UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
DIRECCIÓN DE POSGRADO

DIPLOMADO ESTADÍSTICA APLICADA A LA TOMA DE DECISIONES

SEGUNDA VERSIÓN

LABORATORIO KSQL

NOMBRE : HEBERT JUAN DE DIOS DELGADILLO FERNANDEZ
CARLOS ALFREDO ORIHUELA BERRIOS

DOCENTE : DANNY LUIS HUANCA SEVILLA

Cochabamba – Bolivia
2023

Antecedente

Subir el contenido del compreso (luego de descomprimir): código_auxiliar a la dirección de la MV /home/curso/cursoConfluent

Debe quedar así:

```
(base) curso@cursobigdata:~$ mkdir cursoConfluent
(base) curso@cursobigdata:~$ ls
 Downloads          datos              iniciarHadoop       iniciarLabA                                    stage
'Untitled Folder'   derby.log          iniciarHive         iniciarLaboratorio
 anaconda3          hiveserver2log     iniciarJupyter      pararLaboratorio
 cursoConfluent     iniciarConfluent   iniciarKSQLserver   spark-streaming-kafka-0-8-assembly_2.11-2.4.7.jar
(base) curso@cursobigdata:~$
```

```
(base) curso@cursobigdata:~$ cd cursoConfluent/
(base) curso@cursobigdata:~/cursoConfluent$ ls
perfilusuario.avro  queryConformato.ksql
(base) curso@cursobigdata:~/cursoConfluent$
```

1. iniciar la plataforma confluent, que contiene los servidores de Kafka y de Zookeeper instalados

a) Para iniciar la plataforma ejecute el siguiente comando en una conexión ssh.

iniciarConfluent

Se debe tener lo siguiente levantado:

```
(base) curso@cursobigdata:~$ iniciarConfluent
Updates are available for confluent from (current: v0.239.0, latest: v1.22.0). To install them, please run
$ confluent update

    The local commands are intended for a single-node development environment
    only, NOT for production usage. https://docs.confluent.io/current/cli/index.html

Using CONFLUENT_CURRENT: /tmp/confluent.7iZDqJPK
Starting zookeeper
zookeeper is [UP]
Starting kafka
kafka is [UP]
Starting schema-registry
schema-registry is [UP]
Starting kafka-rest
kafka-rest is [UP]
Starting connect
connect is [UP]
Starting ksql-server
ksql-server is [UP]
Starting control-center
control-center is [UP]
    The local commands are intended for a single-node development environment
    only, NOT for production usage. https://docs.confluent.io/current/cli/index.html
```

Para parar todos los servicios (mejor de esta manera pues puede crear conflicto con otras aplicaciones). Ejecutar el script:

pararLaboratorio

b) Iniciar el cliente para interactuar con la plataforma.

ksql;

```
(base) curso@cursobigdata:~$ ksql

                  ===========================================
                  =        _  __  _____   ____   _         =
                  =       | |/ // ____|/ __ \| |          =
                  =       | ' /| (___ | |  | | |          =
                  =       |  <  \___ \| |  | | |          =
                  =       | . \ ____) | |__| | |____      =
                  =       |_|\_\_____/ \____/|______|     =
                  =                                       =
                  =  Streaming SQL Engine for Apache Kafka® =
                  ===========================================

Copyright 2017-2019 Confluent Inc.

CLI v5.4.0, Server v5.4.0 located at http://localhost:8088

Having trouble? Type 'help' (case-insensitive) for a rundown of how things work!

ksql>
```

c) Liste los tópicos que se encuentran presentes:

list topics;

```
 _confluent-controlcenter-5-4-0-1-MonitoringTriggerStore-changelog          | 2         | 1
 _confluent-controlcenter-5-4-0-1-MonitoringVerifierStore-changelog         | 2         | 1
 _confluent-controlcenter-5-4-0-1-TriggerActionsStore-changelog             | 2         | 1
 _confluent-controlcenter-5-4-0-1-TriggerEventsStore-changelog              | 2         | 1
 _confluent-license                                                         | 1         | 1
 _confluent-metrics                                                         | 12        | 1
 _confluent-monitoring                                                      | 2         | 1
 _schemas                                                                   | 1         | 1
 connect-configs                                                            | 1         | 1
 connect-offsets                                                            | 25        | 1
 connect-statuses                                                           | 5         | 1
 default_ksql_processing_log                                                | 1         | 1
 etiquetas                                                                  | 2         | 1
 mensajes                                                                   | 5         | 1
---------------------------------------------------------------------------------------------
ksql>
```

Crear un tópico denominado USUARIOS mediante la siguiente sentencia en línea de comandos:

kafka-topics --zookeeper localhost:2181 --create --partitions 1 --replication-factor 1 --topic USUARIOS;

```
(base) curso@cursobigdata:~$ kafka-topics --zookeeper localhost:2181 --create --partitions 1 --replication-factor 1 --topic US
UARIOS;
Created topic USUARIOS.
(base) curso@cursobigdata:~$
```

MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

d) Ejecute el listado de tópicos de nuevo.

list topics;

```
 _confluent-license                                                  | 1          | 1
 _confluent-metrics                                                  | 12         | 1
 _confluent-monitoring                                               | 2          | 1
 _schemas                                                            | 1          | 1
 connect-configs                                                     | 1          | 1
 connect-offsets                                                     | 25         | 1
 connect-statuses                                                    | 5          | 1
 default_ksql_processing_log                                         | 1          | 1
 etiquetas                                                           | 2          | 1
 mensajes                                                            | 5          | 1
 USUARIOS                                                            | 1          | 1
---------------------------------------------------------------------------------------
ksql>
```

e) Crear un productor para el tópico USUARIOS

kafka-console-producer --broker-list localhost:9092 --topic USUARIOS

```
(base) curso@cursobigdata:~$ kafka-console-producer --broker-list localhost:9092 --topic USUARIOS
>
```

d) En el cliente ksql imprimir el topico USUARIOS con la finalidad de utilizarlo como consumidor del tópico USUARIOS.

print 'USUARIOS';

```
ksql> print 'USUARIOS';

Press CTRL-C to interrupt
```

MobaXterm by subscribing to the professional edition here:

Envíe pares de usuarios y sus países en el formato: “nombre, abreviación país”.

```
USUARIOS

------------------------------------------------------------
ksql> print 'USUARIOS';
Format:STRING
9/7/23 8:36:02 AM UTC , NULL , Remmy,bo
9/7/23 8:36:14 AM UTC , NULL , Beto,pa
9/7/23 8:36:22 AM UTC , NULL , Ramiro,bra
9/7/23 8:36:44 AM UTC , NULL , Pedro,pa
```

☑ Exclude "192.168.206.128 (curso)" from MultiExec mode

```
                                        host:port. Multip
                                        given to allow fa
(base) curso@cursobigdata:~$ kafka-topics --zookeeper localh
UARIOS;
Created topic USUARIOS.
(base) curso@cursobigdata:~$ kafka-console-producer --broker
>Remmy,bo
>Beto,pa
>Ramiro,bra
>Pedro,pa
>
```

☐ Exclude "192.168.206.128 (curso)" from MultiExec mode

e) Para observar toda la data desde el inicio del tópico se puede usar un complemento del comando print. Con el complemento from beginning.

```
------------------------------------------------------------
ksql> print 'USUARIOS';
Format:STRING
9/7/23 8:36:02 AM UTC , NULL , Remmy,bo
9/7/23 8:36:14 AM UTC , NULL , Beto,pa
Press CTRL-C to interrupt


9/7/23 8:45:05 AM UTC , NULL , danny,bo
```

