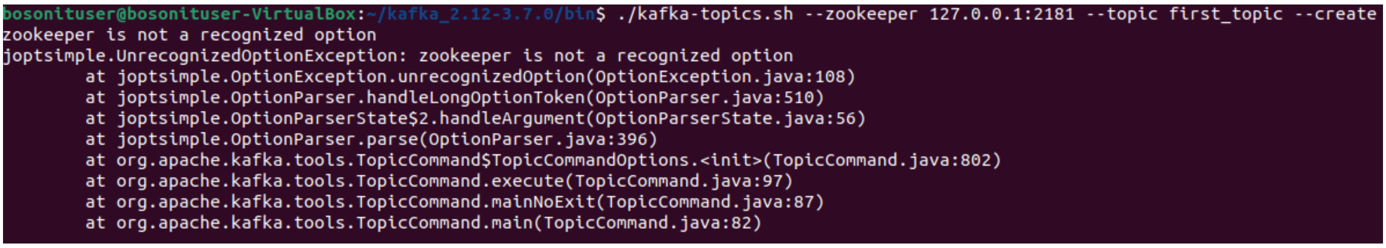
GUIA PRACTICA KAFKA

INICIAR KAFKA:

bin/zookeeper-server-start.sh config/zookeeper.properties

bin/kafka-server-start.sh config/server.properties

TOPICS:



```
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0/bin$ ./kafka-topics.sh --zookeeper 127.0.0.1:2181 --topic first_topic --create
zookeeper is not a recognized option
joptsimple.UnrecognizedOptionException: zookeeper is not a recognized option
        at joptsimple.OptionException.unrecognizedOption(OptionException.java:108)
        at joptsimple.OptionParser.handleLongOptionToken(OptionParser.java:510)
        at joptsimple.OptionParserState$2.handleArgument(OptionParserState.java:56)
        at joptsimple.OptionParser.parse(OptionParser.java:396)
        at org.apache.kafka.tools.TopicCommand$TopicCommandOptions.<init>(TopicCommand.java:802)
        at org.apache.kafka.tools.TopicCommand.execute(TopicCommand.java:97)
        at org.apache.kafka.tools.TopicCommand.mainNoExit(TopicCommand.java:87)
        at org.apache.kafka.tools.TopicCommand.main(TopicCommand.java:82)
```

- Crear first_topic: bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic first_topic --create

- Listar topics: bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --list

- Describir: bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic first_topic --describe

- Eliminar first_topic: bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic first_topic --delete

PRODUCERS:

bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic first_topic

CONSUMERS:

bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first_topic --from-beginning

CONSUMER GROUPS:

bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic first_topic --group my-first-app

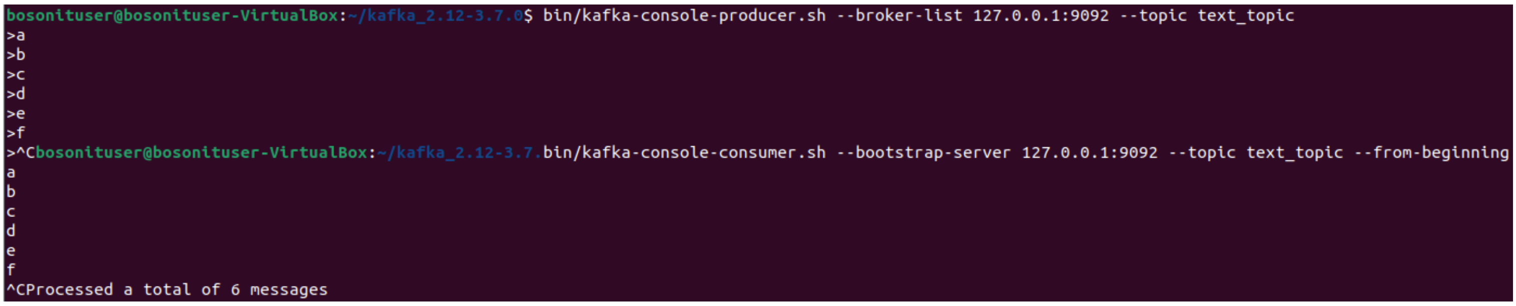
bin/kafka-consumer-groups.sh --bootstrap-server 127.0.0.1:9092 --list

bin/kafka-consumer-groups.sh --bootstrap-server 127.0.0.1:9092 --describe --group my-first-app

GUIA KAFKA + PRESTO -> KAFKA + TRINO

1. Crear text_topic y enviar mensajes con producer, comprobar con consumer:

bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic text_topic --create



```
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic text_topic
>a
>b
>c
>d
>e
>f
>^Cbosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic text_topic --from-beginning
a
b
c
d
e
f
^CProcessed a total of 6 messages
```

