Lab 9

4. Run the commands inside KSQL

CREATE STREAM ConfluentTextLinesStream (line VARCHAR) WITH (KAFKA_TOPIC='ConfluentTextLinesTopic', VALUE_FORMAT='KAFKA');

```
CREATE STREAM ConfluentTextLinesStream (line VARCHAR) WITH (KAFKA_TOPIC='ConfluentTextLinesTopic', VALUE_FORMAT='KAFKA');
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE STREAM CONFLUENTTEXTLINESSTREAM (LINE STRING) WITH (CLEANUP_POLICY='delete',
KAFKA_TOPIC='ConfluentTextLinesTopic', KEY_FORMAT='KAFKA', VALUE_FORMAT='KAFKA');",
  "commandId": "stream/`CONFLUENTTEXTLINESSTREAM`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Stream created",
    "queryId": null
  },
  "commandSequenceNumber": 6,
  "warnings": []
}
```


SELECT * FROM ConfluentTextLinesStream EMIT CHANGES;

```
SELECT * FROM ConfluentTextLinesStream EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Running... Stop

Data structure
STREAM

Total messages
- -

Messages/sec
- -

Total message bytes
- -

Filter by keyword


```
{"LINE":"\"beta\""}
```

```
{"LINE":"\"alpha\""}
```

SELECT Ucase(line) FROM ConfluentTextLinesStream EMIT CHANGES;

```
1  SELECT Ucase(line) FROM ConfluentTextLinesStream EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Running... Stop

Data structure
STREAM

Total messages
- -

Messages/sec
- -

Total message bytes

Filter by keyword

```
{"KSQL_COL_0":"\"BETA\""}
```

```
{"KSQL_COL_0":"\"ALPHA\""}
```

CREATE STREAM UppercasedTextLinesStream WITH (KAFKA_TOPIC='UppercasedTextLinesTopic', VALUE_FORMAT='KAFKA') AS SELECT Ucase(line) FROM ConfluentTextLinesStream;

```
1  CREATE STREAM UppercasedTextLinesStream WITH (KAFKA_TOPIC='UppercasedTextLinesTopic',  VALUE_FORMAT='KAFKA') AS SELECT Ucas
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
1  {
2    "@type": "currentStatus",
3    "statementText": "CREATE STREAM UPPERCASEDTEXTLINESSTREAM WITH (CLEANUP_POLICY='delete',
     KAFKA_TOPIC='UppercasedTextLinesTopic', PARTITIONS=10, REPLICAS=3, RETENTION_MS=604800000, VALUE_FORMAT='KAFKA') AS
     SELECT UCASE(CONFLUENTTEXTLINESSTREAM.LINE) KSQL_COL_0\nFROM CONFLUENTTEXTLINESSTREAM CONFLUENTTEXTLINESSTREAM\nEMIT
     CHANGES;",
4    "commandId": "stream/`UPPERCASEDTEXTLINESSTREAM`/create",
5    "commandStatus": {
6      "status": "SUCCESS",
7      "message": "Created query with ID CSAS_UPPERCASEDTEXTLINESSTREAM_7",
8      "queryId": "CSAS_UPPERCASEDTEXTLINESSTREAM_7"
9    },
10   "commandSequenceNumber": 8,
11   "warnings": []
12 }
```

5. Check UppercasedTextLinesTopic topic;

SELECT * FROM UppercasedTextLinesStream EMIT CHANGES;

```
SELECT * FROM UppercasedTextLinesStream EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Running... Stop

Data structure
STREAM

Total messages
- -

Messages/sec
- -

Total message bytes
- -

Filter by keyword

```
{"KSQL_COL_0":"\"BETA\""}
```

```
{"KSQL_COL_0":"\"ALPHA\""}
```

CREATE STREAM readings (sensor VARCHAR KEY, val DOUBLE, location VARCHAR) WITH (
kafka_topic='readings',
partitions=6,
value_format='JSON');