☑ Exclude "192.168.206.128 (curso)" from MultiExec mode

```
UARIOS;
Created topic USUARIOS.
(base) curso@cursobigdata:~$ kafka-console-producer --broker
>Remmy,bo
>Beto,pa
>Ramiro,bra
>Pedro,pa
>danny,bo
>
```

☐ Exclude "192.168.206.128 (curso)" from MultiExec mode

print 'USUARIOS' from beginning;

```
ksql> print 'USUARIOS' from beginning;
Format:STRING
9/7/23 8:36:02 AM UTC , NULL , Remmy,bo
9/7/23 8:36:14 AM UTC , NULL , Beto,pa
9/7/23 8:36:22 AM UTC , NULL , Ramiro,bra
9/7/23 8:36:44 AM UTC , NULL , Pedro,pa
9/7/23 8:45:05 AM UTC , NULL , danny,bo

Press CTRL-C to interrupt
```

f) Se puede limitar el contenido mediante el complemento: limit

print 'USUARIOS' from beginning limit 3;

```
ksql> print 'USUARIOS' from beginning limit 3;
Format:STRING
9/7/23 8:36:02 AM UTC , NULL , Remmy,bo
9/7/23 8:36:14 AM UTC , NULL , Beto,pa
9/7/23 8:36:22 AM UTC , NULL , Ramiro,bra
ksql>
```

g) Por otra parte se puede limitar la muestra a intervalos. Use el complemento: interval.

print 'USUARIOS' from beginning interval 3 limit 6;

```
ksql> print 'USUARIOS' from beginning interval 3 limit 6;
Format:STRING
9/7/23 8:36:02 AM UTC , NULL , Remmy,bo
9/7/23 8:36:44 AM UTC , NULL , Pedro,pa

Press CTRL-C to interrupt
```

2. Crear un stream KSQL tipo CSV

a) Crear un stream denominado stream_usuarios para consumir desde el tópico **USUARIOS**.

create stream stream_usuarios (nombre varchar, codigo_pais varchar) WITH (KAFKA_TOPIC='USUARIOS', VALUE_FORMAT='DELIMITED');

Debe obtener el siguiente mensaje:

```
ksql> create stream stream_usuarios (nombre varchar, codigo_pais varchar) WITH (KAFKA_TOPIC='USUARIOS', VALUE_FORMAT='DELIMITE
D');

 Message
----------------
 Stream created
----------------
ksql>
```

b) Listar el stream creado mediante la sentencia list streams;

list streams;

```
ksql> list streams;

 Stream Name         | Kafka Topic                 | Format
-----------------------------------------------------------------
 KSQL_PROCESSING_LOG | default_ksql_processing_log | JSON
 STREAM_USUARIOS     | USUARIOS                    | DELIMITED
-----------------------------------------------------------------
ksql>
```

c) Realizar una consulta KSQL al stream creado.

select nombre, codigo_pais from stream_usuarios emit changes;

Deberiamos obtener algo asi:

```
ksql> select nombre, codigo_pais from stream_usuarios emit changes;

Press CTRL-C to interrupt
```

No podrá obtener valores ya que se deben enviar valores nuevos. Escriba algunos nombres y ciudades en el productor. ¿Aparecen los valores?

```
|NOMBRE
  |CODIGO_PAIS
    |
+-----------------------------------------------------------
--+-----------------------------------------------------------
----+
|maria                    |br                       |
|juana                    |pe                       |
```

Exclude "192.168.206.128 (curso)" from MultiExec mode

```
(base) curso@cursobigdata:~$ kafka-console-producer --broker
>Remmy,bo
>Beto,pa
>Ramiro,bra
>Pedro,pa
>danny,bo
>maria,br
>juana,pe
>
```

Exclude "192.168.206.128 (curso)" from MultiExec mode

d) Se requiere leer desde el inicio.

Para tal efecto se puede cambiar una configuración de la sesión:

set 'auto.offset.reset'='earliest';

```
^CQuery terminated
ksql> sfully changed local property 'auto.offset.reset' to 'earliest'. Use the UNSET command to revert your change.
```

Esta configuración solo será en la sesión

Realice el query select de nuevo, puede observar el cambio?

```
ksql> select nombre, codigo_pais from stream_usuarios emit changes;
+-----------------------------------------------------------+-----------------------------------------------------------+
|NOMBRE                                                     |CODIGO_PAIS                                                |
+-----------------------------------------------------------+-----------------------------------------------------------+
|Remmy                                                      |bo                                                         |
|Beto                                                       |pa                                                         |
|Ramiro                                                     |bra                                                        |
|Pedro                                                      |pa                                                         |
|danny                                                      |bo                                                         |
|maria                                                      |br                                                         |
|juana                                                      |pe                                                         |

Press CTRL-C to interrupt
```

e) limitar la sentencia a 5 registros. Continua el flujo o termina?

select nombre, codigo_pais from stream_usuarios emit changes limit 5;

```
ksql> select nombre, codigo_pais from stream_usuarios emit changes limit 5;
+-----------------------------------------------------------+-----------------------------------------------------------+
|NOMBRE                                                     |CODIGO_PAIS                                                |
+-----------------------------------------------------------+-----------------------------------------------------------+
|Remmy                                                      |bo                                                         |
|Beto                                                       |pa                                                         |
|Ramiro                                                     |bra                                                        |
|Pedro                                                      |pa                                                         |
|danny                                                      |bo                                                         |
Limit Reached
Query terminated
ksql>
```

f) Se desea agrupar los datos por país, cuántas personas existen por país?

select codigo_pais, count(*) from stream_usuarios group by codigo_pais emit changes;

```
ksql> select codigo_pais, count(*) from stream_usuarios group by codigo_pais emit changes;
+-----------------------------------------------------------+-----------------------------------------------------------+
|CODIGO_PAIS                                                |KSQL_COL_1                                                 |
+-----------------------------------------------------------+-----------------------------------------------------------+
|bra                                                        |1                                                          |
|pa                                                         |2                                                          |
|bo                                                         |2                                                          |
|br                                                         |1                                                          |
|pe                                                         |1                                                          |

Press CTRL-C to interrupt
```

baXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

g) Para borrar el stream se ejecuta, de la misma manera como si fuera una tabla SQL:

drop stream stream_usuarios;

```
ksql> drop stream stream_usuarios;

 Message
----------------------------------------------------------------
 Source `STREAM_USUARIOS` (topic: USUARIOS) was dropped.
----------------------------------------------------------------
ksql>
```

En caso se requiera verificar su existencia además de borrar el tópico asociado se tiene:

drop stream if exists stream_usuarios delete topic;

Debe obtener:

```
ksql> drop stream if exists stream_usuarios delete topic;
Could not find source to delete topic for: ConfiguredStatement{statement=drop stream if exists stream_usuarios delete topic;,
overrides={auto.offset.reset=earliest}, config=io.confluent.ksql.util.KsqlConfig@88402c5a}
ksql>
```

h) ¿Se borró la información realmente? Mostrar los streams, mostrar los tópicos

El stream si fue borrado, antes existían 2, uno era nuestro stream_usuarios que ya no está.

show streams;

```
ksql> show streams;

 Stream Name         | Kafka Topic                 | Format
-----------------------------------------------------------
 KSQL_PROCESSING_LOG | default_ksql_processing_log | JSON
-----------------------------------------------------------
ksql>
```

show topics;

```
 connect-configs                                                    | 1          | 1
 connect-offsets                                                    | 25         | 1
 connect-statuses                                                   | 5          | 1
 default_ksql_processing_log                                        | 1          | 1
 etiquetas                                                          | 2          | 1
 mensajes                                                           | 5          | 1
 USUARIOS                                                           | 1          | 1
-----------------------------------------------------------------------------------------
ksql>
```