2. Pasamos a Trino, configuramos el conector de kafka:

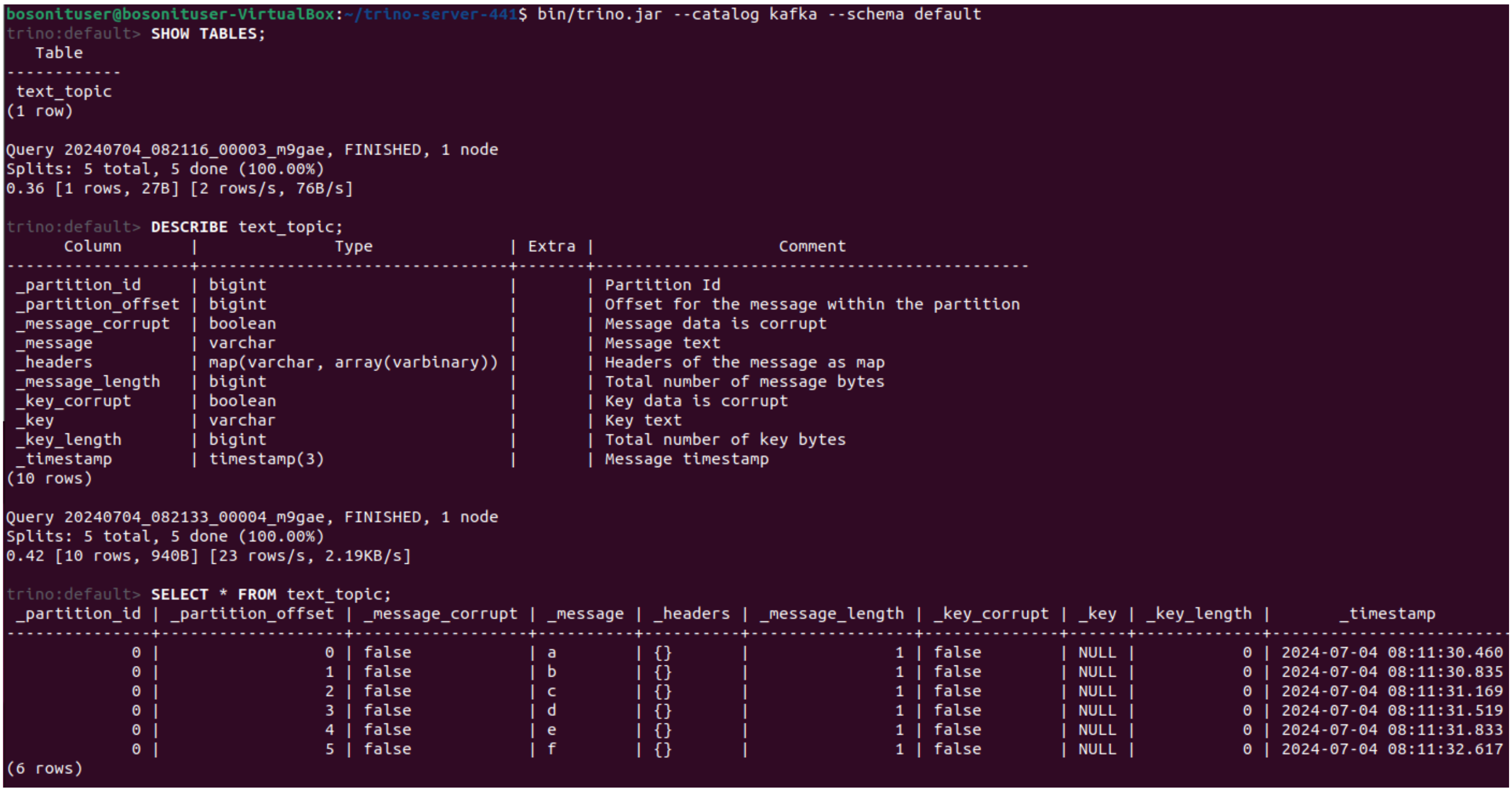
cd etc -> mkdir catalog -> nano kafka.properties

connector.name=kafka
kafka.nodes=localhost:9092
kafka.table-names=text_topic
kafka.hide-internal-columns=false

3. Iniciar y consultamos Trino

./bin/launcher start -> started as 12424

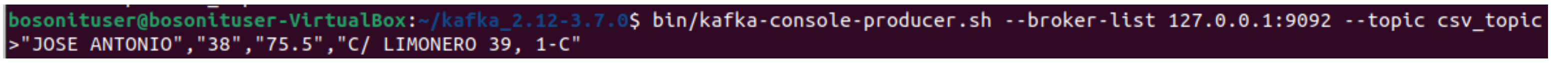
./bin/trino.jar --catalog kafka --schema default
CSV



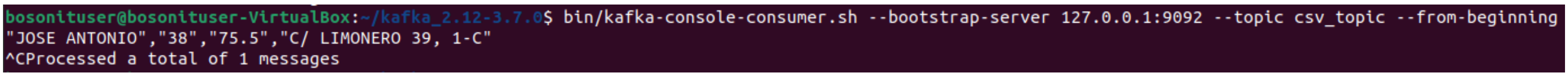
```
bosonituser@bosonituser-VirtualBox:~/trino-server-441$ bin/trino.jar --catalog kafka --schema default
trino:default> SHOW TABLES;
   Table
------------
 text_topic
(1 row)

Query 20240704_082116_00003_m9gae, FINISHED, 1 node
Splits: 5 total, 5 done (100.00%)
0.36 [1 rows, 27B] [2 rows/s, 76B/s]

trino:default> DESCRIBE text_topic;
      Column       |              Type              | Extra |                   Comment
-------------------+--------------------------------+-------+---------------------------------------------
 _partition_id     | bigint                         |       | Partition Id
 _partition_offset | bigint                         |       | Offset for the message within the partition
 _message_corrupt  | boolean                        |       | Message data is corrupt
 _message          | varchar                        |       | Message text
 _headers          | map(varchar, array(varbinary)) |       | Headers of the message as map
 _message_length   | bigint                         |       | Total number of message bytes
 _key_corrupt      | boolean                        |       | Key data is corrupt
 _key              | varchar                        |       | Key text
 _key_length       | bigint                         |       | Total number of key bytes
 _timestamp        | timestamp(3)                   |       | Message timestamp
(10 rows)

Query 20240704_082133_00004_m9gae, FINISHED, 1 node
Splits: 5 total, 5 done (100.00%)
0.42 [10 rows, 940B] [23 rows/s, 2.19KB/s]

trino:default> SELECT * FROM text_topic;
 _partition_id | _partition_offset | _message_corrupt | _message | _headers | _message_length | _key_corrupt | _key | _key_length |       _timestamp
---------------+-------------------+------------------+----------+----------+-----------------+--------------+------+-------------+------------------------
             0 |                 0 | false            | a        | {}       |               1 | false        | NULL |           0 | 2024-07-04 08:11:30.460
             0 |                 1 | false            | b        | {}       |               1 | false        | NULL |           0 | 2024-07-04 08:11:30.835
             0 |                 2 | false            | c        | {}       |               1 | false        | NULL |           0 | 2024-07-04 08:11:31.169
             0 |                 3 | false            | d        | {}       |               1 | false        | NULL |           0 | 2024-07-04 08:11:31.519
             0 |                 4 | false            | e        | {}       |               1 | false        | NULL |           0 | 2024-07-04 08:11:31.833
             0 |                 5 | false            | f        | {}       |               1 | false        | NULL |           0 | 2024-07-04 08:11:32.617
(6 rows)
```

