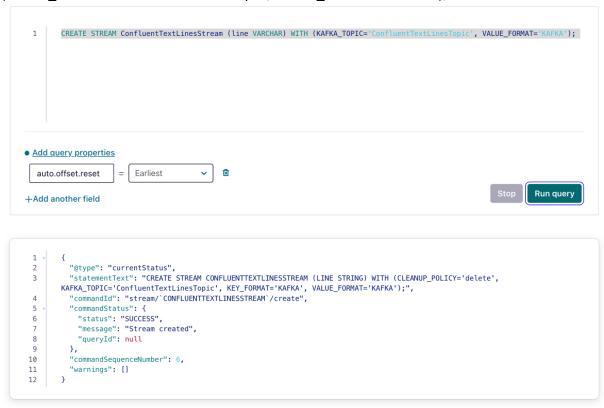
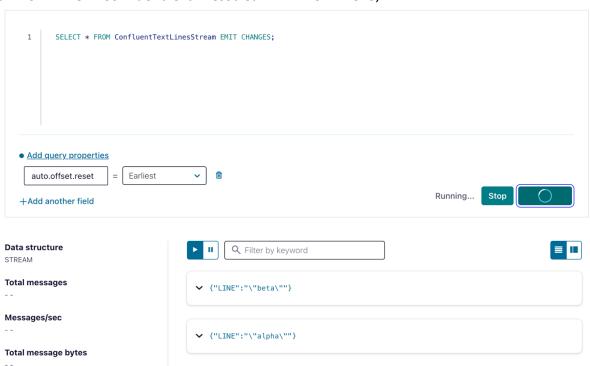
Lab 9

4. Run the commands inside KSQL

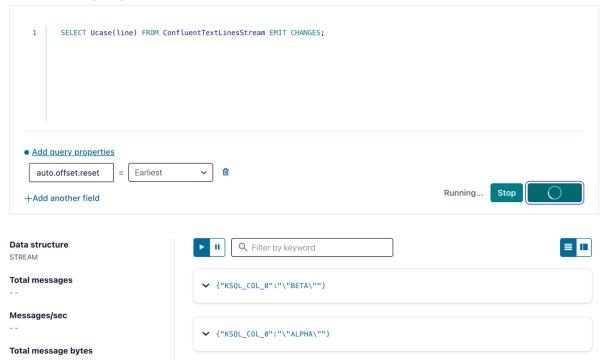
CREATE STREAM ConfluentTextLinesStream (line VARCHAR) WITH (KAFKA_TOPIC='ConfluentTextLinesTopic', VALUE_FORMAT='KAFKA');



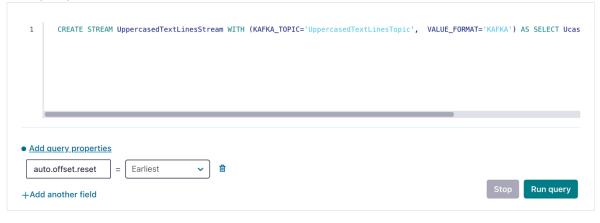
SELECT * FROM ConfluentTextLinesStream EMIT CHANGES;



SELECT Ucase(line) FROM ConfluentTextLinesStream EMIT CHANGES;

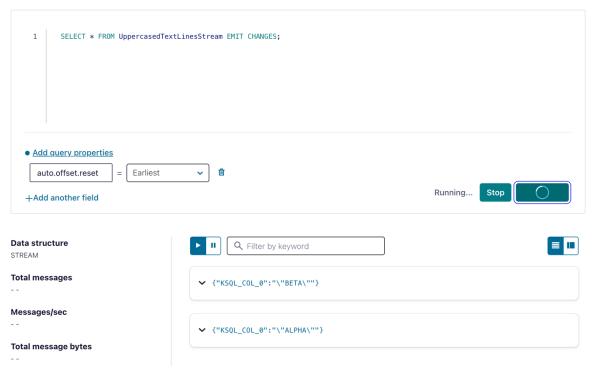


CREATE STREAM UppercasedTextLinesStream WITH (KAFKA_TOPIC='UppercasedTextLinesTopic', VALUE_FORMAT='KAFKA') AS SELECT Ucase(line) FROM ConfluentTextLinesStream;