3. Crear Streams JSON

a) En el Shell. Crear un tópico perfil de usuario de tipo JSON.

kafka-topics --zookeeper localhost:2181 --create --partitions 1 --replication-factor 1 --topic PERFILUSUARIO

```
(base) curso@cursobigdata:~$ kafka-topics --zookeeper localhost:2181 --create --partitions 1 --replication-factor 1 --topic PE
RFILUSUARIO
Created topic PERFILUSUARIO.
(base) curso@cursobigdata:~$
```

b) En el Shell. Crear un productor asociado al tópico PERFILUSUARIO e insertar dos registros JSON.

kafka-console-producer --broker-list localhost:9092 --topic PERFILUSUARIO

<<EOF

{"id_usuario": 1000, "nombre": "Jaime", "apellido": "Perez", "codigo_pais": "bo", "calificacion": 5.2}

EOF

kafka-console-producer --broker-list localhost:9092 --topic PERFILUSUARIO

<<EOF

{"id_usuario": 1001, "nombre":"Jimena", "apellido":"Vargas", "codigo_pais":"bo", "calificacion":7.2 }

EOF

```
(base) curso@cursobigdata:~$ kafka-console-producer --broker-list localhost:9092 --topic PERFILUSUARIO<<EOF
>
> {"id_usuario": 1000,    "nombre": "Jaime",    "apellido": "Perez",    "codigo_pais": "bo",    "calificacion": 5.2}
>
> EOF
>>>>(base) curso@cursobigdata:~$ kafka-console-producer --broker-list localhost:9092 --topic PERFILUSUARIO<<EOF
>
> {"id_usuario": 1001, "nombre":"Jimena",   "apellido":"Vargas", "codigo_pais":"bo",    "calificacion":7.2 }
> EOF
>>>(base) curso@cursobigdata:~$
```

También se puede enviar los Json sin necesidad de especificar el EOF (End of File)

{"id_usuario": 1003, "nombre":"Dayana", "apellido":"Perez", "codigo_pais":"bo", "calificacion":9.2 }

{"id_usuario": 1004, "nombre":"Rita", "apellido":"Gomez", "codigo_pais":"pe", "calificacion":10.2 }

```
>>>(base) curso@cursobigdata:~$ kafka-console-producer --broker-list localhost:9092 --topic PERFILUSUARIO
>{"id_usuario": 1003, "nombre":"Dayana",   "apellido":"Perez", "codigo_pais":"bo",    "calificacion":9.2 }
>{"id_usuario": 1004, "nombre":"Rita",   "apellido":"Gomez", "codigo_pais":"pe",    "calificacion":10.2 }
>
```

```
ksql> print 'PERFILUSUARIO';
Format:JSON
{"ROWTIME":1694078521574,"ROWKEY":"null","id_usuario":1000,"nombre":"Jaime","apellido":"Perez","codigo_pais":"bo","calificacio
n":5.2}
{"ROWTIME":1694078544421,"ROWKEY":"null","id_usuario":1001,"nombre":"Jimena","apellido":"Vargas","codigo_pais":"bo","calificac
ion":7.2}
{"ROWTIME":1694079003934,"ROWKEY":"null","id_usuario":1003,"nombre":"Dayana","apellido":"Perez","codigo_pais":"bo","calificaci
on":9.2}
{"ROWTIME":1694079017838,"ROWKEY":"null","id_usuario":1004,"nombre":"Rita","apellido":"Gomez","codigo_pais":"pe","calificacion
":10.2}

Press CTRL-C to interrupt
```

c) En el cliente ksql. Crear un flujo, un stream denominado **usuario** que consuma datos del tópico PERFILUSUARIO. En este punto se está adicionando el schema sobre el tópico PERFILUSUARIO.

CREATE STREAM usuario (id_usuario INT, nombre VARCHAR, apellido VARCHAR, codigo_pais VARCHAR, calificación DOUBLE) WITH (VALUE_FORMAT = 'JSON', KAFKA_TOPIC = 'PERFILUSUARIO');

```
ksql> CREATE STREAM usuario (id_usuario INT, nombre VARCHAR, apellido VARCHAR, codigo_pais VARCHAR, calificacion DOUBLE) WITH
(VALUE_FORMAT = 'JSON', KAFKA_TOPIC = 'PERFILUSUARIO');

 Message
----------------
 Stream created
----------------
ksql>
```

d) En el cliente ksql. Exponer los datos del stream mediante la sentencia **select** y que se pueda observar los 2 datos introducidos (desde el inicio)

SET 'auto.offset.reset'='earliest';

```
ksql> SET 'auto.offset.reset'='earliest';
Successfully changed local property 'auto.offset.reset' from 'earliest' to 'earliest'.
ksql>
```

select nombre, apellido, codigo_pais, calificacion from usuario emit changes;

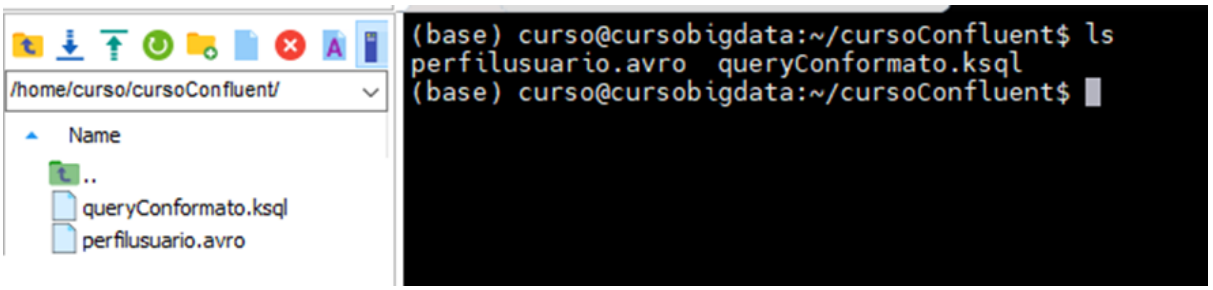
```
ksql> select nombre, apellido, codigo_pais, calificacion from usuario emit changes;
+----------------------------+----------------------------+----------------------------+----------------------------+
|NOMBRE                      |APELLIDO                    |CODIGO_PAIS                 |CALIFICACION                |
+----------------------------+----------------------------+----------------------------+----------------------------+
|Jaime                       |Perez                       |bo                          |5.2                         |
|Jimena                      |Vargas                      |bo                          |7.2                         |
|Dayana                      |Perez                       |bo                          |9.2                         |
|Rita                        |Gomez                       |pe                          |10.2                        |

Press CTRL-C to interrupt
```

4. Crear datos simulados

En KSQL se pueden crear datos simulados mediante KSQL Datagen, que genera datos aleatorios en distintos formatos: CSV, JSON, AVRO.

a) Subir el archivo /código_auxiliar/perfilusuario.avro a la carpeta: /home/curso/cursoConfluent



b) Dirigirse al **home**, que es el lugar en el que se está corriendo los scripts. Ejecutar:

ksql-datagen schema=./cursoConfluent/perfilusuario.avro format=json topic=PERFILUSUARIO key=id_usuario maxInterval=5000 iterations=100

