Create a WAU (Weekly Active User) chart from Preset or Docker Superset. Total 10 points

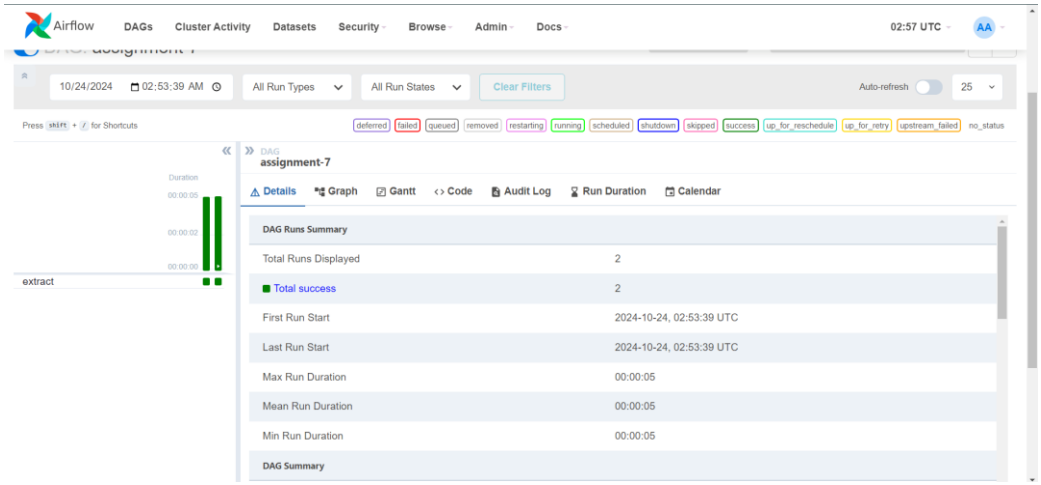
Import two tables in your SnowflakeLinks to an external site. as an ETL DAG in your Airflow (+3pt)

user_session_channel and session_timestamp (under raw_data schema or equivalent)

```
cur.execute("CREATE DATABASE IF NOT EXISTS dev;")
        cur.execute("USE DATABASE dev;")
        cur.execute("CREATE SCHEMA IF NOT EXISTS
dev.raw_data;")
        cur.execute("USE SCHEMA dev.raw_data;")

        # Create user_session_channel table
        cur.execute(
            """
        CREATE TABLE IF NOT EXISTS
dev.raw_data.user_session_channel (
            userId int NOT NULL,
            sessionId varchar(32) PRIMARY KEY,
            channel varchar(32) DEFAULT 'direct'
        );
        """
        )

        # Create session_timestamp table
        cur.execute(
            """
        CREATE TABLE IF NOT EXISTS
dev.raw_data.session_timestamp (
            sessionId varchar(32) PRIMARY KEY,
            ts timestamp
        );
```

```
        """
        )

        # Create stage for S3 data
        cur.execute(
            """
        CREATE OR REPLACE STAGE dev.raw_data.blob_stage
        url = 's3://s3-geospatial/readonly/'
        file_format = (type = csv, skip_header = 1,
field_optionally_enclosed_by = '"');
        """
        )

        # Copy data into user_session_channel table from
S3
        cur.execute(
            """
        COPY INTO dev.raw_data.user_session_channel
        FROM
@dev.raw_data.blob_stage/user_session_channel.csv;
        """
        )

        # Copy data into session_timestamp table from S3
        cur.execute(
            """
        COPY INTO dev.raw_data.session_timestamp
        FROM
@dev.raw_data.blob_stage/session_timestamp.csv;
        """
        )
```

Capture the screenshot of this DAG's detailed page from the Web UI (#1)



Create a ELT DAG in your Airflow to create a JOINED table of the two (+3pt)

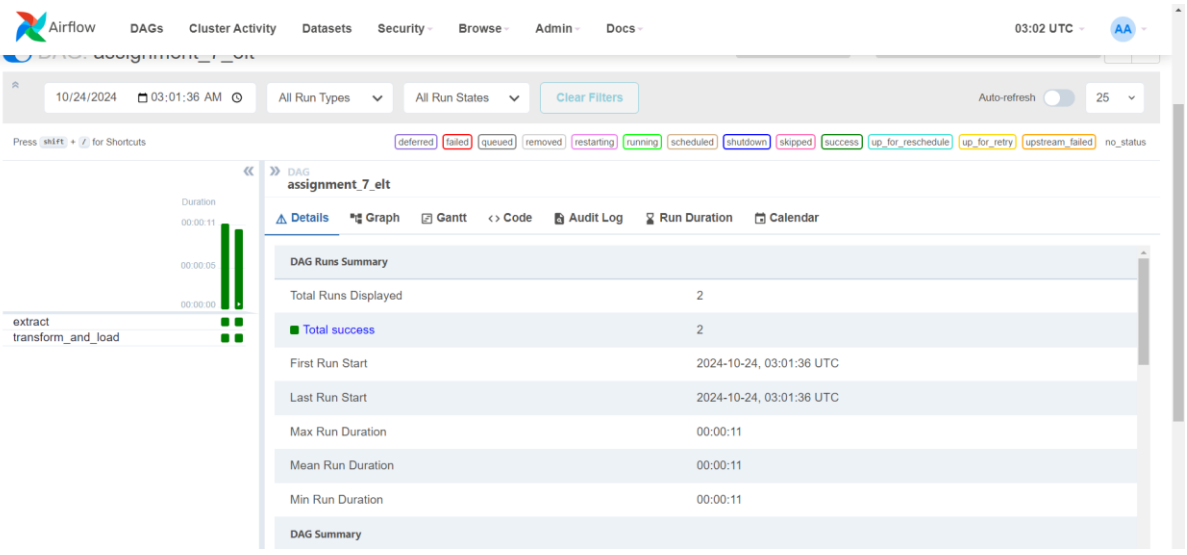
session_summary (under analytics)

