

# ID5 Match Preview - Snowflake Native Application

09/14/2026 12:19 pm EDT

## Set Up the ID5 Match Preview Native App in Snowflake

This guide walks you through running a free match test against the ID5 identity universe using the **ID5 Match Preview** native app in Snowflake. The app is publicly listed on the Snowflake Marketplace, no prior access request needed. You provide your first-party identifiers and the app returns a coverage report showing how much of that input we can resolve within the ID5 identity graph. The report breaks down match rates and unique match counts against ID5 IDs, MAIDs, CTV IDs, and partner cookies, so you can see exactly where your data resolves before committing to activation.

### Prerequisites

- An active Snowflake account (any edition), Before proceeding, a user with the ORGADMIN role (or higher) must accept the Snowflake Consumer Terms of Service — see [Snowflake's instructions](#).
- A table in Snowflake containing your HEMs formatted as described below available in AWS / eu-central-1
- A warehouse (compute) available to run queries — medium size is recommended for most use cases

## Step 1: Prepare Your HEM Table

Create a table in your Snowflake account with the following structure:

| Column  | Type    | Description                                                       |
|---------|---------|-------------------------------------------------------------------|
| id      | VARCHAR | The (hashed) email (SHA-256 hash of the lowercased email address) |
| id_type | VARCHAR | <code>HASHED_EMAIL</code> or <code>EMAIL</code>                   |

### SQL Example:

```
CREATE TABLE my_database.my_schema.hem_input (  
  id VARCHAR NOT NULL,  
  id_type VARCHAR NOT NULL  
);  
  
INSERT INTO my_database.my_schema.hem_input VALUES  
  ('b642b4217b34b1e8d3bd915fc65c4452...', 'HASHED_EMAIL'),  
  ('e99a18c428cb38d5f260853678922e03...', 'HASHED_EMAIL');
```

### Important formatting notes:

- Emails must be lowercased before hashing.
- The hashing algorithm must be SHA-256.
- If emails are not pre-hashed, we hash them before onboarding — please ensure they are normalized first, as described in [Normalizing Emails Prior to Hashing](#).
- Both the `id` and `id_type` columns must be `NOT NULL`.

- Duplicate rows are handled automatically by the app — no need to deduplicate beforehand.

## Step 2: Install the Match Preview App



The [ID5 listing terms](#) are accepted upon installing the application from the Snowflake Marketplace.

1. In Snowflake, navigate to **Marketplace** in the left-hand menu.
2. Search for **"ID5 Match Preview."**
3. Select **ID5 Match Preview** from the results.
4. Click **Get**.
5. Accept the permissions prompt — this allows the app to access only the resources you explicitly grant in the next step.
6. Optionally rename the app (e.g. `MATCH_PREVIEW`) and confirm installation.
7. Once installed, the app will appear under **My Apps** — installation takes less than 2 minutes.

The screenshot shows the 'My apps' section in Snowflake. A search bar at the top contains 'Search apps by name or description'. Below it, a table lists installed applications. The 'ID5 Match Preview' app is highlighted with a yellow box. It shows a status of 'New' and was installed '25 seconds ago' by 'ACCOUNTADMIN'.