El resultado similar a:

```
        ksql.logging.processing.topic.partitions = 1
        ksql.logging.processing.topic.replication.factor = 1
 (io.confluent.ksql.logging.processing.ProcessingLogConfig:347)
1000 --> ([ '1000' | 'Danny' | 'Vargas' | 'US' | '9.0' ]) ts:1694079530264
1001 --> ([ '1001' | 'Maria' | 'Quispe' | 'MX' | '6.4' ]) ts:1694079530974
1002 --> ([ '1002' | 'Ana' | 'Quispe' | 'US' | '6.4' ]) ts:1694079533626
1003 --> ([ '1003' | 'Maria' | 'Jimenes' | 'BO' | '4.8' ]) ts:1694079537272
1004 --> ([ '1004' | 'Julio' | 'Ballivian' | 'US' | '5.2' ]) ts:1694079540584
1005 --> ([ '1005' | 'Pedro' | 'Jimenes' | 'MX' | '6.5' ]) ts:1694079542413
1006 --> ([ '1006' | 'Juan' | 'Ballivian' | 'MX' | '9.0' ]) ts:1694079542536
1007 --> ([ '1007' | 'Danny' | 'Quispe' | 'BO' | '5.2' ]) ts:1694079543225
1008 --> ([ '1008' | 'Luis' | 'Salazar' | 'BO' | '9.0' ]) ts:1694079547781
1009 --> ([ '1009' | 'Luis' | 'Montero' | 'AR' | '4.7' ]) ts:1694079550068
1010 --> ([ '1010' | 'Maria' | 'Ballivian' | 'US' | '6.4' ]) ts:1694079553011
1011 --> ([ '1011' | 'Danny' | 'Montero' | 'BO' | '9.0' ]) ts:1694079556448
1012 --> ([ '1012' | 'Pedro' | 'Peres' | 'US' | '4.8' ]) ts:1694079557249
1013 --> ([ '1013' | 'Luis' | 'Ballivian' | 'US' | '9.0' ]) ts:1694079561203
1014 --> ([ '1014' | 'Julio' | 'Quispe' | 'AR' | '9.0' ]) ts:1694079565254
1015 --> ([ '1015' | 'Julio' | 'Salazar' | 'AR' | '9.0' ]) ts:1694079565931
1016 --> ([ '1016' | 'Ramon' | 'Vargas' | 'MX' | '6.5' ]) ts:1694079570412
1017 --> ([ '1017' | 'Luis' | 'Salazar' | 'BO' | '4.7' ]) ts:1694079573967
1018 --> ([ '1018' | 'Danny' | 'Montero' | 'AR' | '6.4' ]) ts:1694079575083
1019 --> ([ '1019' | 'Danny' | 'Salazar' | 'BO' | '4.8' ]) ts:1694079577355
1020 --> ([ '1020' | 'Fernando' | 'Quispe' | 'AR' | '6.4' ]) ts:1694079580833
1021 --> ([ '1021' | 'Ana' | 'Peres' | 'MX' | '9.0' ]) ts:1694079581352
1022 --> ([ '1022' | 'Luis' | 'Peres' | 'US' | '4.7' ]) ts:1694079584468
1023 --> ([ '1023' | 'Juan' | 'Montero' | 'US' | '6.5' ]) ts:1694079587487
1024 --> ([ '1024' | 'Ramon' | 'Quispe' | 'MX' | '9.0' ]) ts:1694079591712
1025 --> ([ '1025' | 'Juan' | 'Peres' | 'MX' | '5.2' ]) ts:1694079594958
1026 --> ([ '1026' | 'Julio' | 'Ballivian' | 'BO' | '5.2' ]) ts:1694079595496
1027 --> ([ '1027' | 'Julio' | 'Peres' | 'BO' | '6.5' ]) ts:1694079599130
1028 --> ([ '1028' | 'Juan' | 'Peres' | 'AR' | '4.7' ]) ts:1694079600962
1029 --> ([ '1029' | 'Luis' | 'Quispe' | 'MX' | '6.5' ]) ts:1694079602619
```

c) En el cliente ksql. Se puede imprimir el tópico mediante la sentencia

```
ksql> select nombre, apellido, codigo_pais, calificacion from usuario emit changes;
+----------------------------+----------------------------+----------------------------+----------------------------+
|NOMBRE                      |APELLIDO                    |CODIGO_PAIS                 |CALIFICACION                |
+----------------------------+----------------------------+----------------------------+----------------------------+
|Jaime                       |Perez                       |bo                          |5.2                         |
|Jimena                      |Vargas                      |bo                          |7.2                         |
|Dayana                      |Perez                       |bo                          |9.2                         |
|Rita                        |Gomez                       |pe                          |10.2                        |
|Danny                       |Vargas                      |US                          |9.0                         |
|Maria                       |Quispe                      |MX                          |6.4                         |
|Ana                         |Quispe                      |US                          |6.4                         |
|Maria                       |Jimenes                     |BO                          |4.8                         |
|Julio                       |Ballivian                   |US                          |5.2                         |
|Pedro                       |Jimenes                     |MX                          |6.5                         |
|Juan                        |Ballivian                   |MX                          |9.0                         |
|Danny                       |Quispe                      |BO                          |5.2                         |
|Luis                        |Salazar                     |BO                          |9.0                         |
|Luis                        |Montero                     |AR                          |4.7                         |
|Maria                       |Ballivian                   |US                          |6.4                         |
|Danny                       |Montero                     |BO                          |9.0                         |
|Pedro                       |Peres                       |US                          |4.8                         |
|Luis                        |Ballivian                   |US                          |9.0                         |
|Julio                       |Quispe                      |AR                          |9.0                         |
|Julio                       |Salazar                     |AR                          |9.0                         |
|Ramon                       |Vargas                      |MX                          |6.5                         |
|Luis                        |Salazar                     |BO                          |4.7                         |
|Danny                       |Montero                     |AR                          |6.4                         |
|Danny                       |Salazar                     |BO                          |4.8                         |
|Fernando                    |Quispe                      |AR                          |6.4                         |
|Ana                         |Peres                       |MX                          |9.0                         |
|Luis                        |Peres                       |US                          |4.7                         |
|Juan                        |Montero                     |US                          |6.5                         |
```

5) Uso de Streams

a) Descripción de la metadata de un stream. Ejecutar el siguiente comando:

describe USUARIO;

se debe obtener:

```
ksql> describe USUARIO;

Name                 : USUARIO
 Field        | Type
----------------------------------------
 ROWTIME      | BIGINT           (system)
 ROWKEY       | VARCHAR(STRING)  (system)
 ID_USUARIO   | INTEGER
 NOMBRE       | VARCHAR(STRING)
 APELLIDO     | VARCHAR(STRING)
 CODIGO_PAIS  | VARCHAR(STRING)
 CALIFICACION | DOUBLE
----------------------------------------
For runtime statistics and query details run: DESCRIBE EXTENDED <Stream,Table>;
ksql>
```

b) En el Shell. Ejecute el generador de datos en el punto 3.b

```
        ksql.logging.processing.topic.partitions = 1
        ksql.logging.processing.topic.replication.factor = 1
 (io.confluent.ksql.logging.processing.ProcessingLogConfig:347)
1000 --> ([ '1000' | 'Danny' | 'Peres' | 'AR' | '6.5' ]) ts:1693786536684
1001 --> ([ '1001' | 'Juan' | 'Ballivian' | 'US' | '9.0' ]) ts:1693786540710
1002 --> ([ '1002' | 'Fernando' | 'Ballivian' | 'BO' | '5.2' ]) ts:1693786545616
1003 --> ([ '1003' | 'Ramon' | 'Ballivian' | 'AR' | '6.4' ]) ts:1693786545769
1004 --> ([ '1004' | 'Pedro' | 'Salazar' | 'AR' | '6.5' ]) ts:1693786546536
```

c) En el cliente ksql. Ejecute la siguiente consulta:

select rowtime, nombre from usuario emit changes;

```
ksql> select rowtime, nombre from usuario emit changes;
+-----------------------------------------------------------+-----------------------------------
|ROWTIME                                                    |NOMBRE
+-----------------------------------------------------------+-----------------------------------
|1694078521574                                              |Jaime
|1694078544421                                              |Jimena
|1694079003934                                              |Dayana
|1694079017838                                              |Rita
|1694079530264                                              |Danny
|1694079530974                                              |Maria
|1694079533626                                              |Ana
|1694079537272                                              |Maria
|1694079540584                                              |Julio
|1694079542413                                              |Pedro
|1694079542536                                              |Juan
|1694079543225                                              |Danny
|1694079547781                                              |Luis
|1694079550068                                              |Luis
|1694079553011                                              |Maria
|1694079556448                                              |Danny
|1694079557249                                              |Pedro
|1694079561203                                              |Luis
|1694079565254                                              |Julio
|1694079565931                                              |Julio
|1694079570412                                              |Ramon
|1694079573967                                              |Luis
|1694079575083                                              |Danny
|1694079577355                                              |Danny
|1694079580833                                              |Fernando
|1694079581352                                              |Ana
|1694079584468                                              |Luis
|1694079587487                                              |Juan
```

d) En el cliente ksql. Se puede dar formato a las salidas existentes, por ejemplo la columna de tiempo rowtime.