Extra point: add one more condition to check duplicate records (+1pt)

```
try:
        cur = return_snowflake_conn()

        # Create analytics schema if it doesn't exist
        cur.execute("CREATE SCHEMA IF NOT EXISTS
dev.analytics;")
        cur.execute("USE SCHEMA dev.analytics;")

        # Create session_summary table with data joined
from user_session_channel and session_timestamp
        cur.execute(
            """
        CREATE TABLE IF NOT EXISTS
dev.analytics.session_summary AS
        SELECT
            u.userId,
            u.sessionId,
            u.channel,
            s.ts
        FROM
            dev.raw_data.user_session_channel u
```

```
        JOIN
            dev.raw_data.session_timestamp s
        ON
            u.sessionId = s.sessionId;
        """
        )

        # Remove duplicates from session_summary based on
sessionId
        cur.execute(
            """
        DELETE FROM dev.analytics.session_summary
        WHERE sessionId IN (
            SELECT sessionId
            FROM (
                SELECT sessionId, COUNT(*) AS cnt
                FROM dev.analytics.session_summary
                GROUP BY sessionId
                HAVING cnt > 1
            )
        );
        """
        )
```

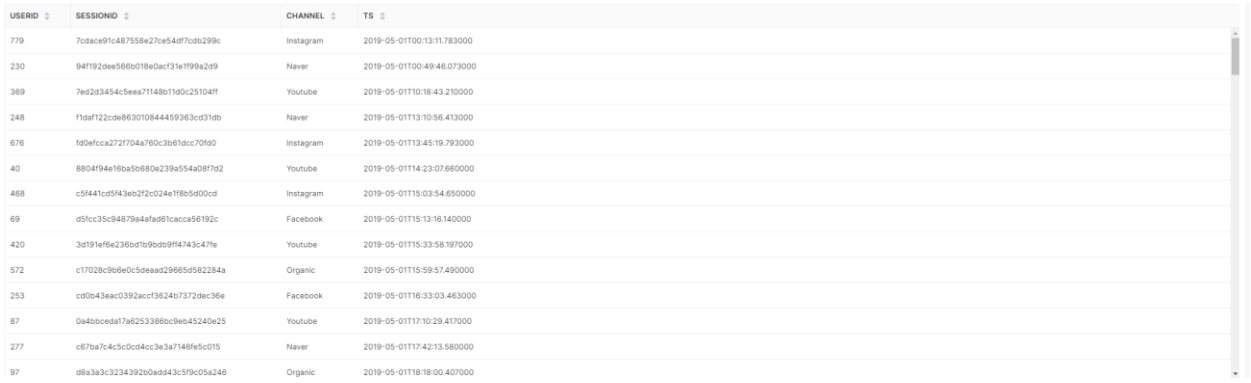
Capture the screenshot of this DAG's detailed page from the Web UI (#2)



Set up your Preset account or Docker Superset environmentLinks to an external site. (+2pt)

This includes setting up Snowflake connection and import session_summary from step 2

Capture the screenshot of your Datasets (#3)



USERID	SESSIONID	CHANNEL	TS
779	7cdace91c487558e27ce54df7cdb299c	Instagram	2019-05-01T00:13:11.783000
230	94f192dee566b018e0acf31e1f99a2d9	Naver	2019-05-01T00:49:46.073000
369	7ed2d3454c5eea71148b11d0c25104ff	Youtube	2019-05-01T10:18:43.210000
248	f1daf122cde863010844459363cd31db	Naver	2019-05-01T13:10:56.413000
676	fd0efcca272f704a760c3b61dcc70fd0	Instagram	2019-05-01T13:45:19.793000
40	8804f94e16ba5b680e239a554a08f7d2	Youtube	2019-05-01T14:23:07.660000
468	c5f441cd5f43eb2f2c024e1f8b5d00cd	Instagram	2019-05-01T15:03:54.650000
69	d5fcc35c94879e4afad61cacca56192c	Facebook	2019-05-01T15:13:16.140000
420	3d191ef6e236bd1b9bdb9ff4743c47fe	Youtube	2019-05-01T15:33:58.197000
572	c17028c9b6e0c5deaad29665d582284a	Organic	2019-05-01T15:59:57.490000
253	cd0b43eac0392accf3624b7372dec36e	Facebook	2019-05-01T16:33:03.463000
87	0a4bbceda17a6253386bc9eb45240e25	Youtube	2019-05-01T17:10:29.417000
277	c67ba7c4c5c0cd4cc3e3a7146fe5c015	Naver	2019-05-01T17:42:13.580000
97	d8a3a3c3234392b0add43c5f9c05a246	Organic	2019-05-01T18:18:00.407000

Create your WAU chart (+2pt)

Make sure you rename the metrics field to WAU

Capture the screenshot of the chart (#4)