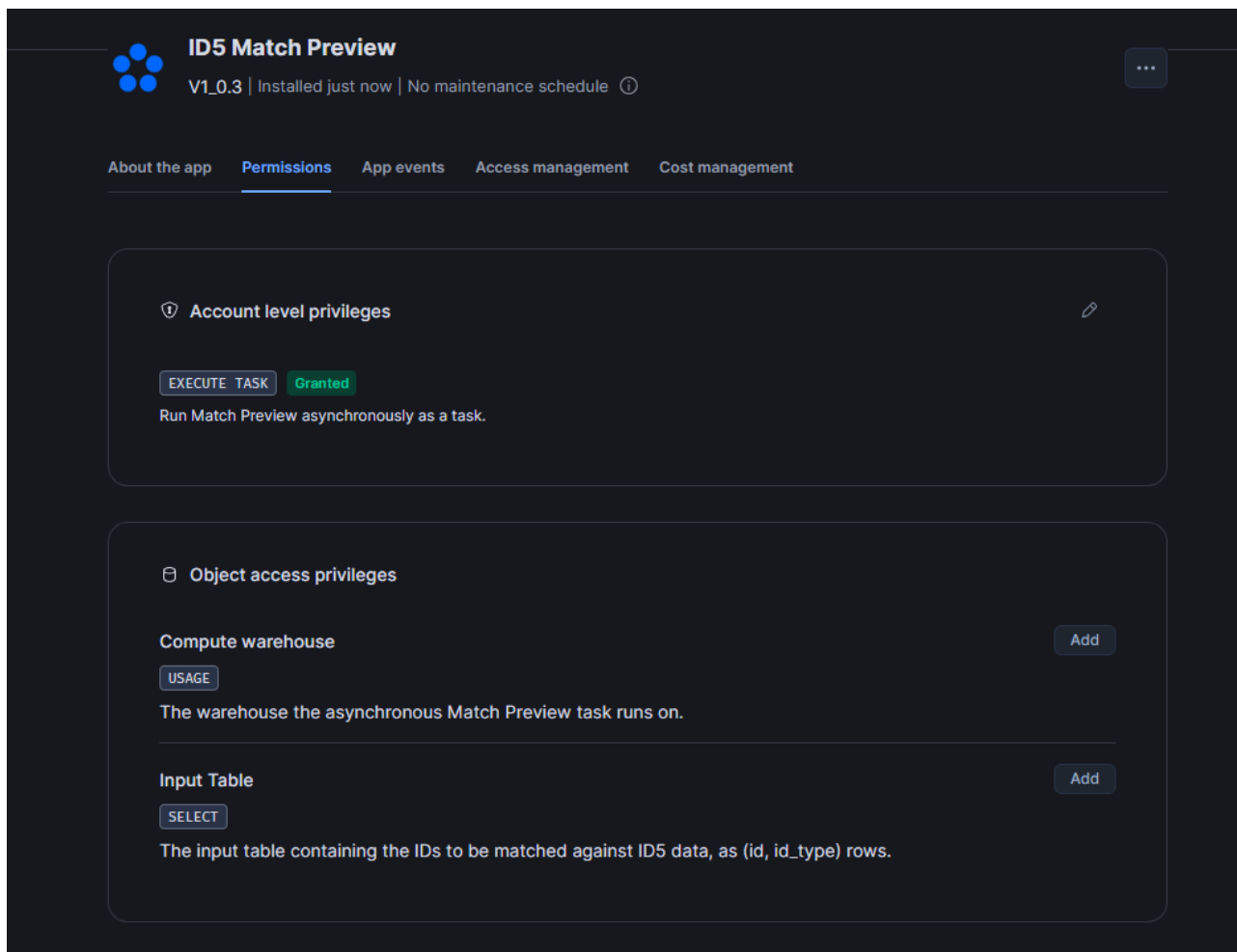
| TITLE                                                                                                                                                                                                                              | LAST MODIFIED  | OWNER ROLE   | STATUS |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|--------------|--------|
| <b>ID5 Data Onboarding Audience Translation (Priv)</b><br>Powered by the Adaptive Identity technology, the ID5 Data Onboarding and Activation application resolves your email-keyed segments to the ID5 ID—the most adopted cro... | 4 hours ago    | ACCOUNTADMIN | New    |
| <b>ID5 Match Preview</b><br>The ID5 Match Preview application calculates how much of your first-party data can be onboarded and resolved to digital identifiers, so you can estimate the campaign on-...                           | 25 seconds ago | ACCOUNTADMIN | New    |