1. Crear csv_topic y enviar mensajes con producer, comprobar con consumer:

bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic csv_topic --create



```
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic csv_topic
>"JOSE ANTONIO","38","75.5","C/ LIMONERO 39, 1-C"
```



```
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic csv_topic --from-beginning
"JOSE ANTONIO","38","75.5","C/ LIMONERO 39, 1-C"
^CProcessed a total of 1 messages
```

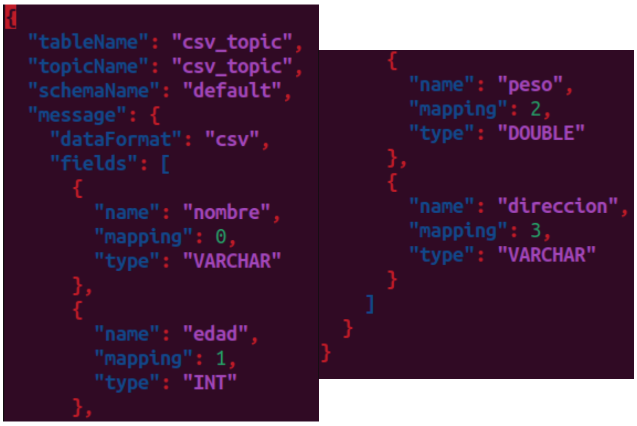
2. Añadir el csv_topic al conector de kafka:

cd etc -> cd catalog -> nano kafka.properties

```
connector.name=kafka
kafka.nodes=localhost:9092
kafka.table-names=text_topic,csv_topic
kafka.hide-internal-columns=false
```

3. Crear un fichero json con la estructura de la tabla (csv):

cd etc -> cd kafka -> nano csv_topic.json

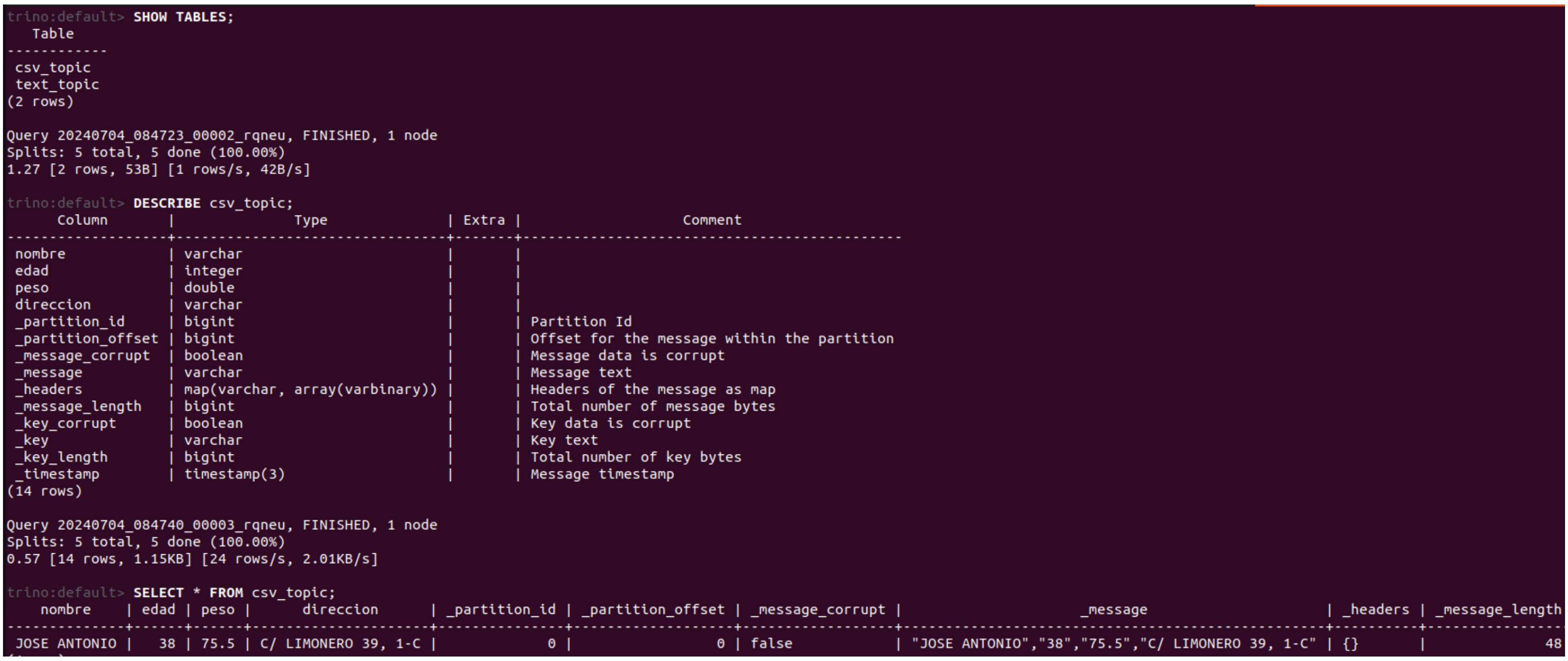


```
{
  "tableName": "csv_topic",
  "topicName": "csv_topic",
  "schemaName": "default",
  "message": {
    "dataFormat": "csv",
    "fields": [
      {
        "name": "nombre",
        "mapping": 0,
        "type": "VARCHAR"
      },
      {
        "name": "edad",
        "mapping": 1,
        "type": "INT"
      },
```

```
      {
        "name": "peso",
        "mapping": 2,
        "type": "DOUBLE"
      },
      {
        "name": "direccion",
        "mapping": 3,
        "type": "VARCHAR"
      }
    ]
  }
}
```