SELECT TIMESTAMPTOSTRING(rowtime, 'dd/MMM HH:mm') as fechaCreacion, nombre

from usuario emit changes;

```
ksql> SELECT TIMESTAMPTOSTRING(rowtime, 'dd/MMM HH:mm') as fechaCreacion, nombre
>from usuario emit changes;
>
+-----------------------------------------------------------+---------------------------
|FECHACREACION                                              |NOMBRE
+-----------------------------------------------------------+---------------------------
|07/Sep 09:22                                               |Jaime
|07/Sep 09:22                                               |Jimena
|07/Sep 09:30                                               |Dayana
|07/Sep 09:30                                               |Rita
|07/Sep 09:38                                               |Danny
|07/Sep 09:38                                               |Maria
|07/Sep 09:38                                               |Ana
|07/Sep 09:38                                               |Maria
|07/Sep 09:39                                               |Julio
|07/Sep 09:39                                               |Pedro
|07/Sep 09:39                                               |Juan
|07/Sep 09:39                                               |Danny
|07/Sep 09:39                                               |Luis
|07/Sep 09:39                                               |Luis
|07/Sep 09:39                                               |Maria
|07/Sep 09:39                                               |Danny
|07/Sep 09:39                                               |Pedro
|07/Sep 09:39                                               |Luis
|07/Sep 09:39                                               |Julio
|07/Sep 09:39                                               |Julio
|07/Sep 09:39                                               |Ramon
|07/Sep 09:39                                               |Luis
|07/Sep 09:39                                               |Danny
|07/Sep 09:39                                               |Danny
|07/Sep 09:39                                               |Fernando
```

e) Se puede concatenar columnas así como otras operaciones que se pueden probar en la siguiente dirección:

select TIMESTAMPTOSTRING(rowtime, 'dd/MMM HH:mm') as fechaCreacion,

nombre + ' ' + ucase(apellido) as nombre_completo

from usuario emit changes;

```
ksql> select TIMESTAMPTOSTRING(rowtime, 'dd/MMM HH:mm') as fechaCreacion,
>nombre + ' ' + ucase(apellido) as nombre_completo
>from usuario emit changes;
>
+-------------------------------------------------------------+-----------------------------
|FECHACREACION                                                |NOMBRE_COMPLETO
+-------------------------------------------------------------+-----------------------------
|07/Sep 09:22                                                 |Jaime PEREZ
|07/Sep 09:22                                                 |Jimena VARGAS
|07/Sep 09:30                                                 |Dayana PEREZ
|07/Sep 09:30                                                 |Rita GOMEZ
|07/Sep 09:38                                                 |Danny VARGAS
|07/Sep 09:38                                                 |Maria QUISPE
|07/Sep 09:38                                                 |Ana QUISPE
|07/Sep 09:38                                                 |Maria JIMENES
|07/Sep 09:39                                                 |Julio BALLIVIAN
|07/Sep 09:39                                                 |Pedro JIMENES
|07/Sep 09:39                                                 |Juan BALLIVIAN
|07/Sep 09:39                                                 |Danny QUISPE
|07/Sep 09:39                                                 |Luis SALAZAR
|07/Sep 09:39                                                 |Luis MONTERO
|07/Sep 09:39                                                 |Maria BALLIVIAN
|07/Sep 09:39                                                 |Danny MONTERO
|07/Sep 09:39                                                 |Pedro PERES
|07/Sep 09:39                                                 |Luis BALLIVIAN
|07/Sep 09:39                                                 |Julio QUISPE
|07/Sep 09:39                                                 |Julio SALAZAR
|07/Sep 09:39                                                 |Ramon VARGAS
|07/Sep 09:39                                                 |Luis SALAZAR
|07/Sep 09:39                                                 |Danny MONTERO
|07/Sep 09:39                                                 |Danny SALAZAR
|07/Sep 09:39                                                 |Fernando QUISPE
|07/Sep 09:39                                                 |Ana PERES
```

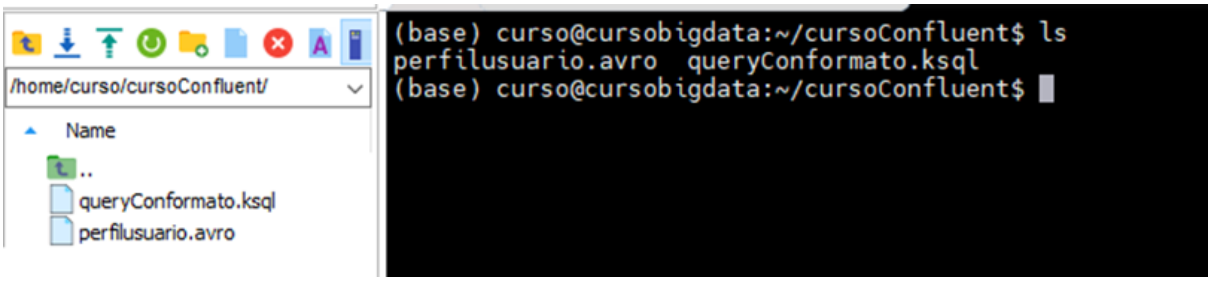
Probe otro comando: select TIMESTAMPTOSTRING(rowtime, 'dd/MMM HH:mm') as fechaCreacion, nombre + ' ' + ucase(apellido) as nombre_completo,ucase(codigo_pais) as Pais_de_origen,calificacion as calificacion_obtenida from usuario emit changes;

```
ksql> select TIMESTAMPTOSTRING(rowtime, 'dd/MMM HH:mm') as fechaCreacion,
>nombre + ' ' + ucase(apellido) as nombre_completo,ucase(codigo_pais) as Pais_de_origen,calificacion as califi
cacion_obtenida
>from usuario emit changes;
+-------------------------+-------------------------+-------------------------+-------------------------+
|FECHACREACION            |NOMBRE_COMPLETO          |PAIS_DE_ORIGEN           |CALIFICACION_OBTENIDA    |
+-------------------------+-------------------------+-------------------------+-------------------------+
|04/Sep 00:03             |Jaime PEREZ              |BO                       |5.2                      |
|04/Sep 00:04             |Jimena VARGAS            |BO                       |7.2                      |
|04/Sep 00:06             |Dayana PEREZ             |BO                       |9.2                      |
|04/Sep 00:07             |Rita GOMEZ               |PE                       |10.2                     |
|04/Sep 00:11             |Danny JIMENES            |BO                       |9.0                      |
|04/Sep 00:11             |Pedro JIMENES            |AR                       |9.0                      |
|04/Sep 00:11             |Danny JIMENES            |AR                       |6.5                      |
|04/Sep 00:11             |Ramon JIMENES            |MX                       |6.4                      |
|04/Sep 00:11             |Ramon MONTERO            |BO                       |9.0                      |
|04/Sep 00:12             |Danny VARGAS             |MX                       |4.7                      |
|04/Sep 00:12             |Juan PERES               |MX                       |5.2                      |
|04/Sep 00:12             |Danny BALLIVIAN          |BO                       |6.5                      |
|04/Sep 00:12             |Ramon BALLIVIAN          |AR                       |9.0                      |
|04/Sep 00:12             |Juan MONTERO             |AR                       |4.8                      |
|04/Sep 00:12             |Fernando BALLIVIAN       |MX                       |5.2                      |
|04/Sep 00:12             |Luis QUISPE              |US                       |6.5                      |
|04/Sep 00:12             |Juan VARGAS              |BO                       |4.7                      |
|04/Sep 00:12             |Danny MONTERO            |MX                       |9.0                      |
```