## Step 3: Set Up Permissions (References)

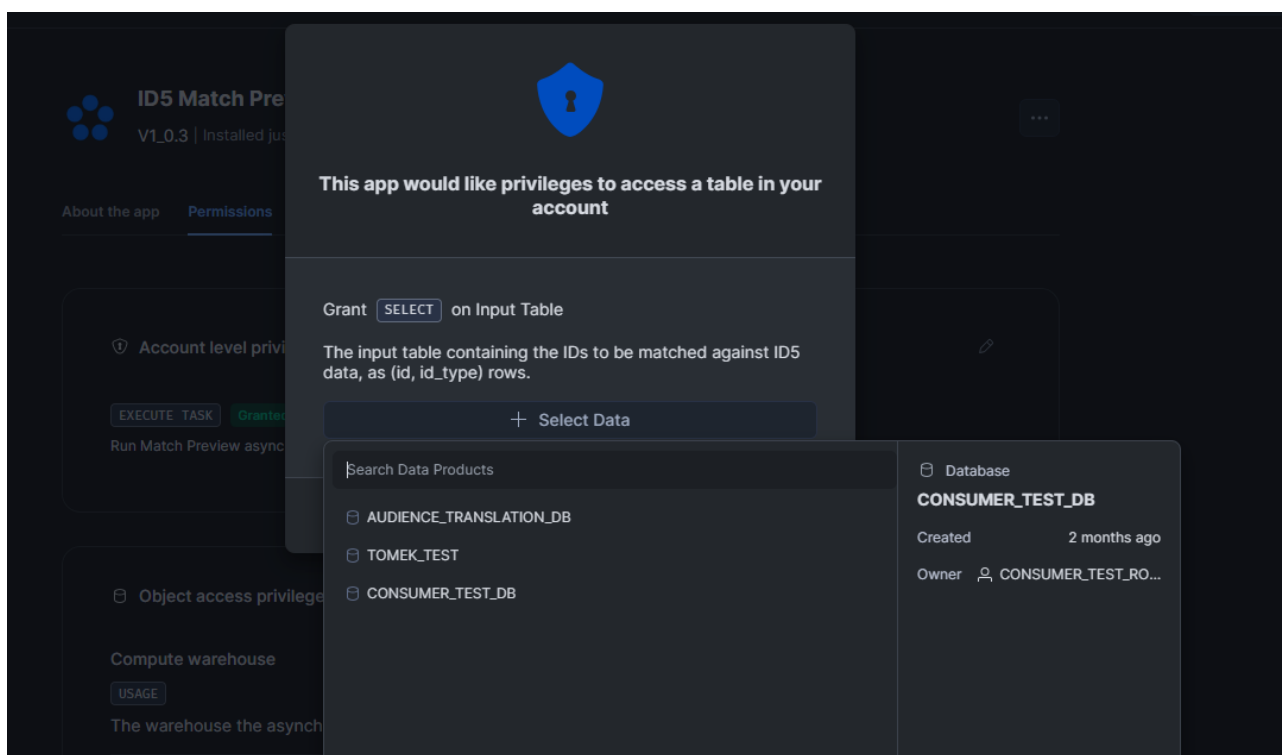


This step is required for privacy reasons — the app can only access resources you explicitly allow. It only requests read (SELECT) access to your input table; your data is never retained inside the app.

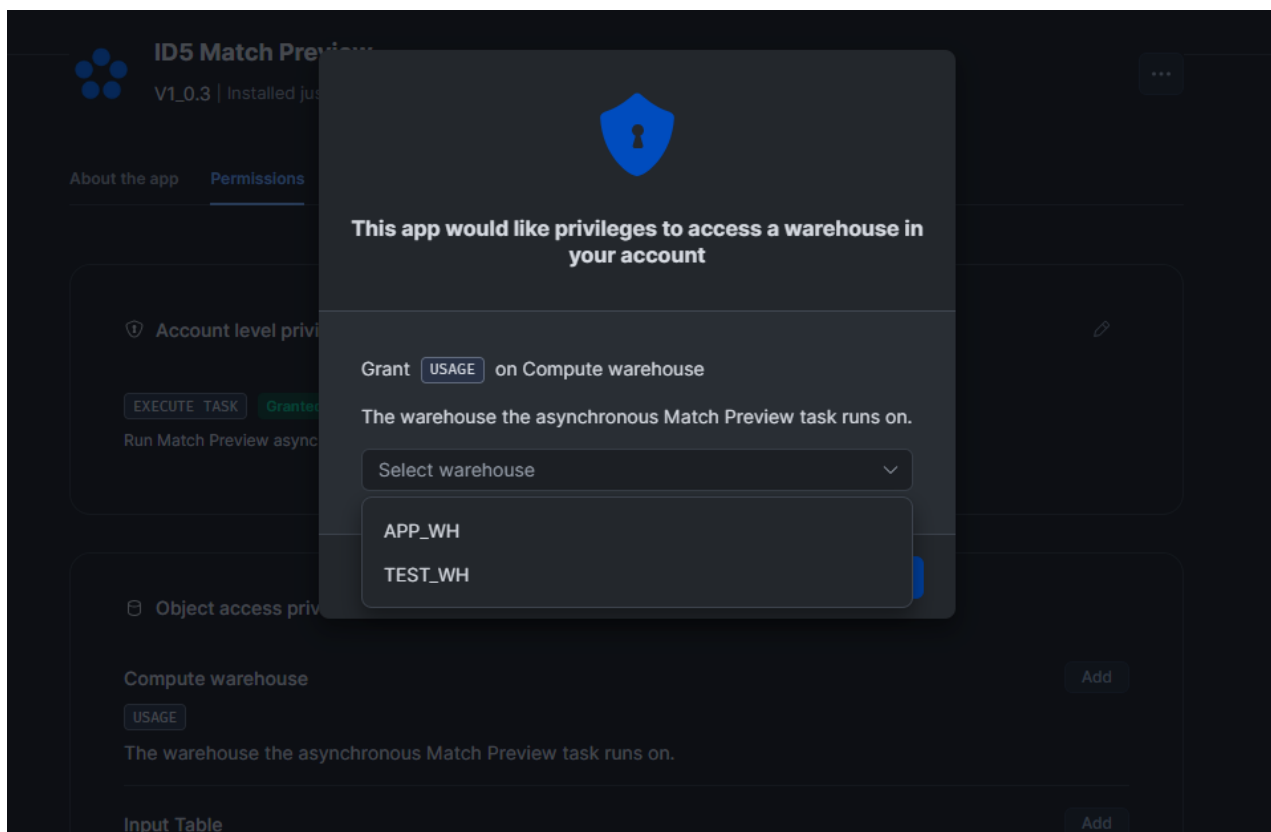
1. After installation, you need to grant the app access to your warehouse and input table. Go to **Apps > Select Match Preview App > Permissions**:



2. Grant the application access to your data by adding your table name to the **Input Table**. This table should contain the input signals you want to join on (e.g. your SHA-256 normalized hashed emails), to get a high-level understanding of match rates via the app.