4. Reiniciar trino y consultar csv_topic:

./bin/launcher restart -> ./bin/trino.jar --catalog kafka --schema default

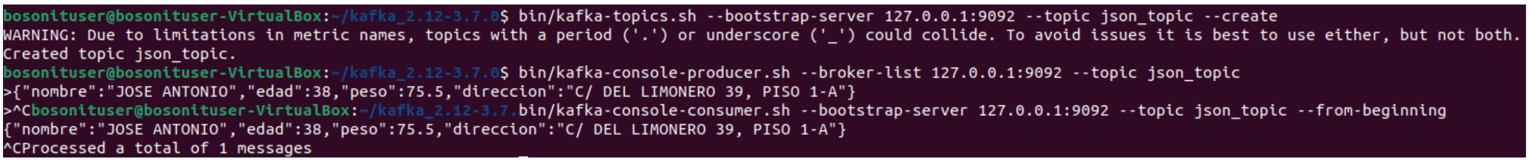


```
trino:default> SHOW TABLES;
   Table
------------
 csv_topic
 text_topic
(2 rows)

Query 20240704_084723_00002_rqneu, FINISHED, 1 node
Splits: 5 total, 5 done (100.00%)
1.27 [2 rows, 53B] [1 rows/s, 42B/s]

trino:default> DESCRIBE csv_topic;
      Column       |              Type              | Extra |                   Comment
-------------------+--------------------------------+-------+---------------------------------------------
 nombre            | varchar                        |       |
 edad              | integer                        |       |
 peso              | double                         |       |
 direccion         | varchar                        |       |
 _partition_id     | bigint                         |       | Partition Id
 _partition_offset | bigint                         |       | Offset for the message within the partition
 _message_corrupt  | boolean                        |       | Message data is corrupt
 _message          | varchar                        |       | Message text
 _headers          | map(varchar, array(varbinary)) |       | Headers of the message as map
 _message_length   | bigint                         |       | Total number of message bytes
 _key_corrupt      | boolean                        |       | Key data is corrupt
 _key              | varchar                        |       | Key text
 _key_length       | bigint                         |       | Total number of key bytes
 _timestamp        | timestamp(3)                   |       | Message timestamp
(14 rows)

Query 20240704_084740_00003_rqneu, FINISHED, 1 node
Splits: 5 total, 5 done (100.00%)
0.57 [14 rows, 1.15KB] [24 rows/s, 2.01KB/s]

trino:default> SELECT * FROM csv_topic;
    nombre    | edad | peso |      direccion      | _partition_id | _partition_offset | _message_corrupt |                    _message                      | _headers | _message_length
--------------+------+------+---------------------+---------------+-------------------+------------------+--------------------------------------------------+----------+----------------
 JOSE ANTONIO |   38 | 75.5 | C/ LIMONERO 39, 1-C |             0 |                 0 | false            | "JOSE ANTONIO","38","75.5","C/ LIMONERO 39, 1-C" | {}       |              48
```

JSON

1. Crear json_topic y enviar mensajes con producer, comprobar con consumer:

bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic json_topic --create



```
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic json_topic --create
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic json_topic.
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic json_topic
>{"nombre":"JOSE ANTONIO","edad":38,"peso":75.5,"direccion":"C/ DEL LIMONERO 39, PISO 1-A"}
>^Cbosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.bin/kafka-console-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic json_topic --from-beginning
{"nombre":"JOSE ANTONIO","edad":38,"peso":75.5,"direccion":"C/ DEL LIMONERO 39, PISO 1-A"}
^CProcessed a total of 1 messages
```

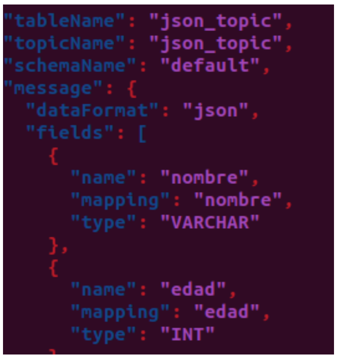
2. Añadir el json_topic al conector de kafka:

cd etc -> cd catalog -> nano kafka.properties

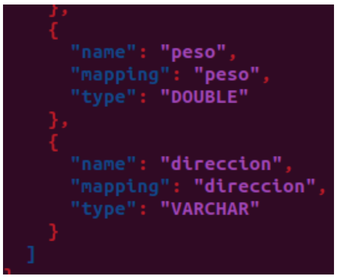
connector.name=kafka
kafka.nodes=localhost:9092
kafka.table-names=text_topic,csv_topic,json_topic
kafka.hide-internal-columns=false

