Case Study:

We’re working with the Growth analytics team to improve how we attribute user sign ups to our acquisition channels and we want to use custom rules to define which channel should be awarded the credit for a given conversion. We will want to run this attribution model on a daily basis to track our acquisition channel results and we will also continue iterating on the model logic over time.

Data Modeling & Approach:

Rule-based attribution model helps in this use case to define the rules and logic for awarding credit to different acquisition channels based on specific criteria (PAID click, Impression, Organic), which includes below.

- Identify Attribution Factors – includes first-touch, last-touch, time decay consideration
- Defining Attribution rules – determine how credit should be awarded based on the identified factors
- Assign credit based on rules – Apply the attribution rules to each conversion or user interaction
- Aggregate attribution data –calculate the total credit awarded to each acquisition channel over a given rule

Why Rule-based attribution model:

- Flexible in defining the attribution rules. we can adjust/add the rules based on your specific business requirements and goals
- We can define rules that align with our understanding of user behavior and marketing strategies

Attribution rules:

✓ Paid Click: If a conversion event occurs within 3 hours of a Paid Click session, the Paid Click session will receive 100% attribution credit. It cannot be hijacked by any other session.

✓ Paid Impression: If a conversion event occurs within 1 hour of a Paid Impression session, the Paid Impression session will receive 100% attribution credit. It cannot be hijacked by any other session.

✓ Organic Click: If a conversion event occurs within 12 hours of an Organic Click session and there are no intervening Paid Click or Paid Impression sessions, the Organic Click session will receive 100% attribution credit. However, if there is a Paid Click or Paid Impression session within the 12-hour window, the credit will be attributed to the Paid session.

✓ Direct: If a user signs up without any live session (Paid or Organic), and the medium is "Direct," the Direct channel will receive 100% attribution credit.

✓ Others: If a user signs up without any live session (Paid or Organic) and the medium is not "Direct," the Others channel will receive 100% attribution credit.

Some user scenarios:

Scenario#1:
User A has a Paid Impression session at 10:00 AM.
User A has a Paid Click session at 11:00 AM.
User A converts at 11:30 AM.
In this scenario, the attribution would be as follows:

First touch: Paid Impression (within 1 hour)
Last touch: Paid Click (within 3 hours)
Attribution: 100% credit to the Paid Click channel, as it is the last touchpoint within the allowed time frame.

Scenario#2:
User B has an Organic Click session at 9:00 AM.
User B has a Paid Impression session at 10:00 AM.
User B converts at 11:30 AM.
In this scenario, the attribution would be as follows:

First touch: Organic Click (within 12 hours)
Last touch: Paid Impression (within 1 hour)
Attribution: 100% credit to the Organic channel, as it is the last touchpoint within the allowed time frame.

Scenario#3:
User C has an Organic Click session at 9:00 AM.
User C has a Paid Click session at 10:30 AM.
User C converts at 12:00 PM.
In this scenario, the attribution would be as follows:

First touch: Organic Click (within 12 hours)
Last touch: Paid Click (within 3 hours)
Attribution: 100% credit to the Paid Click channel, as it is the last touchpoint within the allowed time frame.

Scenario#4:
User A has a Paid Impression session at 9:00 AM.
User A has a Paid Click session at 10:30 AM.
User A converts at 12:00 PM.
In this scenario, the attribution would be as follows:

First touch: Paid Impression (within 1 hour)
Last touch: Paid Click (within 3 hours)
Attribution: 100% credit to the Paid Click channel, as it is the last touchpoint within the allowed time frame

Scenario#5:
User C receives an email campaign at 9:00 AM.

User C has a Paid Click session at 10:30 AM.
User C converts at 11:45 AM.
In this scenario, the attribution would be as follows:

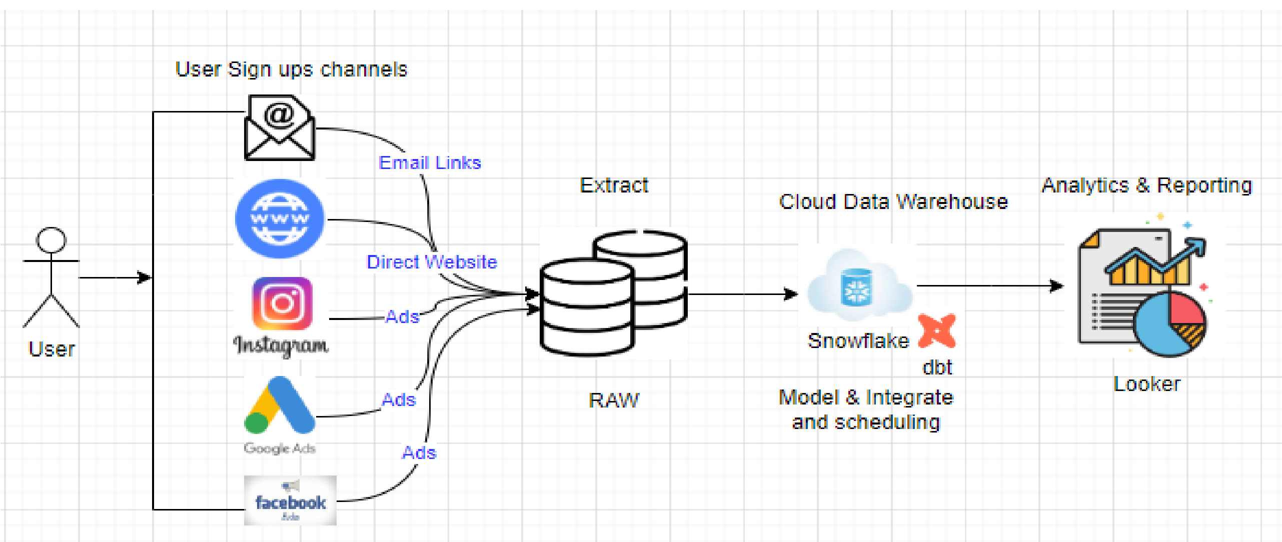
First touch: Paid Click (within 3 hours)
Last touch: Email (no live session)
Attribution: 100% credit to the Paid Click channel, as it is the first touchpoint within the allowed time frame

<u>Scenario#6:</u>

User D does not have any live sessions (paid or organic).
User D converts at 2:00 PM.
In this scenario, since User D does not have any live sessions, the attribution would be:

Attribution: If the medium is "Direct", then credit is assigned to "Direct". Otherwise, credit is assigned to "Others".

High level design & implementation:



<u>Pre-requisites:</u>

1. Create warehouse and database in Snowflake
    create warehouse vkd_test_wh;
    create database vkd_analytics_db
    create database raw
    create schema raw

2. Create raw tables for sessions and coversions:

```
create or replace TABLE RAW.RAW.CONVERSIONS (
        USER_ID NUMBER,
        REGISTRATION_TIME TIMESTAMP
);

create or replace TABLE RAW.RAW.SESSIONS (
        USER_ID NUMBER(38,0),
        TIME_STARTED TIMESTAMP_NTZ(9),
        MEDIUM VARCHAR(16777216)
);
```

3. Load sessions and conversions raw files:

   Note: Due to limitations in my Snowflake trial version, I was unable to load the complete data from the provided CSV files into the tables. As a result, I only processed a few records for testing purposes. Consequently, there are only a few matching records between the two tables based on the user_id column. To cover various scenarios for the attribution model, I have created test data to simulate different situations. To provide you with the necessary files, I have prepared a .zip file that includes the test data and the corresponding dbt project.

I have categories medium into three bucket Paid (Click or Impression) and Organic click as below
Paid click:

- PAID SOCIAL
- PAID SEARCH

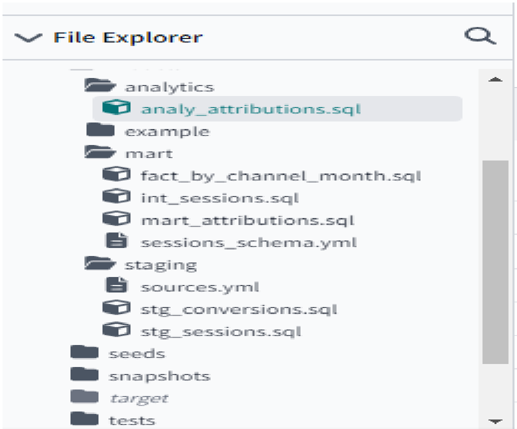
Paid impression:

- IMPRESSION

Organic click:

- REFERRAL
- ORGANIC SEARCH
- DIRECT
- MARKETPLACE
- INVITES
- PRIVATE_BOARD
- OTHER
- MOBILE_POPUP
- SSO
- SOCIAL
- DIRECTORIES
- MAIL

NOTE: sharing sessions.csv and conversions.csv dbt project structure:

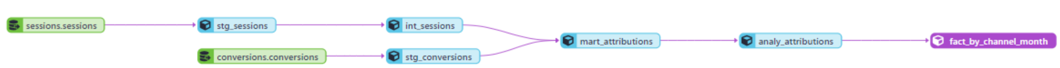


Data lineage:

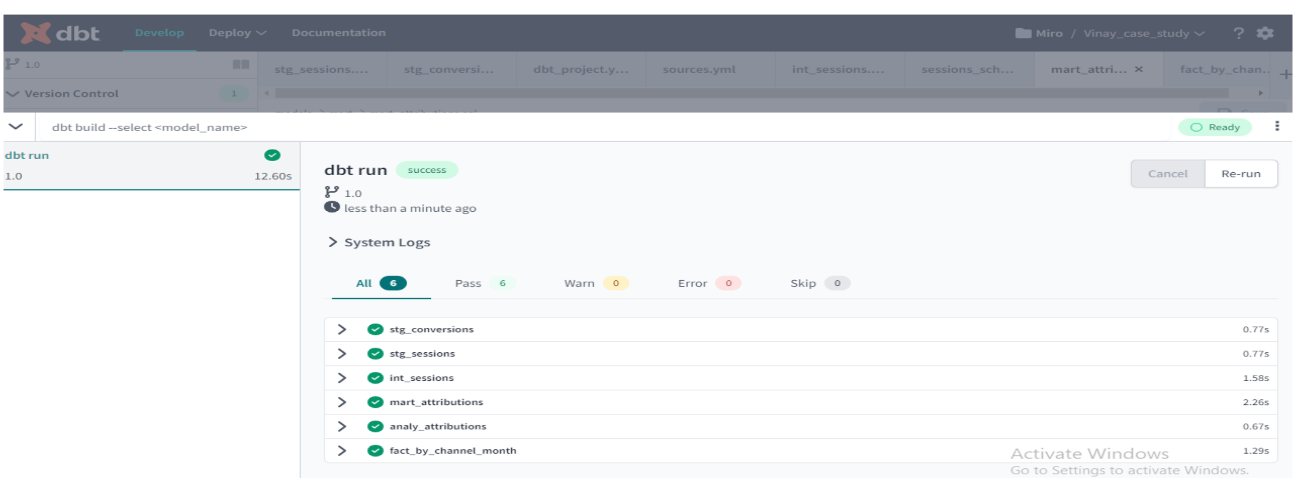
- The data flow starts by moving raw data to staging tables. next, transformations are applied to the int_sessions table, incorporating necessary rules and logic. These transformations ensure data consistency and structure.

- Once transformed, the mart_attributions view is created, implementing attribution rules such as lifespan limits, session hijacking prevention, and assignment of Direct or Others for sign-ups without sessions. Finally, the analy_attributions are generated for the users.

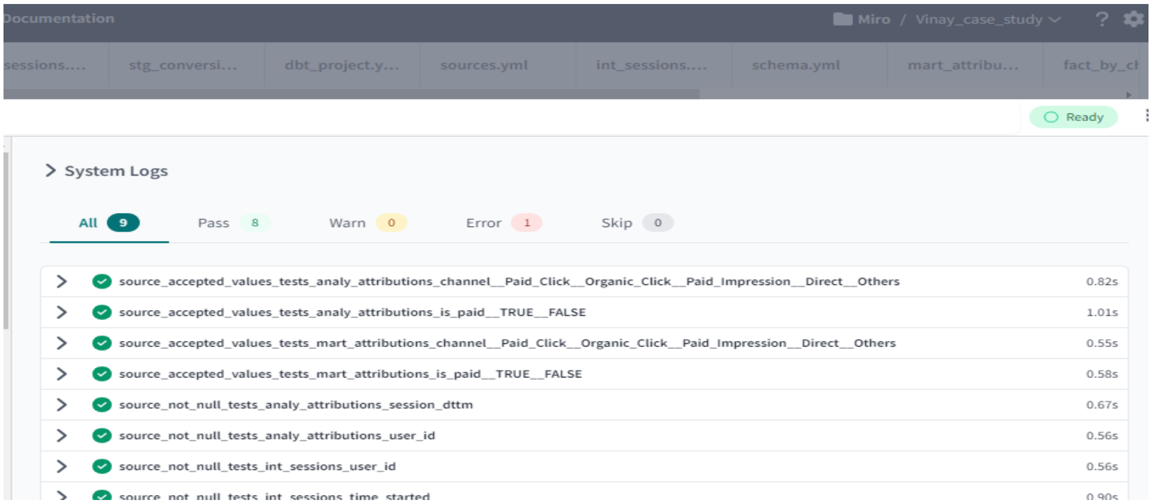
- This data flow, along with transformations and rule implementation, ensures accurate and meaningful attributions for analysis and reporting purposes