6. Ejecución de archivos con RUN_SCRIPT

a) Los querys pueden ser almacenados en archivos, para luego ser invocados desde el cliente para su ejecución.

b) Subir el archivo: /código_auxiliar/queryConformato.ksql a la dirección de la MV /home/curso/cursoConfluent



c) Ejecutar el archivo mediante el comando run script

Se obtiene:

run script './cursoConfluent/queryConformato.ksql';

```
ksql> run script './cursoConfluent/queryConformato.ksql';

 Message
----------------------------------------------------------------------------------------------------
 Stream USUARIOCONFORMATO created and running. Created by query with query ID: CSAS_USUARIOCONFORMATO_4
----------------------------------------------------------------------------------------------------
ksql>
```

d) listar los streams para asegurar que el stream fue creado.

```
ksql> list streams;

 Stream Name         | Kafka Topic                 | Format
-------------------------------------------------------------
 KSQL_PROCESSING_LOG | default_ksql_processing_log | JSON
 USUARIO             | PERFILUSUARIO               | JSON
 USUARIOCONFORMATO   | USUARIOCONFORMATO           | JSON
-------------------------------------------------------------
ksql>
```

e) Para observar más propiedades del stream se puede usar la opción:

describe extended USUARIOCONFORMATO;

```
ksql> describe extended USUARIOCONFORMATO;

Name                 : USUARIOCONFORMATO
Type                 : STREAM
Key field            :
Key format           : STRING
Timestamp field      : Not set - using <ROWTIME>
Value format         : JSON
Kafka topic          : USUARIOCONFORMATO (partitions: 1, replication: 1)

 Field       | Type
----------------------------------------
 ROWTIME     | BIGINT           (system)
 ROWKEY      | VARCHAR(STRING)  (system)
 DESCRIPCION | VARCHAR(STRING)
----------------------------------------

Queries that write from this STREAM
-----------------------------------
CSAS_USUARIOCONFORMATO_4 : CREATE STREAM USUARIOCONFORMATO WITH (KAFKA_TOPIC='USUARIOCONFORMATO', PARTITIONS=1, REPLICAS=1) AS
 SELECT ((((((((USUARIO.NOMBRE + ' ') + UCASE(USUARIO.APELLIDO)) + ' desde ') + USUARIO.CODIGO_PAIS) + ' tiene una calificacio
n de ') + CAST(USUARIO.CALIFICACION AS STRING)) + ' estrellas. ') + (CASE WHEN (USUARIO.CALIFICACION < 2.5) THEN 'Pobre' WHEN
(USUARIO.CALIFICACION BETWEEN 2.5 AND 4.2) THEN 'Buena' ELSE 'Excelente' END)) "DESCRIPCION"
FROM USUARIO USUARIO
EMIT CHANGES;

For query topology and execution plan please run: EXPLAIN <QueryId>

Local runtime statistics
------------------------
messages-per-sec:         0   total-messages:       107     last-message: 2023-09-07T09:57:59.641Z

(Statistics of the local KSQL server interaction with the Kafka topic USUARIOCONFORMATO)
ksql>
```

F) Para observar el stream en ejecución

select descripcion from usuarioConformato emit changes;

```
ksql> select descripcion from usuarioConformato emit changes;
+-------------------------------------------------------------------------------------------
|DESCRIPCION
+-------------------------------------------------------------------------------------------
|Jaime PEREZ desde bo tiene una calificacion de 5.2 estrellas. Excelente
|Jimena VARGAS desde bo tiene una calificacion de 7.2 estrellas. Excelente
|Dayana PEREZ desde bo tiene una calificacion de 9.2 estrellas. Excelente
|Rita GOMEZ desde pe tiene una calificacion de 10.2 estrellas. Excelente
|Danny VARGAS desde US tiene una calificacion de 9.0 estrellas. Excelente
|Maria QUISPE desde MX tiene una calificacion de 6.4 estrellas. Excelente
|Ana QUISPE desde US tiene una calificacion de 6.4 estrellas. Excelente
|Maria JIMENES desde BO tiene una calificacion de 4.8 estrellas. Excelente
|Julio BALLIVIAN desde US tiene una calificacion de 5.2 estrellas. Excelente
|Pedro JIMENES desde MX tiene una calificacion de 6.5 estrellas. Excelente
|Juan BALLIVIAN desde MX tiene una calificacion de 9.0 estrellas. Excelente
|Danny QUISPE desde BO tiene una calificacion de 5.2 estrellas. Excelente
|Luis SALAZAR desde BO tiene una calificacion de 9.0 estrellas. Excelente
|Luis MONTERO desde AR tiene una calificacion de 4.7 estrellas. Excelente
|Maria BALLIVIAN desde US tiene una calificacion de 6.4 estrellas. Excelente
|Danny MONTERO desde BO tiene una calificacion de 9.0 estrellas. Excelente
|Pedro PERES desde US tiene una calificacion de 4.8 estrellas. Excelente
|Luis BALLIVIAN desde US tiene una calificacion de 9.0 estrellas. Excelente
|Julio QUISPE desde AR tiene una calificacion de 9.0 estrellas. Excelente
|Julio SALAZAR desde AR tiene una calificacion de 9.0 estrellas. Excelente
|Ramon VARGAS desde MX tiene una calificacion de 6.5 estrellas. Excelente
|Luis SALAZAR desde BO tiene una calificacion de 4.7 estrellas. Excelente
|Danny MONTERO desde AR tiene una calificacion de 6.4 estrellas. Excelente
|Danny SALAZAR desde BO tiene una calificacion de 4.8 estrellas. Excelente
|Fernando QUISPE desde AR tiene una calificacion de 6.4 estrellas. Excelente
|Ana PERES desde MX tiene una calificacion de 9.0 estrellas. Excelente
|Luis PERES desde US tiene una calificacion de 4.7 estrellas. Excelente
|Juan MONTERO desde US tiene una calificacion de 6.5 estrellas. Excelente
```

g) Para eliminar el stream usuarioConformato ejecutar lo siguiente:

drop stream usuarioConformato;

Se borra el stream?

```
ksql>  drop stream usuarioConformato;
Cannot drop USUARIOCONFORMATO.
The following queries read from this source: [].
The following queries write into this source: [CSAS_USUARIOCONFORMATO_4].
You need to terminate them before dropping USUARIOCONFORMATO.
ksql>
```

No, esto debido a que existe un query activo que debe cortarse. Para realizar el borrado del mismo se debe borrar primero ese query activo.

terminate query CSAS_USUARIOCONFORMATO_4;

```
ksql> terminate query CSAS_USUARIOCONFORMATO_4;

 Message
--------------------
 Query terminated.
--------------------
ksql>
```

Ejecutar de nuevo el primer comando y se podrá borrar el stream

```
ksql>  drop stream usuarioConformato;

 Message
----------------------------------------------------------------
 Source `USUARIOCONFORMATO` (topic: USUARIOCONFORMATO) was dropped.
----------------------------------------------------------------
ksql>
```

h) Que sucede si intento borrar de nuevo el objeto?