3. Crear un fichero json con la estructura de la tabla (json):

cd etc -> cd kafka -> nano json_topic.json



```
"tableName": "json_topic",
"topicName": "json_topic",
"schemaName": "default",
"message": {
  "dataFormat": "json",
  "fields": [
    {
      "name": "nombre",
      "mapping": "nombre",
      "type": "VARCHAR"
    },
    {
      "name": "edad",
      "mapping": "edad",
      "type": "INT"
```

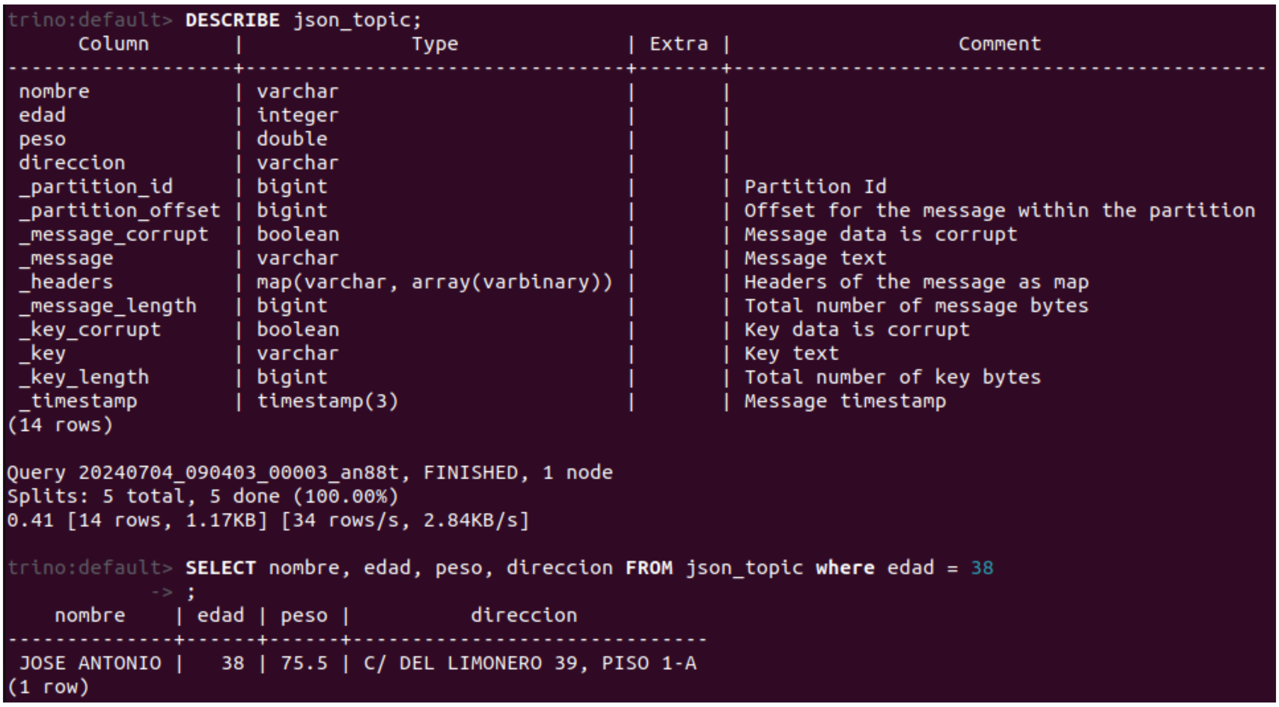


```
    },
    {
      "name": "peso",
      "mapping": "peso",
      "type": "DOUBLE"
    },
    {
      "name": "direccion",
      "mapping": "direccion",
      "type": "VARCHAR"
    }
  ]
```

4. Reiniciar trino y consultar json_topic:

./bin/launcher restart -> ./bin/trino.jar --catalog kafka --schema default

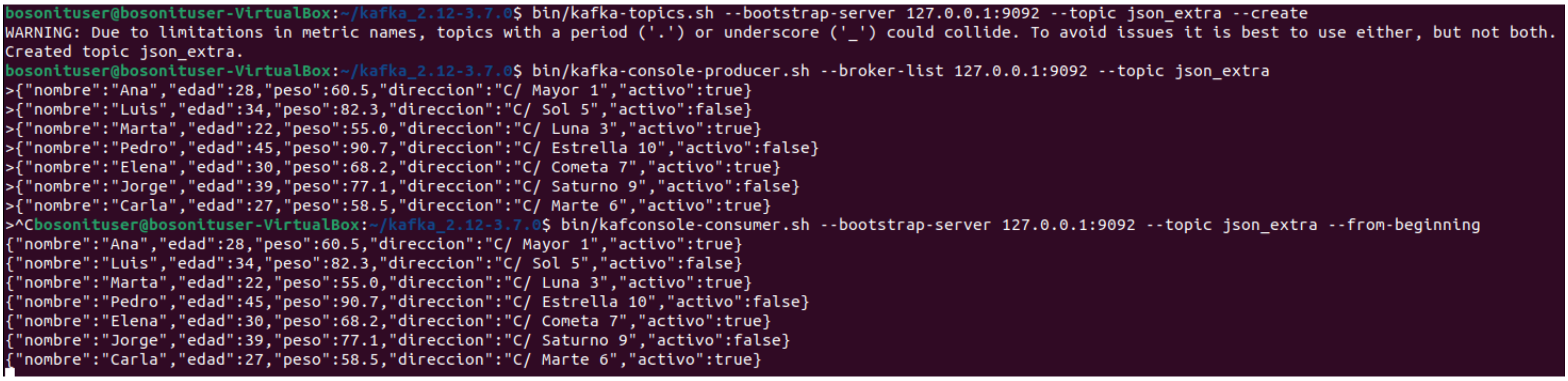
EJERCICIO EXTRA



```
trino:default> DESCRIBE json_topic;
      Column       |              Type              | Extra |                  Comment
-------------------+--------------------------------+-------+--------------------------------------------
 nombre            | varchar                        |       |
 edad              | integer                        |       |
 peso              | double                         |       |
 direccion         | varchar                        |       |
 _partition_id     | bigint                         |       | Partition Id
 _partition_offset | bigint                         |       | Offset for the message within the partition
 _message_corrupt  | boolean                        |       | Message data is corrupt
 _message          | varchar                        |       | Message text
 _headers          | map(varchar, array(varbinary)) |       | Headers of the message as map
 _message_length   | bigint                         |       | Total number of message bytes
 _key_corrupt      | boolean                        |       | Key data is corrupt
 _key              | varchar                        |       | Key text
 _key_length       | bigint                         |       | Total number of key bytes
 _timestamp        | timestamp(3)                   |       | Message timestamp
(14 rows)

Query 20240704_090403_00003_an88t, FINISHED, 1 node
Splits: 5 total, 5 done (100.00%)
0.41 [14 rows, 1.17KB] [34 rows/s, 2.84KB/s]

trino:default> SELECT nombre, edad, peso, direccion FROM json_topic where edad = 38
            -> ;
    nombre    | edad | peso |          direccion
--------------+------+------+-----------------------------
 JOSE ANTONIO |   38 | 75.5 | C/ DEL LIMONERO 39, PISO 1-A
(1 row)
```