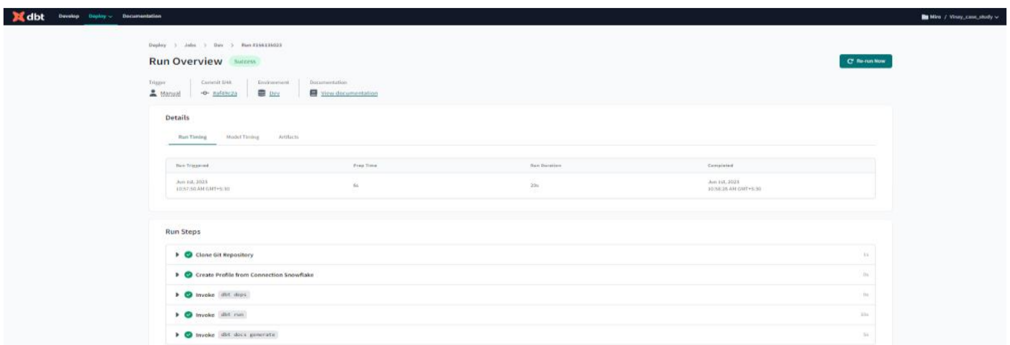
dbt run:



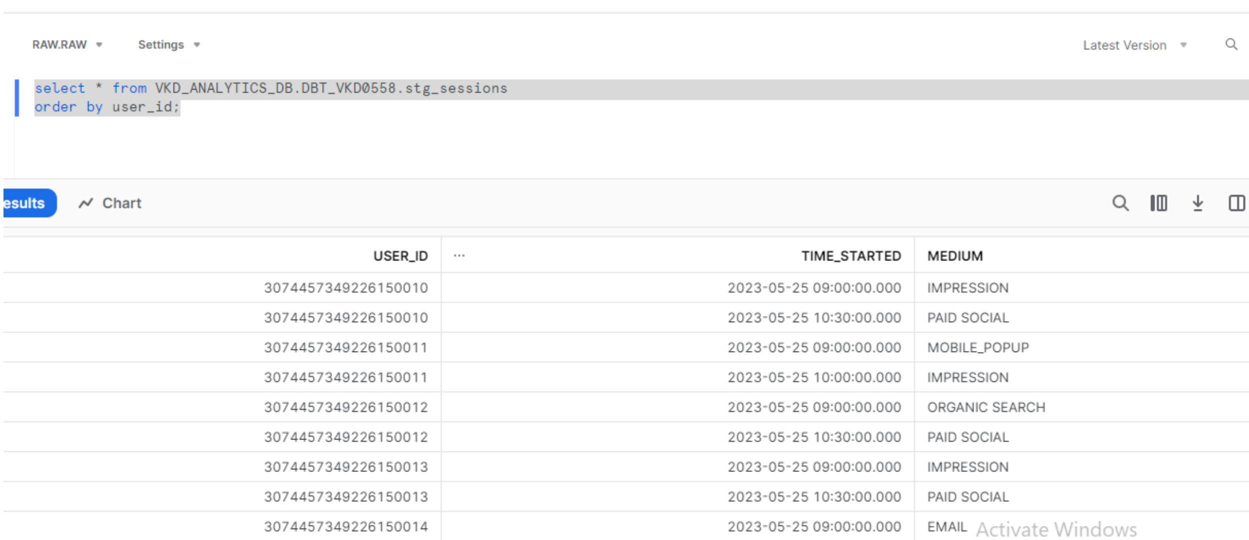
Tests run:



Job schedule:



stg_sessions sample records:



stg_conversions sample records:

RAW.RAW Settings Latest Version

```
select * from VKD_ANALYTICS_DB.DBT_VKD0558.stg_conversions
order by user_id;
```

Results Chart

USER_ID	REGISTRATION_TIME
3074457349226150010	2023-05-25 12:00:00.000
3074457349226150011	2023-05-25 11:30:00.000
3074457349226150012	2023-05-25 12:00:00.000
3074457349226150013	2023-05-25 12:00:00.000
3074457349226150014	2023-05-25 11:45:00.000
3074457349226150015	2023-04-20 10:00:00.000
3074457349226150016	2023-04-20 11:30:00.000
3074457349226150017	2023-04-20 12:00:00.000

analy_attributions output after rules:

ACCOUNTADMIN • VKD_TEST_WH Share

RAW.RAW Settings Latest Version Draft

```
select * from VKD_ANALYTICS_DB.DBT_VKD0558.analy_attributions
order by user_id;
```

Results Chart

	USER_ID	SESSION_DTTM	REGISTRATION_DTTM	MEDIUM	CHANNEL	IS_PAID
1	3074457349226150010	2023-05-25 09:00:00.000	2023-05-25 12:00:00.000	IMPRESSION	Paid Impression	FALSE
2	3074457349226150010	2023-05-25 10:30:00.000	2023-05-25 12:00:00.000	PAID SOCIAL	Paid Click	TRUE
3	3074457349226150011	2023-05-25 09:00:00.000	2023-05-25 11:30:00.000	MOBILE_POPUP	Organic Click	TRUE
4	3074457349226150011	2023-05-25 10:00:00.000	2023-05-25 11:30:00.000	IMPRESSION	Paid Impression	FALSE
5	3074457349226150012	2023-05-25 09:00:00.000	2023-05-25 12:00:00.000	ORGANIC SEARCH	Organic Click	FALSE
6	3074457349226150012	2023-05-25 10:30:00.000	2023-05-25 12:00:00.000	PAID SOCIAL	Paid Click	TRUE
7	3074457349226150013	2023-05-25 09:00:00.000	2023-05-25 12:00:00.000	IMPRESSION	Paid Impression	FALSE
8	3074457349226150013	2023-05-25 10:30:00.000	2023-05-25 12:00:00.000	PAID SOCIAL	Paid Click	TRUE
9	3074457349226150014	2023-05-25 09:00:00.000	2023-05-25 11:45:00.000	EMAIL	Organic Click	FALSE

The results of the attribution by channel and by month

```
with
   attributions as (select * from {{ ref("analy_attributions") }})
SELECT
  channel,
  TO_CHAR(DATE_TRUNC('month', REGISTRATION_DTTM), 'Mon') AS attribution_month,
  COUNT(DISTINCT user_id) AS attributed_users,
  COUNT(CASE WHEN is_paid = 'TRUE' THEN 1 END) AS attributed_conversions
FROM attributions
GROUP BY channel, attribution_month
ORDER BY attribution_month, channel;
```

Preview Selection | Compile Selection | Build | Format | Results | Compiled Code | Lineage

4.23s | Returned 12 rows. Downloa

CHANNEL	ATTRIBUTION_MONTH	ATTRIBUTED_USERS	ATTRIBUTED_CONVERSIONS
Others	Apr	1	1
Paid Click	Apr	1	1
Paid Impression	Apr	1	0
Paid Click	Feb	1	1
Organic Click	Jan	1	1
Paid Impression	Mar	1	1
Organic Click	May	2	1
Others	May	1	0
Paid Click	May	4	4
Paid Impression	May	3	0

The tests used to validate the correctness and completeness of the data

Generic test for unique, not null, accepted values, I have **schema.yml** file under mart folder

```yaml
version: 2

sources:
 - name: tests
   schema: dbt_vkd0558
   database: vkd_analytics_db
   tables:
     - name: int_sessions
       columns:
         - name: user_id
           description: "Not null user_id check int_sessions table"
           tests:
             - not_null
         - name: time_started
           tests:
             - not_null
     - name: mart_attributions
       columns:
         - name: is_paid
           description: "accepted values"
           tests:
             - accepted_values:
                 values: ['TRUE', 'FALSE']
         - name: channel
           description: "accepted values"
           tests:
             - accepted_values:
                 values: ['Paid Click', 'Organic Click', 'Paid Impression', 'Direct', 'Others']
```

```yaml
  - name: analy_attributions
    columns:
      - name: user_id
        description: "Not null user_id check int_sessions table"
        tests:
          - not_null
      - name: session_dttm
        tests:
          - not_null
      - name: is_paid
        description: "accepted values"
        tests:
          - accepted_values:
              values: ['TRUE', 'FALSE']
      - name: channel
        description: "accepted values"
        tests:
          - accepted_values:
              values: ['Paid Click', 'Organic Click', 'Paid Impression', 'Direct', 'Others']
```