Sale un mensaje indicando que el stream no existe.

```
ksql> drop stream usuarioConformato;

 Message
-------------------------------------------
 Source `USUARIOCONFORMATO` does not exist.
-------------------------------------------
ksql>
```

I) Existe la posibilidad de que el elemento ya haya sido borrado, en casos como estos se utiliza la sentencia IF EXISTS

```
ksql> drop stream IF EXISTS usuarioConformato;
Source USUARIOCONFORMATO does not exist.
ksql>
```

7. Crear una tabla.

a) Crear un tópico llamado PAIS

kafka-topics --zookeeper localhost:2181 --topic PAIS --create --partitions 1 --replication-factor 1

```
(base) curso@cursobigdata:~$ kafka-topics --zookeeper localhost:2181 --topic PAIS --create --partitions 1 --replication-factor
 1
Created topic PAIS.
(base) curso@cursobigdata:~$ kafka-topics --zookeeper localhost:2181 --list
PAIS
PERFILUSUARIO
USUARIOCONFORMATO
USUARIOS
__confluent.support.metrics
__consumer_offsets
_confluent-command
_confluent-controlcenter-5-4-0-1-AlertHistoryStore-changelog
_confluent-controlcenter-5-4-0-1-Group-ONE_MINUTE-changelog
_confluent-controlcenter-5-4-0-1-Group-THREE_HOURS-changelog
_confluent-controlcenter-5-4-0-1-KSTREAM-OUTERJOIN-0000000105-store-changelog
```

b) Crear un productor en el que se especifique la llave

kafka-console-producer --broker-list 127.0.0.1:9092 --topic PAIS --property "parse.key=true" --property "key.separator=:"

```
(base) curso@cursobigdata:~$ kafka-console-producer --broker-list 127.0.0.1:9092 --topic PAIS --property "parse.key=true" --pr
operty "key.separator=:"
>
```

c) Enviar datos de países con el formato: clave, codigo_pais, país, desde el productor al tópico.

```
(base) curso@cursobigdata:~$ kafka-console-producer --broker-list 127.0.0.1:9092 --topic PAIS --property "parse.key=true" --pr
operty "key.separator=:"
>bo:bo,Bolivia
>pe:pe,Peru
>ec:ec,Ecuador
>co:co,Colombia
>
```

d) En el cliente KSQL. Crear la tabla, especificando el tópico del que se obtendrán los datos

create table tabla_pais(codigo_pais varchar, nombre varchar) WITH (KAFKA_TOPIC='PAIS', VALUE_FORMAT='DELIMITED', KEY='codigo_pais');

```
ksql> create table tabla_pais(codigo_pais varchar, nombre varchar) WITH (KAFKA_TOPIC='PAIS', VALUE_FORMAT='DELIMITED', KEY='co
digo_pais');

 Message
---------------
 Table created
---------------
ksql>
```

e) En el cliente KSQL. Mostrar las tablas existentes:

show tables;

```
ksql> show tables;

 Table Name | Kafka Topic | Format    | Windowed
-------------------------------------------------
 TABLA_PAIS | PAIS        | DELIMITED | false
-------------------------------------------------
ksql>
```

f) En el cliente KSQL. Describir la tabla

describe tabla_pais;

```
ksql> describe tabla_pais;

Name                 : TABLA_PAIS
 Field       | Type
----------------------------------------
 ROWTIME     | BIGINT           (system)
 ROWKEY      | VARCHAR(STRING)  (system)
 CODIGO_PAIS | VARCHAR(STRING)
 NOMBRE      | VARCHAR(STRING)
----------------------------------------
For runtime statistics and query details run: DESCRIBE EXTENDED <Stream,Table>;
ksql>
```

g) utilizar la descripción completa de la tabla para identificar el campo que se encuentra como llave.

describe extended tabla_pais;

```
ksql> describe extended tabla_pais;

Name                 : TABLA_PAIS
Type                 : TABLE
Key field            : CODIGO_PAIS
Key format           : STRING
Timestamp field      : Not set - using <ROWTIME>
Value format         : DELIMITED
Kafka topic          : PAIS (partitions: 1, replication: 1)

 Field       | Type
----------------------------------------
 ROWTIME     | BIGINT           (system)
 ROWKEY      | VARCHAR(STRING)  (system)
 CODIGO_PAIS | VARCHAR(STRING)
 NOMBRE      | VARCHAR(STRING)
----------------------------------------

Local runtime statistics
------------------------


(Statistics of the local KSQL server interaction with the Kafka topic PAIS)
ksql>
```

h) En el cliente KSQL. Realizar un query a la tabla, para identificar los valores que contiene apuntando al tópico.

select codigo_pais, nombre from tabla_pais emit changes;

```
ksql> select codigo_pais, nombre from tabla_pais emit changes;
+-------------------------------------------------------------+-------------------------------------
|CODIGO_PAIS                                                  |NOMBRE
+-------------------------------------------------------------+-------------------------------------
|bo                                                           |Bolivia
|pe                                                           |Peru
|ec                                                           |Ecuador
|co                                                           |Colombia

Press CTRL-C to interrupt
```

i) En el cliente KSQL. Ejecutar un query para obtener el registro correspondiente a Bolivia:

select codigo_pais, nombre from tabla_pais where codigo_pais = 'bo' emit changes;

```
ksql> select codigo_pais, nombre from tabla_pais where codigo_pais = 'bo' emit changes;
+-------------------------------------------------------------+----------------------------
|CODIGO_PAIS                                                  |NOMBRE
+-------------------------------------------------------------+----------------------------
|bo                                                           |Bolivia

Press CTRL-C to interrupt
```

j) En el cliente ksql. Generar un query para obtener un solo valor a partir del índice igual a 'bo' que pertenece al país Bolivia

select codigo_pais, nombre from tabla_pais where codigo_pais='bo' emit changes limit 1;

```
ksql> select codigo_pais, nombre from tabla_pais where codigo_pais='bo' emit changes limit 1;
+-----------------------------------------------------------+-------------------------------
|CODIGO_PAIS                                                |NOMBRE
+-----------------------------------------------------------+-------------------------------
|bo                                                         |Bolivia
Limit Reached
Query terminated
ksql>
```

k) En la consola del productor. Probar un update cambiando el nombre de Bolivia por Bosnia (Update)

bo:bo, Bosnia

```
operty "key.separator=:"
>bo:bo,Bolivia
>pe:pe,Peru
>ec:ec,Ecuador
>co:co,Colombia
>bo:bo,Bosnia
>
```

Realizando el query en el cliente ksql para ver el cambio:

select codigo_pais, nombre from tabla_pais where codigo_pais='bo' emit changes limit 1;

```
ksql> select codigo_pais, nombre from tabla_pais where codigo_pais='bo' emit changes limit 1;
+-----------------------------------------------------------+-------------------------------
|CODIGO_PAIS                                                |NOMBRE
+-----------------------------------------------------------+-------------------------------
|bo                                                         |Bosnia
Limit Reached
Query terminated
ksql>
```

l) En la consola del productor. Insertar el valor br de Brasil.

br:br, Brasil

```
operty "key.separator=:"
>bo:bo,Bolivia
>pe:pe,Peru
>ec:ec,Ecuador
>co:co,Colombia
>bo:bo,Bosnia
>br:br,Brasil
>
```

Verificando la inserción.