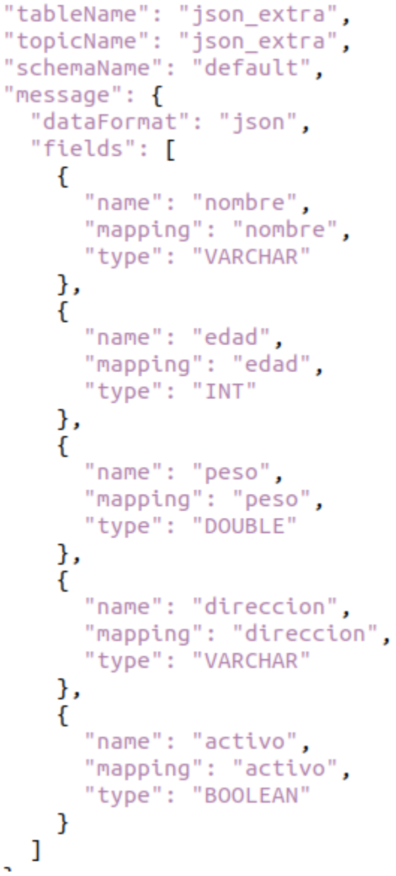
Crea un nuevo topic que maneje datos con formato JSON con variedad de atributos (que tengan cadenas de caracteres, enteros, y tipos de datos reales/flotantes) inserta varios registros (al menos unos 6-10 registros) y crea una tabla en presto asociada a ese topic y lanza consultas sobre el topic (que incluso podría estar en una continua evolución, asumiendo que sigan insertando elementos) con condiciones (cláusulas WHERE) basadas en algunos de los atributos que contienen dichos JSON.

- Nuevo topic json_extra con varios registros (string, int, float, boolean):

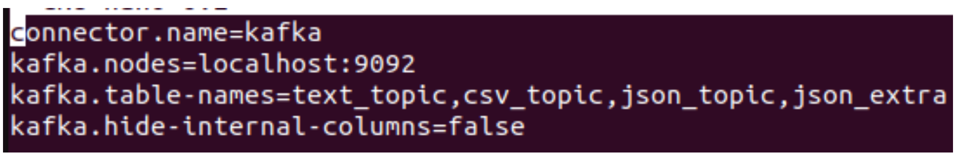


```
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafka-topics.sh --bootstrap-server 127.0.0.1:9092 --topic json_extra --create
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic json_extra.
bosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafka-console-producer.sh --broker-list 127.0.0.1:9092 --topic json_extra
>{"nombre":"Ana","edad":28,"peso":60.5,"direccion":"C/ Mayor 1","activo":true}
>{"nombre":"Luis","edad":34,"peso":82.3,"direccion":"C/ Sol 5","activo":false}
>{"nombre":"Marta","edad":22,"peso":55.0,"direccion":"C/ Luna 3","activo":true}
>{"nombre":"Pedro","edad":45,"peso":90.7,"direccion":"C/ Estrella 10","activo":false}
>{"nombre":"Elena","edad":30,"peso":68.2,"direccion":"C/ Cometa 7","activo":true}
>{"nombre":"Jorge","edad":39,"peso":77.1,"direccion":"C/ Saturno 9","activo":false}
>{"nombre":"Carla","edad":27,"peso":58.5,"direccion":"C/ Marte 6","activo":true}
>^Cbosonituser@bosonituser-VirtualBox:~/kafka_2.12-3.7.0$ bin/kafconsole-consumer.sh --bootstrap-server 127.0.0.1:9092 --topic json_extra --from-beginning
{"nombre":"Ana","edad":28,"peso":60.5,"direccion":"C/ Mayor 1","activo":true}
{"nombre":"Luis","edad":34,"peso":82.3,"direccion":"C/ Sol 5","activo":false}
{"nombre":"Marta","edad":22,"peso":55.0,"direccion":"C/ Luna 3","activo":true}
{"nombre":"Pedro","edad":45,"peso":90.7,"direccion":"C/ Estrella 10","activo":false}
{"nombre":"Elena","edad":30,"peso":68.2,"direccion":"C/ Cometa 7","activo":true}
{"nombre":"Jorge","edad":39,"peso":77.1,"direccion":"C/ Saturno 9","activo":false}
{"nombre":"Carla","edad":27,"peso":58.5,"direccion":"C/ Marte 6","activo":true}
```

