using the order management example on the diagram below this will be Order Processing Events stream joined with Orders KTable on OrderId key resulting in a stream of completed orders.
- As the second step the stream of completed orders could be left joined with Order Details table using the foreign key (OrderId) resulting in a stream of competed orders with corresponding order details. If order to order details relation 1:n then additional aggregation on OrderId will be required.

This KIP proposes the new Kafka Streams feature with capability to join the KStream with KTable using left Foreign Key Join

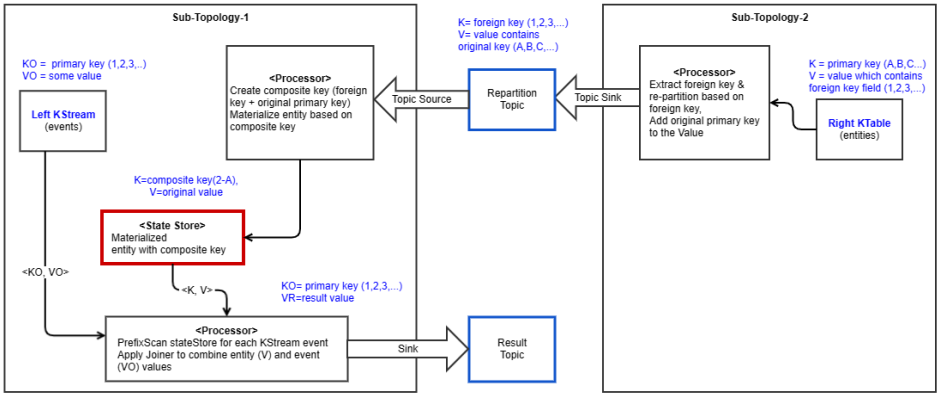


Public Interfaces

```
/**
 * Join records of this {@code KStream} with {@code KTable} using non-windowed left join.
 * <p>
 * This is a foreign key join, where the joining key is determined by the {@code foreignKeyExtractor}.
 *
 * @param rightTable          the {@code KTable} on the right side of join to be joined with this
 *                            {@code KStream}. Keyed by KO.
 * @param foreignKeyExtractor a {@link Function} that extracts the key (KO) from this table's value (V). If the
 *                            result is null, the update is ignored as invalid.
 * @param joiner              a {@link ValueJoiner} that computes the join result for a pair of matching records
 * @param <VR>                the value type of the result {@code KTable}
 * @param <KO>                the key type of the right {@code KTable}
 * @param <VO>                the value type of the right {@code KTable}
 * @return a {@code KStream} that contains the result of joining this stream with {@code rightTable}
 */
<VR, KO, VO> KStream<K, VR> leftJoin(final KTable<KO, VO> rightTable,
                                     final Function<VO, KO> foreignKeyExtractor,
                                     final ValueJoiner<V, VO, VR> joiner);
```

Proposed Changes

The design for this new public interface is based on old KIP-213 (Table-Table join on foreign key). For stream-table left join on FK the KIP-213 design could be significantly simplified because output stream events are only generated for each input stream events. Please refer to KIP-213 for the design details on most complex part which is materializing the composite key in State Store and scanning for the composite key prefix in State Store in order to match stream events with the table entities based on foreign key.



1) If the **Right KTable** is updated, it propagates a single update to the State Store

2) If the **Left KStream** is updated, it performs a prefixScan on **Materialized CombinedKey StateStore** and propagates all records which have the required prefix through the workflow.

Compatibility, Deprecation, and Migration Plan

- There is no impact to existing users because this is a new Interface

Test Plan

Proposed new functionality should be tested with the functional test cases to ensure that results of stream-table foreign key join are consistent with results of the standard RDBMS SQL for foreign key left join between two database tables with foreign key relations. The database table that has foreign key field corresponds to the KTable that are on the right of the left join and database table that has a primary key matching foreign key corresponds to the KStream to the left of join.

Rejected Alternatives

None