Instructions

For each answer, please include your answer as text, and any screenshot(s) which demonstrate your answer was executed. Most importantly, make sure to include evidence your answer is correct. This will most likely be a screenshot. If you had issues, problems, or had to make assumptions include them in your answer.

Your Answers:

1. Write KSQL to create a stream named **weblogs** from the JSON keys in the **weblogs** Kafka topic. Make sure to set the TIMESTAMP property to the timestamp from the stream.

   create a stream named weblogs
   > create stream weblogs (browser varchar, OS varchar, Uri varchar, User varchar, timestamp bigint) with (kafka_topic = 'weblogs', value_format = 'json', timestamp = 'timestamp');

   ```
   ksql> create stream weblogs (browser varchar, OS varchar, Uri varchar, User varchar, timestamp big
   int) with (kafka_topic = 'weblogs', value_format = 'json', timestamp = 'timestamp');

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

2. Write a KSQL statement create a persistent stream/table called **homepage** which only displays visitors to the root of the website (/). It should display all columns from the **weblogs** stream.

   create a persistent stream called homepage
   create stream homepage as select * from weblogs where Uri = '/';

   ```
   ksql> create stream homepage as select * from weblogs where Uri = '/';

    Message
   ---------------------------
    Stream created and running
   ---------------------------
   ksql>
   ```

3. Write a KSQL statement to count operating systems users (os) in 60 second windows. After 60 seconds, the counter should reset, and counts should begin again.

   count operating systems users (os) in 60 second window
   select OS, count(*) as OS_count from weblogs window tumbling (size = 60 seconds) group by OS;

```
ksql> select OS, count(*) as OS_count from weblogs window tumbling (size 60 seconds) group by O
osx | 1
win | 1
win | 2
win | 3
osx | 2
win | 1
osx | 1
```

4. Write a KSQL persistent stream/table called **user_activity** which will display a count of user activity on the website within 1-minute sessions.

   display count of user activity in 1 minute sessions
   > select User, count(*) as user_count from weblogs window session (60 seconds) group by User;

```
ksql> select User, count(*) as user_count from weblogs window session (60 seconds) group by User;
lisa | 1
chris | 1
abby | 1
ida | 1
zeke | 1
otto | 1
walt | 1
elle | 1
otto | 2
otto | 3
```

5. Write a KSQL statement to display users who have more than 1 pages of activity in a 1-minute session.

   display count of user activity in 1 minute sessions having > 1 page of activity
   > select User, count(*) as user_count from weblogs window session (60 seconds) group by User having count(*) > 1;

```
ksql> select User, count(*) as user_count from weblogs window session (60 seconds) group by User h
aving count(*) > 1;
walt | 2
surah | 2
bob | 2
```