- Incluir topic en el conector, y crear jason_extra.json para la estructura de la tabla:

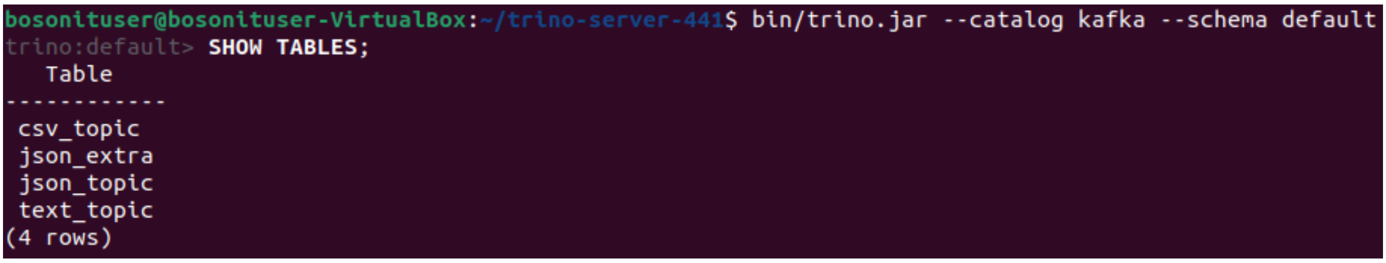


```
"tableName": "json_extra",
"topicName": "json_extra",
"schemaName": "default",
"message": {
  "dataFormat": "json",
  "fields": [
    {
      "name": "nombre",
      "mapping": "nombre",
      "type": "VARCHAR"
    },
    {
      "name": "edad",
      "mapping": "edad",
      "type": "INT"
    },
    {
      "name": "peso",
      "mapping": "peso",
      "type": "DOUBLE"
    },
    {
      "name": "direccion",
      "mapping": "direccion",
      "type": "VARCHAR"
    },
    {
      "name": "activo",
      "mapping": "activo",
      "type": "BOOLEAN"
    }
  ]
```

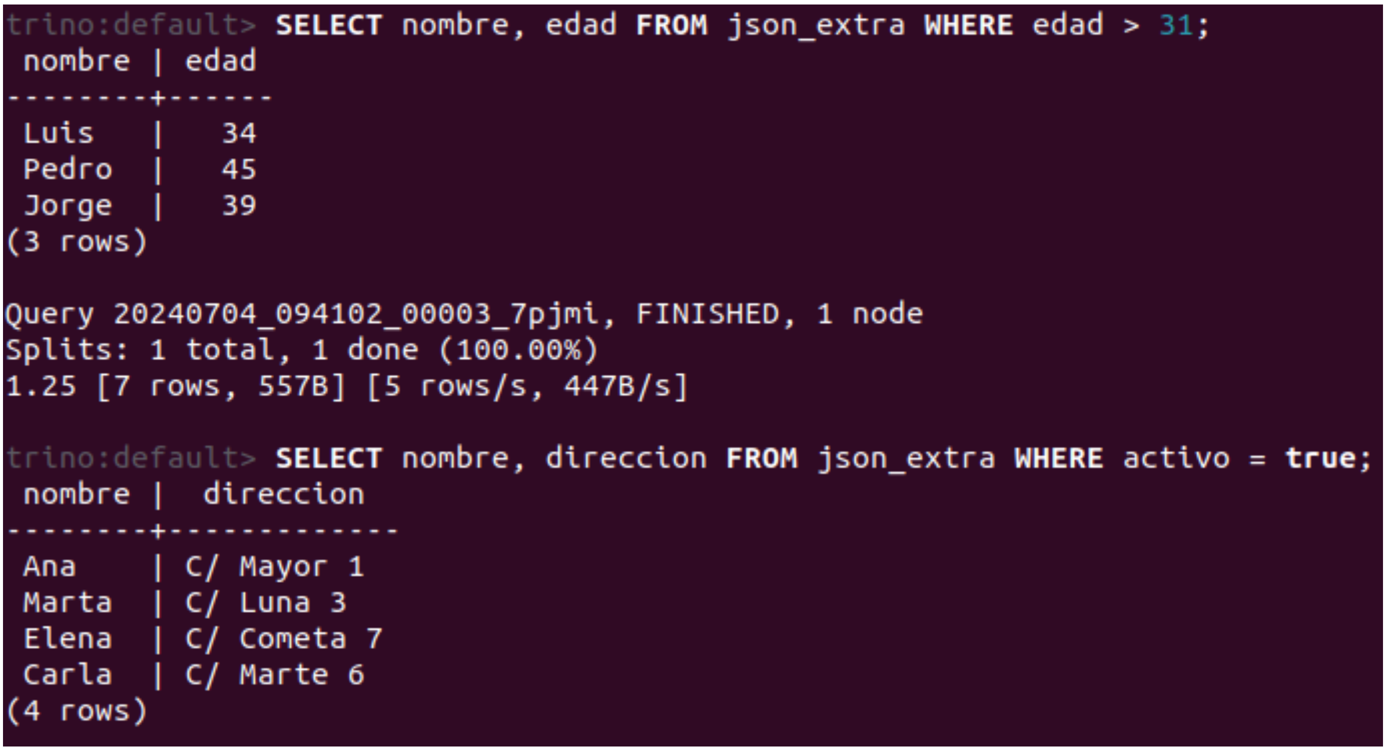


```
connector.name=kafka
kafka.nodes=localhost:9092
kafka.table-names=text_topic,csv_topic,json_topic,json_extra
kafka.hide-internal-columns=false
```

