Data Engineering Project in der kSQL-DB

kSQL-Queries zum Prozessieren der Coincap-Daten

```
CREATE STREAM COINCAP_STREAM (
TIMESTAMP BIGINT,
DATA ARRAY<STRUCT<id   VARCHAR,
rank VARCHAR,
symbol VARCHAR,
priceUsd DOUBLE,
volumeUsd24Hr DOUBLE,
marketCapUsd DOUBLE,
CHANGEPERCENT24HR DOUBLE>>)
WITH (KAFKA_TOPIC='Coin_data', VALUE_FORMAT='JSON');
```

```
CREATE STREAM COINCAP_STREAM2 WITH (KAFKA_TOPIC='COINCAP_STREAM2', PARTITIONS=6,
REPLICAS=3) AS SELECT
TIMESTAMPTOSTRING(TIMESTAMP, 'yyyy-MM-dd HH:mm') TIMESTAMP_FORMATTED,
EXPLODE(COINCAP_STREAM.DATA) as data
FROM COINCAP_STREAM COINCAP_STREAM
EMIT CHANGES;
```

```
CREATE TABLE COINCAP_Table WITH (KAFKA_TOPIC='Coincap_Table', KEY_FORMAT='JSON',
PARTITIONS=1, REPLICAS=3, VALUE_FORMAT='JSON') AS SELECT
data->symbol+','+TIMESTAMP_FORMATTED as TIMESTAMP_SYMBOL_KEY,
AVG(data->priceusd) as AVG_priceusd,
AVG(data->volumeusd24hr) as AVG_volumeusd24hr,
AVG(data-> CHANGEPERCENT24HR) as AVG_CHANGEPERCENT24HR,
AVG(data->marketcapusd) as AVG_marketCapUsd
FROM COINCAP_STREAM2
GROUP BY data->symbol+','+TIMESTAMP_FORMATTED
EMIT CHANGES;
```

kSQL-Queries zum Prozessieren der Twitter-Daten

```
CREATE STREAM TWITTER_DATA (
CREATED_AT VARCHAR,
ID VARCHAR,
ID_STR VARCHAR,
LANG VARCHAR,
TIMESTAMP_MS BIGINT,
ENTITIES STRUCT<HASHTAGS ARRAY<STRUCT<TEXT VARCHAR>>>)
WITH (KAFKA_TOPIC='Twitter_data', KEY_FORMAT='KAFKA', VALUE_FORMAT='JSON');
```

```
CREATE STREAM TWITTER_STREAM2 WITH (KAFKA_TOPIC='TWITTER_STREAM2', PARTITIONS=1,
REPLICAS=3) AS SELECT
  TWITTER_DATA.CREATED_AT CREATED_AT,
  TWITTER_DATA.ID ID,
  TIMESTAMPTOSTRING(TWITTER_DATA.TIMESTAMP_MS, 'yyyy-MM-dd HH:mm') DATETIME,
  EXPLODE(TWITTER_DATA.ENTITIES->HASHTAGS)->TEXT HASHTAG
FROM TWITTER_DATA
EMIT CHANGES;
```

```
CREATE STREAM TWITTER_STREAM3 WITH (KAFKA_TOPIC='TWITTER_STREAM3', PARTITIONS=1,
REPLICAS=3) AS SELECT
CREATED_AT,
ID,
DATETIME,
HASHTAG,
HASHTAG +','+ DATETIME as HASHDATE
FROM TWITTER_STREAM2
EMIT CHANGES;
```

```
CREATE TABLE Twitter_Table WITH (KAFKA_TOPIC='Twitter_Table', KEY_FORMAT='JSON',
PARTITIONS=1, REPLICAS=3, VALUE_FORMAT='JSON') AS SELECT
count(id) as Anzahl_Tweets,
HASHDATE
from Twitter_Stream3
group by HASHDATE
emit changes;
```

kSQL-Query zum Joinen der Coincap-Daten mit den Twitter-Daten

```
CREATE TABLE COINCAP_TWITTER WITH (KAFKA_TOPIC='COINCAP_TWITTER', PARTITIONS=1,
REPLICAS=3) AS SELECT
A.TIMESTAMP_SYMBOL_KEY,
A.AVG_PRICEUSD,
A. AVG_VOLUMEUSD24HR,
A. AVG_CHANGEPERCENT24HR,
A. AVG_marketCapUsd,
B.ANZAHL_TWEETS ANZAHL_TWEETS
FROM COINCAP_TABLE A
LEFT OUTER JOIN TWITTER_TABLE B ON ((A.TIMESTAMP_SYMBOL_KEY = B.HASHDATE))
EMIT CHANGES;
```