```
CREATE STREAM readings  (
  sensor VARCHAR KEY,
  val DOUBLE,
  location VARCHAR) WITH (
    kafka_topic='readings',
    partitions=6,
    value_format='JSON');
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE STREAM READINGS (SENSOR STRING KEY, VAL DOUBLE, LOCATION STRING) WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='readings', KEY_FORMAT='KAFKA', PARTITIONS=6, VALUE_FORMAT='JSON');",
  "commandId": "stream/`READINGS`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Stream created",
    "queryId": null
  },
  "commandSequenceNumber": 10,
  "warnings": []
}
```

INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 45, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-2', 41, 'motor');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 42, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 42, 'muffler');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 48, 'muffler');

```
INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 45, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-2', 41, 'motor');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 42, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 42, 'muffler');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 48, 'muffler');
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query

Data structure
- -

Total messages
- -

Messages/sec
- -

Filter by keyword

No new messages

The message browser shows messages that have arrived since this page was opened.

4. Run some queries

CREATE STREAM clean AS SELECT sensor, val, UCASE(location) as location FROM readings EMIT CHANGES;

```
CREATE STREAM clean AS SELECT sensor, val, UCASE(location) as location FROM readings EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE STREAM CLEAN WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-n7kwdCLEAN', PARTITIONS=6,
REPLICAS=3, RETENTION_MS=604800000) AS SELECT\n  READINGS.SENSOR SENSOR,\n  READINGS.VAL VAL,\n
UCASE(READINGS.LOCATION) LOCATION\nFROM READINGS READINGS\nEMIT CHANGES;",
  "commandId": "stream/`CLEAN`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Created query with ID CSAS_CLEAN_11",
    "queryId": "CSAS_CLEAN_11"
  },
  "commandSequenceNumber": 12,
  "warnings": []
}
```

SELECT sensor, val, location FROM readings WHERE sensor='sensor-1' EMIT CHANGES;

```
SELECT sensor, val, location FROM readings WHERE sensor='sensor-1'  EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Running... Stop

Data structure
STREAM

Total messages
- -

Messages/sec
- -

Total message bytes

Filter by keyword

Newest

```
{"SENSOR":"sensor-1","VAL":42,"LOCATION":"wheel"}
```

```
{"SENSOR":"sensor-1","VAL":45,"LOCATION":"wheel"}
```

CREATE STREAM high_readings AS SELECT sensor, val, location FROM clean where val > 42 EMIT CHANGES;

```
CREATE STREAM high_readings AS SELECT sensor, val, location FROM clean where val > 42 EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE STREAM HIGH_READINGS WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-n7kwdHIGH_READINGS',
PARTITIONS=6, REPLICAS=3, RETENTION_MS=604800000) AS SELECT\n  CLEAN.SENSOR SENSOR,\n  CLEAN.VAL VAL,\n  CLEAN.LOCATION
LOCATION\nFROM CLEAN CLEAN\nWHERE (CLEAN.VAL > 42)\nEMIT CHANGES;",
  "commandId": "stream/`HIGH_READINGS`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Created query with ID CSAS_HIGH_READINGS_13",
    "queryId": "CSAS_HIGH_READINGS_13"
  },
  "commandSequenceNumber": 14,
  "warnings": []
}
```

5. Produce some events into readings topic
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 36, 'motor');

```
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 36, 'motor');
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop | Run query

Data structure
- -

Total messages
- -

Messages/sec
- -

Total message bytes
- -

Filter by keyword

No new messages

The message browser shows messages that have arrived since this page was opened.

6. Run some queries
CREATE STREAM high_pri AS SELECT sensor, val, UCASE(location) as location
FROM readings where val > 42 EMIT CHANGES;

```
CREATE STREAM high_pri AS SELECT sensor, val, UCASE(location) as location
FROM readings where val > 42 EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop | Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE STREAM HIGH_PRI WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-n7kwdHIGH_PRI',
PARTITIONS=6, REPLICAS=3, RETENTION_MS=604800000) AS SELECT\n  READINGS.SENSOR SENSOR,\n  READINGS.VAL VAL,\n
UCASE(READINGS.LOCATION) LOCATION\nFROM READINGS READINGS\nWHERE (READINGS.VAL > 42)\nEMIT CHANGES;",
  "commandId": "stream/`HIGH_PRI`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Created query with ID CSAS_HIGH_PRI_15",
    "queryId": "CSAS_HIGH_PRI_15"
  },
  "commandSequenceNumber": 16,
  "warnings": []
}
```