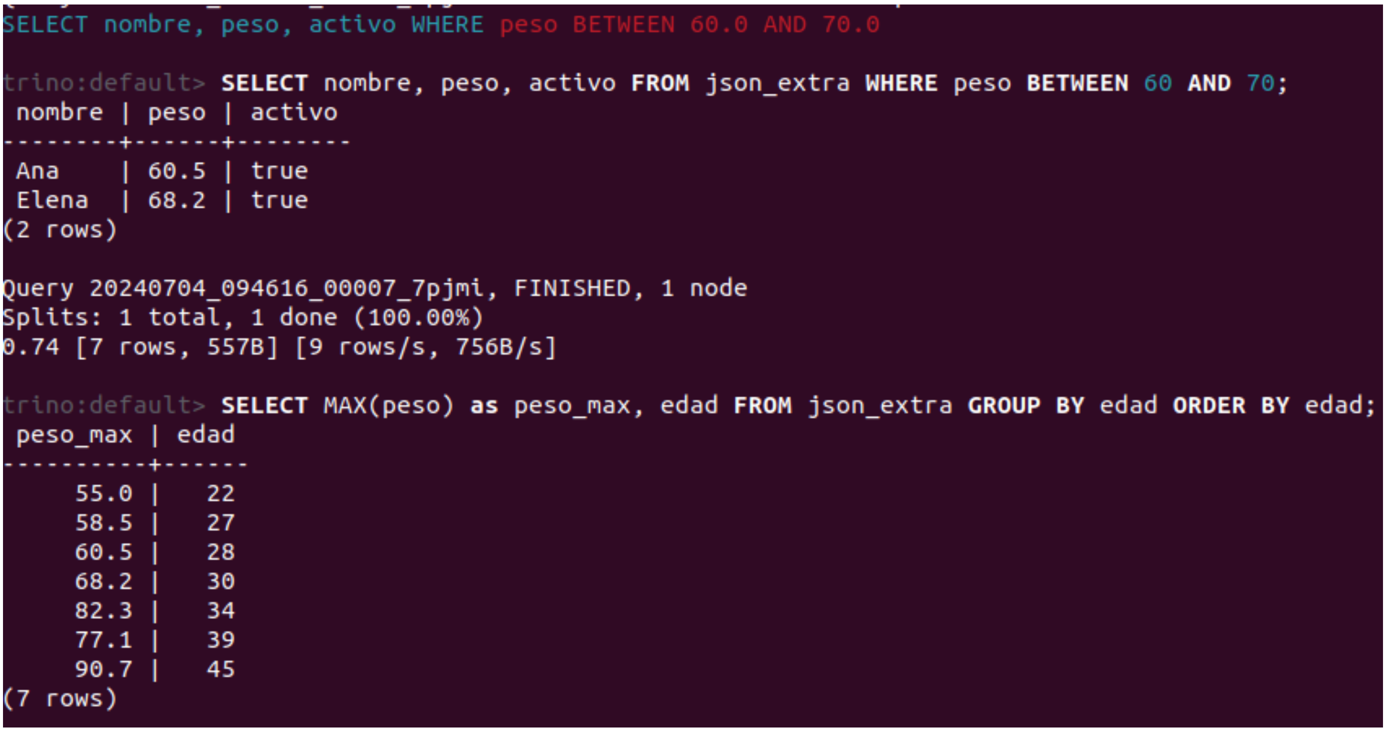
- Reiniciar trino y hacer varias consultas con WHERE:



```
bosonituser@bosonituser-VirtualBox:~/trino-server-441$ bin/trino.jar --catalog kafka --schema default
trino:default> SHOW TABLES;
   Table
------------
 csv_topic
 json_extra
 json_topic
 text_topic
(4 rows)
```



```
trino:default> SELECT nombre, edad FROM json_extra WHERE edad > 31;
 nombre | edad
--------+------
 Luis   |   34
 Pedro  |   45
 Jorge  |   39
(3 rows)

Query 20240704_094102_00003_7pjmi, FINISHED, 1 node
Splits: 1 total, 1 done (100.00%)
1.25 [7 rows, 557B] [5 rows/s, 447B/s]

trino:default> SELECT nombre, direccion FROM json_extra WHERE activo = true;
 nombre |  direccion
--------+-------------
 Ana    | C/ Mayor 1
 Marta  | C/ Luna 3
 Elena  | C/ Cometa 7
 Carla  | C/ Marte 6
(4 rows)
```



```
SELECT nombre, peso, activo WHERE peso BETWEEN 60.0 AND 70.0

trino:default> SELECT nombre, peso, activo FROM json_extra WHERE peso BETWEEN 60 AND 70;
 nombre | peso | activo
--------+------+--------
 Ana    | 60.5 | true
 Elena  | 68.2 | true
(2 rows)

Query 20240704_094616_00007_7pjmi, FINISHED, 1 node
Splits: 1 total, 1 done (100.00%)
0.74 [7 rows, 557B] [9 rows/s, 756B/s]

trino:default> SELECT MAX(peso) as peso_max, edad FROM json_extra GROUP BY edad ORDER BY edad;
 peso_max | edad
----------+------
     55.0 |   22
     58.5 |   27
     60.5 |   28
     68.2 |   30
     82.3 |   34
     77.1 |   39
     90.7 |   45
(7 rows)
```