LAB9_asg1_3
1)

2)

4)

Cluster
cluster_0

- Cluster Overview
  - Networking
  - API Keys
  - Cluster Settings
- Stream Lineage
- Stream Designer
- Topics
- **ksqlDB**
- Connectors
- Clients

Editor   Flow   Streams   Tables   Persistent queries   Performance   Settings   CLI instructions

```
1    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\""}

Cluster
cluster_0

- Cluster Overview
  - Networking
  - API Keys
  - Cluster Settings
- Stream Lineage
- Stream Designer
- Topics
- **ksqlDB**
- Connectors
- Clients

Editor   Flow   Streams   Tables   Persistent queries   Performance   Settings   CLI instructions

```
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\""}

Cluster
cluster_0

Editor | Flow | Streams | Tables | Persistent queries | Performance | Settings | CLI instructions

```
1    CREATE STREAM UppercasedTextLinesStream WITH (KAFKA_TOPIC='UppercasedTextLinesTopic',  VALUE_FORMAT='KAFKA
```

● 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_9",
8      "queryId": "CSAS_UPPERCASEDTEXTLINESSTREAM_9"
9    },
10   "commandSequenceNumber": 10,
11   "warnings": []
12 }
```

Cluster
cluster_0

ksqlDB_cluster_0

Editor | Flow | Streams | Tables | Persistent queries | Performance | Settings | CLI instructions

```
1    SELECT * FROM UppercasedTextLinesStream EMIT CHANGES;
2
```

● Add query properties

auto.offset.reset = Earliest 🗑

+ Add another field

Running... | Stop | ◯

Data structure
STREAM

Total messages
- -

Messages/sec
- -

Total message bytes
- -

Message fields

🔍 Filter by keyword

∨ {"KSQL_COL_0":"\"BETA\""}

∨ {"KSQL_COL_0":"\"ALPHA\""}

LAB9_asg1_4

1)

cluster_0

- Cluster Overview
  - Networking
  - API Keys
  - Cluster Settings
- Stream Lineage
- Stream Designer
- Topics
- ksqlDB
- Connectors
- Clients

Topics

Store more data for less with 20% lower storage prices
Increase retention of your topics to unlock more streaming use cases with historical context, gain deeper insights, and elevate customer experiences. Read more

Topic name	Partitions	Production (last min)	Consumption (last min)	Schema
ConfluentTextLinesTopic	10	0B/s	0B/s	Set a schema
pksqlc-x2xmq-processing-log	8	--	--	Set a schema
pksqlc-x2xmqUSERS_FILTERED	1	0B/s	--	Set a schema
readings	6	--	--	Set a schema
UppercasedTextLinesTopic	10	0B/s	0B/s	Set a schema
users	1	0B/s	0B/s	Set a schema

2)

cluster_0

Editor   Flow   Streams   Tables   Persistent queries   Performance   Settings   CLI instructions

- Cluster Overview
  - Networking
  - API Keys
  - Cluster Settings
- Stream Lineage
- Stream Designer
- Topics
- ksqlDB
- Connectors
- Clients
- Schema Registry
- CLI and Tools

```
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');
8
```

● Add query properties

auto.offset.reset = Earliest 🗑

+Add another field

Stop   Run query

```json
1  {
2      "@type": "currentStatus",
3      "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');",
4      "commandId": "stream/`READINGS`/create",
5      "commandStatus": {
6          "status": "SUCCESS",
7          "message": "Stream created",
8          "queryId": null
9      },
10     "commandSequenceNumber": 12,
11     "warnings": []
12  }
```

4)

```
1    CREATE STREAM clean AS SELECT sensor, val, UCASE(location) as location FROM readings EMIT CHANGES;
2
```

● Add query properties

auto.offset.reset = Earliest ▾ 🗑

+Add another field

Stop   Run query

```
1    {
2      "@type": "currentStatus",
3      "statementText": "CREATE STREAM CLEAN WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-
     x2xmqCLEAN', 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;",
4      "commandId": "stream/`CLEAN`/create",
5      "commandStatus": {
6        "status": "SUCCESS",
7        "message": "Created query with ID CSAS_CLEAN_13",
8        "queryId": "CSAS_CLEAN_13"
9      },
10     "commandSequenceNumber": 14,
11     "warnings": []
12   }
```

```
1   SELECT sensor, val, location FROM readings WHERE sensor='sensor-1'  EMIT CHANGES;
2
```

- 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   ☰ ▥

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

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

St CONFL
St HIGH_I
St KSQL_
St READIN
St UPPER
St USERS
St USERS

```
1   CREATE STREAM high_readings AS SELECT sensor, val, location FROM clean where val > 42 EMIT CHANGES;
2
```

- Add query properties

| auto.offset.reset | = | Earliest ▾ | 🗑 |

+Add another field

Stop   Run query

```
1    {
2      "@type": "currentStatus",
3      "statementText": "CREATE STREAM HIGH_READINGS WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-
     x2xmqHIGH_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;",
4      "commandId": "stream/`HIGH_READINGS`/create",
5      "commandStatus": {
6        "status": "SUCCESS",
7        "message": "Created query with ID CSAS_HIGH_READINGS_15",
8        "queryId": "CSAS_HIGH_READINGS_15"
9      },
10     "commandSequenceNumber": 16,
11     "warnings": []
12   }
```

6)

cluster_0

- Cluster Overview
  - **Networking**
  - **API Keys**
  - **Cluster Settings**
- Stream Lineage
- Stream Designer
- Topics
- ksqlDB
- Connectors
- Clients

- Schema Registry
- CLI and Tools