Data Quality checks:

```sql
-- data counts between stages ( raw - staging)
        select count(*) from raw.conversions
        select count(*) from VKD_ANALYTICS_DB.DBT_VKD0558.stg_conversions
        select count(*) from raw.sessions
        select count(*) from VKD_ANALYTICS_DB.DBT_VKD0558.stg_sessions

-- staging to mart
        select count(*)
        from VKD_ANALYTICS_DB.DBT_VKD0558.stg_conversions c
            left join VKD_ANALYTICS_DB.DBT_VKD0558.int_sessions s on c.user_id = s.user_id
            where c.registration_time >= s.time_started

        select count(*) from VKD_ANALYTICS_DB.DBT_VKD0558.mart_attributions;

-- mart to analytics
        select count(*) from VKD_ANALYTICS_DB.DBT_VKD0558.mart_attributions;
        select count(*) from VKD_ANALYTICS_DB.DBT_VKD0558.ANALY_ATTRIBUTIONS;
-- check if any signups before session start
select count(*) from VKD_ANALYTICS_DB.DBT_VKD0558.ANALY_ATTRIBUTIONS
where registration_dttm<session_dttm

-- test_mart_attributions.sql
```

```sql
-- Test that the values in the `channels` column are valid
SELECT COUNT(*)
FROM {{ ref('analy_attributions') }}
WHERE channel NOT IN ('Paid Click', 'Organic Click', 'Paid Impression', 'Direct', 'Others')

UNION ALL

-- Test that the values in the `registration_dttm` column are valid
SELECT COUNT(*)
FROM {{ ref('analy_attributions') }}
WHERE registration_dttm <= session_dttm

UNION ALL

-- Test that the life span of Paid Click sessions is within 3 hours
SELECT COUNT(*)
FROM {{ ref('analy_attributions') }}
WHERE channel = 'Paid Click'
  AND TIMESTAMPDIFF(HOUR, session_dttm, registration_dttm) > 3

UNION ALL

-- Test that the life span of Paid Impression sessions is within 1 hour
SELECT COUNT(*)
FROM {{ ref('analy_attributions') }}
WHERE channel = 'Paid Impression'
  AND TIMESTAMPDIFF(HOUR, session_dttm, registration_dttm) > 1

UNION ALL

-- Test that the life span of Organic Click sessions is within 12 hours

SELECT COUNT(*)
FROM {{ ref('analy_attributions') }}
WHERE channel = 'Organic Click'
  AND TIMESTAMPDIFF(HOUR, session_dttm, registration_dttm) > 12

UNION ALL

-- Test that Paid sessions are not hijacked by other sessions during their life span
SELECT COUNT(*)
FROM {{ ref('analy_attributions') }} AS a
WHERE channel IN ('Paid Click', 'Paid Impression')
  AND EXISTS (
```

```sql
   SELECT 1
   FROM {{ ref('analy_attributions') }} AS b
   WHERE b.user_id = a.user_id
    AND b.channel != a.channel
    AND b.session_dttm <= DATEADD('hour', 3, a.session_dttm)
    AND b.session_dttm >= a.session_dttm
 ) AND IS_PAID='TRUE'

UNION ALL

SELECT COUNT(*)
FROM {{ ref('analy_attributions') }} AS a
LEFT JOIN {{ ref('analy_attributions') }} AS b
   ON a.user_id = b.user_id
   AND (
      (b.channel IN ('Paid Click', 'Paid Impression') AND b.session_dttm <= DATEADD('hour', 3,
a.session_dttm))
      OR (b.channel = 'Organic Click' AND b.session_dttm <= DATEADD('hour', 12, a.session_dttm))
   )
```

If this problem was given without rules and examples, what approach would be to conduct requirements discovery

- Clarify the key goals, desired outcomes, and the specific problem the attribution model aims to solve.
- Engage with analytics to gather their perspectives on attribution, expectations, and factors they consider important in attributing conversions.
- Understand the available data sources involved in attribution to identify relevant data points
- Evolve different attribution models and methodologies, considering how each aligns with business objectives and inputs & Identify key metrics for attribution, document the chosen attribution model(s), data sources, variables, business rules, and reporting formats

Any other comments to either explain work, thought process, or engineering process

In approaching this use case, my main goal was to understand the problem of attribution modeling and provide a solution that aligns with the business objectives considering below aspect of engineering practices.