select codigo_pais, nombre from tabla_pais where codigo_pais='br' emit changes limit 1;

```
ksql> select codigo_pais, nombre from tabla_pais where codigo_pais='br' emit changes limit 1;
+-------------------------------------------------------------+-------------------------------
|CODIGO_PAIS                                                  |NOMBRE
+-------------------------------------------------------------+-------------------------------
|br                                                           |Brasil
Limit Reached
Query terminated
ksql>
```

8 . Realizar Joins

Realizar un join entre el stream creado USUARIO y la tabla TABLA_PAIS, generados en los puntos 3.c y 6.d

a) En el cliente ksql. Generar un select hacia la tabla TABLA_PAIS

select codigo_pais, nombre from tabla_pais emit changes;

```
ksql> select codigo_pais, nombre from tabla_pais emit changes;
+-------------------------------------------------------------+----------------
|CODIGO_PAIS                                                  |NOMBRE
+-------------------------------------------------------------+----------------
|pe                                                           |Peru
|ec                                                           |Ecuador
|co                                                           |Colombia
|bo                                                           |Bosnia
|br                                                           |Brasil

Press CTRL-C to interrupt
```

b) En el cliente ksql. Crear una consulta para el stream USUARIO

select codigo_pais, nombre from usuario emit changes;

```
ksql> select codigo_pais, nombre from usuario emit changes;
+-------------------------------------------------------------+---------------
|CODIGO_PAIS                                                  |NOMBRE
+-------------------------------------------------------------+---------------
|bo                                                           |Jaime
|bo                                                           |Jimena
|bo                                                           |Dayana
|pe                                                           |Rita
|US                                                           |Danny
|MX                                                           |Maria
|US                                                           |Ana
|BO                                                           |Maria
|US                                                           |Julio
|MX                                                           |Pedro
|MX                                                           |Juan
|BO                                                           |Danny
|BO                                                           |Luis
|AR                                                           |Luis
|US                                                           |Maria
|BO                                                           |Danny
|US                                                           |Pedro
|US                                                           |Luis
|AR                                                           |Julio
|AR                                                           |Julio
|MX                                                           |Ramon
|BO                                                           |Luis
|AR                                                           |Danny
|BO                                                           |Danny
|AR                                                           |Fernando
|MX                                                           |Ana
|US                                                           |Luis
```

c) En el cliente ksql. Ejecutar el query que une ambas tablas:

select a.nombre, a.apellido, a.codigo_pais, b.nombre as nombre_pais

from USUARIO a

left join TABLA_PAIS b on a.codigo_pais = b.codigo_pais EMIT CHANGES;

```
ksql> select a.nombre, a.apellido, a.codigo_pais, b.nombre as nombre_pais
>from USUARIO a
>left join TABLA_PAIS b on a.codigo_pais = b.codigo_pais EMIT CHANGES;
>
+----------------------------+----------------------------+----------------------------+----------------------------+
|A_NOMBRE                    |APELLIDO                    |A_CODIGO_PAIS               |NOMBRE_PAIS                 |
+----------------------------+----------------------------+----------------------------+----------------------------+
|Jaime                       |Perez                       |bo                          |Bosnia                      |
|Jimena                      |Vargas                      |bo                          |Bosnia                      |
|Dayana                      |Perez                       |bo                          |Bosnia                      |
|Rita                        |Gomez                       |pe                          |Peru                        |
|Danny                       |Vargas                      |US                          |null                        |
|Maria                       |Quispe                      |MX                          |null                        |
|Ana                         |Quispe                      |US                          |null                        |
|Maria                       |Jimenes                     |BO                          |null                        |
|Julio                       |Ballivian                   |US                          |null                        |
|Pedro                       |Jimenes                     |MX                          |null                        |
|Juan                        |Ballivian                   |MX                          |null                        |
|Danny                       |Quispe                      |BO                          |null                        |
|Luis                        |Salazar                     |BO                          |null                        |
|Luis                        |Montero                     |AR                          |null                        |
|Maria                       |Ballivian                   |US                          |null                        |
|Danny                       |Montero                     |BO                          |null                        |
|Pedro                       |Peres                       |US                          |null                        |
|Luis                        |Ballivian                   |US                          |null                        |
|Julio                       |Quispe                      |AR                          |null                        |
|Julio                       |Salazar                     |AR                          |null                        |
|Ramon                       |Vargas                      |MX                          |null                        |
|Luis                        |Salazar                     |BO                          |null                        |
|Danny                       |Montero                     |AR                          |null                        |
|Danny                       |Salazar                     |BO                          |null                        |
|Fernando                    |Quispe                      |AR                          |null                        |
```

d) En el cliente ksql. Crear un stream basado en un join:

create stream juntos as

select a.nombre +

' ' + ucase(a.apellido) +

' desde ' + b.nombre +

' tiene una calificación de ' + cast(a.calificacion as varchar) + ' estrellas.' as descripcion

from usuario a

left join tabla_pais b on a.codigo_pais=b.codigo_pais;

```
ksql> create stream juntos as
>select a.nombre +
>' ' + ucase(a.apellido) +
>' desde ' + b.nombre +
>' tiene una calificación de ' + cast(a.calificacion as varchar) + ' estrellas.' as descripcion
>from usuario a
>left join tabla_pais b on a.codigo_pais=b.codigo_pais;
>

 Message
----------------------------------------------------------------------------
 Stream JUNTOS created and running. Created by query with query ID: CSAS_JUNTOS_9
----------------------------------------------------------------------------
ksql>
```

e) consultar el stream creado.

Select * from juntos emit changes;

```
ksql> Select * from juntos emit changes;
+----------------------------------------+----------------------------------------+----------------------------------------+
|ROWTIME                                 |ROWKEY                                  |DESCRIPCION                             |
+----------------------------------------+----------------------------------------+----------------------------------------+
|1694078521574                           |bo                                      |Jaime PEREZ desde Bosnia tiene una calif|
|                                        |                                        |icación de 5.2 estrellas.               |
|1694078544421                           |bo                                      |Jimena VARGAS desde Bosnia tiene una cal|
|                                        |                                        |ificación de 7.2 estrellas.             |
|1694079003934                           |bo                                      |Dayana PEREZ desde Bosnia tiene una cali|
|                                        |                                        |ficación de 9.2 estrellas.              |
|1694079017838                           |pe                                      |Rita GOMEZ desde Peru tiene una califica|
|                                        |                                        |ción de 10.2 estrellas.                 |
|1694079530264                           |US                                      |Danny VARGAS desde null tiene una califi|
|                                        |                                        |cación de 9.0 estrellas.                |
|1694079530974                           |MX                                      |Maria QUISPE desde null tiene una califi|
|                                        |                                        |cación de 6.4 estrellas.                |
|1694079533626                           |US                                      |Ana QUISPE desde null tiene una califica|
|                                        |                                        |ción de 6.4 estrellas.                  |
|1694079537272                           |BO                                      |Maria JIMENES desde null tiene una calif|
|                                        |                                        |icación de 4.8 estrellas.               |
|1694079540584                           |US                                      |Julio BALLIVIAN desde null tiene una cal|
|                                        |                                        |ificación de 5.2 estrellas.             |
|1694079542413                           |MX                                      |Pedro JIMENES desde null tiene una calif|
|                                        |                                        |icación de 6.5 estrellas.               |
|1694079542536                           |MX                                      |Juan BALLIVIAN desde null tiene una cali|
|                                        |                                        |ficación de 9.0 estrellas.              |
|1694079543225                           |BO                                      |Danny QUISPE desde null tiene una califi|
|                                        |                                        |cación de 5.2 estrellas.                |
|1694079547781                           |BO                                      |Luis SALAZAR desde null tiene una califi|
|                                        |                                        |cación de 9.0 estrellas.                |
|1694079550068                           |AR                                      |Luis MONTERO desde null tiene una califi|
```