CREATE STREAM by_location AS SELECT * FROM high_pri
PARTITION BY location EMIT CHANGES;

```
CREATE STREAM by_location AS SELECT * FROM high_pri
PARTITION BY location EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE STREAM BY_LOCATION WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-n7kwdBY_LOCATION',
PARTITIONS=6, REPLICAS=3, RETENTION_MS=604800000) AS SELECT *\nFROM HIGH_PRI HIGH_PRI\nPARTITION BY
HIGH_PRI.LOCATION\nEMIT CHANGES;",
  "commandId": "stream/`BY_LOCATION`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Created query with ID CSAS_BY_LOCATION_17",
    "queryId": "CSAS_BY_LOCATION_17"
  },
  "commandSequenceNumber": 18,
  "warnings": []
}
```


CREATE STREAM by_val AS SELECT * FROM high_pri
PARTITION BY val EMIT CHANGES;

```
CREATE STREAM by_val AS SELECT * FROM high_pri
PARTITION BY val EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE STREAM BY_VAL WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-n7kwdBY_VAL', PARTITIONS=6,
REPLICAS=3, RETENTION_MS=604800000) AS SELECT *\nFROM HIGH_PRI HIGH_PRI\nPARTITION BY HIGH_PRI.VAL\nEMIT CHANGES;",
  "commandId": "stream/`BY_VAL`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Created query with ID CSAS_BY_VAL_19",
    "queryId": "CSAS_BY_VAL_19"
  },
  "commandSequenceNumber": 20,
  "warnings": []
}
```

INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 45, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-2', 41, 'motor');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 42, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 42, 'muffler');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 48, 'muffler');

```
INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 45, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-2', 41, 'motor');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-1', 42, 'wheel');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 42, 'muffler');
INSERT INTO readings (sensor, val, location) VALUES ('sensor-3', 48, 'muffler');
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query

Data structure

Total messages

Filter by keyword

CREATE TABLE avg_readings AS
SELECT sensor, AVG(val) as avg
FROM readings
GROUP BY sensor
EMIT CHANGES;

```
CREATE TABLE avg_readings AS
    SELECT sensor, AVG(val) as avg
    FROM readings
    GROUP BY sensor
    EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE TABLE AVG_READINGS WITH (CLEANUP_POLICY='compact', KAFKA_TOPIC='pksqlc-n7kwdAVG_READINGS',
PARTITIONS=6, REPLICAS=3, RETENTION_MS=604800000) AS SELECT\n  READINGS.SENSOR SENSOR,\n  AVG(READINGS.VAL) AVG\nFROM
READINGS READINGS\nGROUP BY READINGS.SENSOR\nEMIT CHANGES;",
  "commandId": "table/`AVG_READINGS`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Created query with ID CTAS_AVG_READINGS_21",
    "queryId": "CTAS_AVG_READINGS_21"
  },
  "commandSequenceNumber": 22,
  "warnings": []
}
```

```
CREATE TABLE part_avg AS
    SELECT location, AVG(val) as avg
    FROM readings
    GROUP BY location
    EMIT CHANGES;
```

```
CREATE TABLE part_avg AS
    SELECT location, AVG(val) as avg
    FROM readings
    GROUP BY location
    EMIT CHANGES;
```

Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query


```
{
  "@type": "currentStatus",
  "statementText": "CREATE TABLE PART_AVG WITH (CLEANUP_POLICY='compact', KAFKA_TOPIC='pksqlc-n7kwdPART_AVG',
PARTITIONS=6, REPLICAS=3, RETENTION_MS=604800000) AS SELECT\n  READINGS.LOCATION LOCATION,\n  AVG(READINGS.VAL)
AVG\nFROM READINGS READINGS\nGROUP BY READINGS.LOCATION\nEMIT CHANGES;",
  "commandId": "table/`PART_AVG`/create",
  "commandStatus": {
    "status": "SUCCESS",
    "message": "Created query with ID CTAS_PART_AVG_23",
    "queryId": "CTAS_PART_AVG_23"
  },
  "commandSequenceNumber": 24,
  "warnings": []
}
```