```sql
1   CREATE STREAM high_pri AS SELECT sensor, val, UCASE(location) as location
2   FROM readings where val > 42 EMIT CHANGES;
3
```

● Add query properties

auto.offset.reset = [Earliest ▾] 🗑

⊕ Add another field

[Stop] [Run query]

```json
1    {
2      "@type": "currentStatus",
3      "statementText": "CREATE STREAM HIGH_PRI WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-
       x2xmqHIGH_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;",
4      "commandId": "stream/`HIGH_PRI`/create",
5      "commandStatus": {
6        "status": "SUCCESS",
7        "message": "Created query with ID CSAS_HIGH_PRI_17",
8        "queryId": "CSAS_HIGH_PRI_17"
9      },
10     "commandSequenceNumber": 18,
11     "warnings": []
12   }
```

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

● Add query properties

| auto.offset.reset | = | Earliest ▾ | 🗑 |

+Add another field

Stop   **Run query**

```
1    {
2      "@type": "currentStatus",
3      "statementText": "CREATE STREAM BY_LOCATION WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-
       x2xmqBY_LOCATION', PARTITIONS=6, REPLICAS=3, RETENTION_MS=604800000) AS SELECT *\nFROM HIGH_PRI
       HIGH_PRI\nPARTITION BY HIGH_PRI.LOCATION\nEMIT CHANGES;",
4      "commandId": "stream/`BY_LOCATION`/create",
5      "commandStatus": {
6        "status": "SUCCESS",
7        "message": "Created query with ID CSAS_BY_LOCATION_19",
8        "queryId": "CSAS_BY_LOCATION_19"
9      },
10     "commandSequenceNumber": 20,
11     "warnings": []
12   }
```

cluster_0

- Cluster Overview
  - Networking
  - API Keys
  - Cluster Settings
- Stream Lineage
- Stream Designer
- Topics
- ksqlDB
- Connectors
- Clients

- Schema Registry
- CLI and Tools

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

● Add query properties

| auto.offset.reset | = | Earliest ⌄ | 🗑 |

Stop   Run query

+Add another field

```
1   {
2     "@type": "currentStatus",
3     "statementText": "CREATE STREAM BY_VAL WITH (CLEANUP_POLICY='delete', KAFKA_TOPIC='pksqlc-
      x2xmqBY_VAL', PARTITIONS=6, REPLICAS=3, RETENTION_MS=604800000) AS SELECT *\nFROM HIGH_PRI
      HIGH_PRI\nPARTITION BY HIGH_PRI.VAL\nEMIT CHANGES;",
4     "commandId": "stream/`BY_VAL`/create",
5     "commandStatus": {
6       "status": "SUCCESS",
7       "message": "Created query with ID CSAS_BY_VAL_21",
8       "queryId": "CSAS_BY_VAL_21"
9     },
10    "commandSequenceNumber": 22,
11    "warnings": []
12  }
```

LAB9_asg1_5
3)

cluster_0

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

● Add query properties

[auto.offset.reset]   =   [Earliest ▾]   🗑

＋Add another field

[Stop]   [**Run query**]

```
 1    {
 2      "@type": "currentStatus",
 3      "statementText": "CREATE TABLE AVG_READINGS WITH (CLEANUP_POLICY='compact', KAFKA_TOPIC='pksqlc-
      x2xmqAVG_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;",
 4      "commandId": "table/`AVG_READINGS`/create",
 5      "commandStatus": {
 6        "status": "SUCCESS",
 7        "message": "Created query with ID CTAS_AVG_READINGS_23",
 8        "queryId": "CTAS_AVG_READINGS_23"
 9      },
10      "commandSequenceNumber": 24,
11      "warnings": []
12    }
```

cluster_0

- **Cluster Overview**
  - Networking
  - API Keys
  - Cluster Settings
- Stream Lineage
- Stream Designer
- Topics
- **ksqlDB**
- Connectors
- Clients

- Schema Registry
- CLI and Tools

```sql
1   CREATE TABLE part_avg AS
2       SELECT location, AVG(val) as avg
3       FROM readings
4       GROUP BY location
5       EMIT CHANGES;
```

● **Add query properties**

[auto.offset.reset] = [Earliest ▾] 🗑

+**Add another field**

[Stop] [**Run query**]

```json
1   {
2     "@type": "currentStatus",
3     "statementText": "CREATE TABLE PART_AVG WITH (CLEANUP_POLICY='compact', KAFKA_TOPIC='pksqlc-
      x2xmqPART_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;",
4     "commandId": "table/`PART_AVG`/create",
5     "commandStatus": {
6       "status": "SUCCESS",
7       "message": "Created query with ID CTAS_PART_AVG_25",
8       "queryId": "CTAS_PART_AVG_25"
9     },
10    "commandSequenceNumber": 26,
11    "warnings": []
12  }
```