- Modular and Extensible Architecture
- Efficient Rule Evaluation
- Robust Data Pipeline

Any other considerations would have regarding how this data could be used for reporting or analytics and caveats associated with it

- Understand the different attribution models used in analysis, such as first touch, last touch, linear etc. and consider the granularity of your attribution data
- Accuracy and completeness of the attribution data i.e. ensure that the data is properly captured and processed without any missing or duplicate records
- Consider the possibility of channel overlap or interaction effects, users interact with multiple channels before converting, and the impact of each channel in terms of assigning credit

How would model this attribution data into a broader data warehouse model

To model attribution data within a broader data warehouse model, we can follow a dimensional modeling approach. Here's a high-level overview of how we can model attribution data

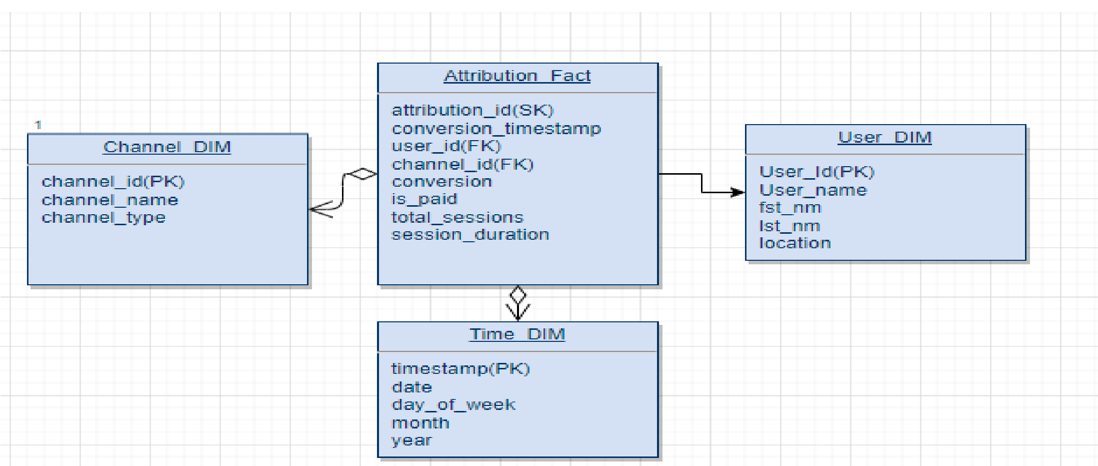
Fact Table: Create a fact table that captures the attribution events or conversions. This table would contain the key metrics related to the attribution, such as conversion timestamp, attribution channel, attributed user, and any other relevant metrics. Each row in the fact table represents a single attribution event

Dimension Tables: Create dimension tables to provide additional context and details about the attribution data. Some possible dimension tables could include:

User Dimension: Contains information about the users, such as user ID, demographics, and behavioral attributes.

Channel Dimension: Provides details about the different channels through which the attribution occurs, including channel ID, channel name, and channel type.

Time Dimension: Captures various time-related attributes like date, day of the week, month, etc. This dimension allows for time-based analysis of attribution data



Disclaimer:

The use case presented is based on trial versions of dbt and Snowflake. Although I have made diligent efforts to execute all scenarios successfully, it is important to acknowledge the limitations and dependencies that have influenced the outcomes.

Please take note of the following considerations:

- Dbt Utilities: Due to the unavailability of certain dbt utilities in the trial version, a few tests were unable to run successfully.
- File Size Limitation: Unfortunately, I encountered challenges when attempting to load the "sessions.csv" and "conversions.csv" files into Snowflake. These limitations, related to the size of the files

Despite these challenges, I have strived to provide a valuable demonstration data modeling & implementation and building pipeline, testing use cases etc…