Check UppercasedTextLinesTopic topic;

SELECT * FROM UppercasedTextLinesStream EMIT CHANGES;



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

```
1 CREATE STREAM readings (
2 sensor VARCHAR KEY,
3 val DOUBLE,
4 location VARCHAR) WITH (
5 kafka_topic='readings',
6 partitions=6,
7 value_format='JSON');

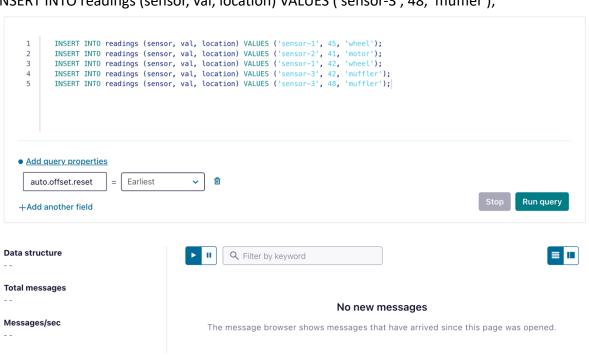
• Add query properties

auto.offset.reset = Earliest 

+Add another field

Stop Run query
```

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');



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;

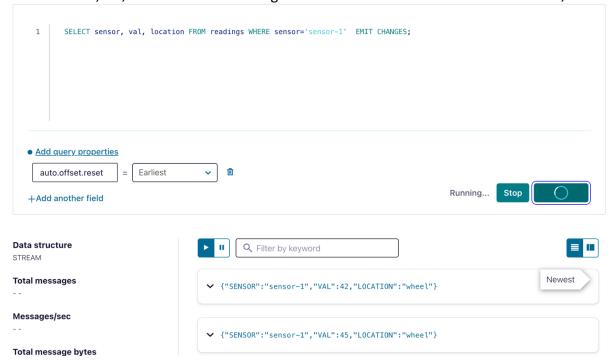
Add query properties

auto.offset.reset = Earliest 

HAdd another field

Stop Run query
```

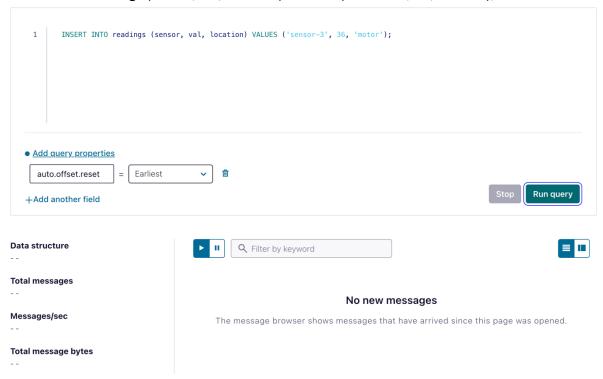
SELECT sensor, val, location FROM readings WHERE sensor='sensor-1' EMIT CHANGES;



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



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



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

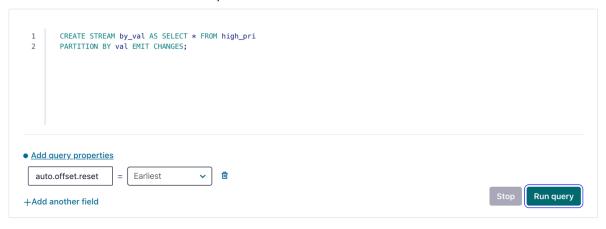
auto.offset.reset = Earliest  

HAdd another field
```

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



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



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');



CREATE TABLE avg_readings AS

SELECT sensor, AVG(val) as avg

FROM readings

GROUP BY sensor

EMIT CHANGES;



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