3. Choose the **compute warehouse** you want to use. Medium sized is recommended for most use cases.



A warehouse is a specification of compute resources used to perform operations, governed by Snowflake's sizing and pricing rules. Warehouses auto-suspend after a period of inactivity. See Snowflake's [warehouse documentation](#) for more on creating and using them.

Alternatively run the following in a Snowflake worksheet:

```
-- Grant the app access to a warehouse
CALL MATCH_PREVIEW.core.register_reference(
  'COMPUTE_WAREHOUSE', 'ADD',
  SYSTEM$REFERENCE('WAREHOUSE', 'MY_WAREHOUSE', 'PERSISTENT', 'USAGE'));
-- replace MY_WAREHOUSE with your warehouse name

-- Grant the app read access to your HEM table
CALL MATCH_PREVIEW.core.register_reference(
  'MATCH_PREVIEW_INPUT_REF', 'ADD',
  SYSTEM$REFERENCE('TABLE', 'MY_DATABASE.MY_SCHEMA.HEM_INPUT', 'PERSISTENT', 'SELECT'));
-- replace MY_DATABASE.MY_SCHEMA.HEM_INPUT with your table path
```

## Step 4 — Run a match test

Run the match procedure with your chosen parameters:

**SQL**

```
CALL MATCH_PREVIEW.core.match_preview(  
  'ID',      -- name of your id column  
  'ID_TYPE', -- name of your id_type column  
  ARRAY_CONSTRUCT('CURRENT', 'INDIVIDUAL') -- match methods (see options below)  
);
```

#### Match method options (at least one required):

| Match Method | Description                                                                     |
|--------------|---------------------------------------------------------------------------------|
| CURRENT      | Direct one-to-one HEM → ID5 ID mapping (last observed HEM to ID5 ID connection) |
| INDIVIDUAL   | Expands to all ID5 IDs linked to the same individual via the ID5 graph          |

The results combine all selected methods. The procedure runs asynchronously — it returns immediately with an execution ID and processes in the background. A few hundred thousand HEMs on a medium warehouse typically completes in a few minutes.

## Step 7 — Check Status

### SQL

```
CALL MATCH_PREVIEW.core.match_preview_status();
```

This returns the status of the most recent run: **SCHEDULED**, **COMPLETED**, or **FAILED**, along with timestamps and the execution ID.

## Step 8 — View Results

Once the status shows **COMPLETED**:

### SQL

```
SELECT * FROM MATCH_PREVIEW.core.match_preview_output;
```

The output returns the amount and percentage of unique HEMs that have been matched to at least one ID5 ID as well as the total matched ID5 IDs.

#### Example output:

```
{
  "matched_id5_ids": {
    "total_matched_id5_ids": 2000,
    "matched_input_hems": 1000,
    "matched_input_hems_perc": 50
  }
}
```

---

## Snowflake compute costs (Client-Side)

Running the match test incurs a small Snowflake compute cost on your side (not charged by ID5). Based on our testing, we recommend starting with a medium warehouse. If you encounter resource errors, increase the warehouse size.

---

## Data Sharing and Telemetry

To help us support you, the app shares aggregated telemetry with ID5. By accepting the app's permissions during installation, you agree to this sharing. The telemetry includes:

- Number of input HEMs and matched ID5 IDs
- Which match methods were used
- Execution duration and status
- Your Snowflake account and organisation identifier (to attribute the run to your account)

ID5 cannot see your individual HEM values. Your raw input data stays within your Snowflake account at all times and is never retained by the app. Telemetry data is kept for 14 days on the ID5 side.

---

## Troubleshooting

If the match preview does not return a result, you can check the following options to troubleshoot.

| Issue                          | What to check                                                                                                                                                                                                                                                       |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| App not visible in Marketplace | Search for "ID5 Match Preview" — if it still doesn't appear, check with your Snowflake admin that you have Marketplace access enabled                                                                                                                               |
| Job fails immediately          | <p>Check that the warehouse and input table references are correctly registered (Step 3)</p> <p>Check if your input data is stored and in <b>AWS / eu-central-1</b>. If not, ask your Snowflake representative to replicate the data to the target destination.</p> |

Low or zero matches

Verify emails are [lowercased and normalised according to ID5 requirements](#) before hashing, and that `id_type` is set to 'HEM'

Job takes a long time

Consider increasing the warehouse size

If you're stuck, share the execution ID with your ID5 account manager and we can check the error log on our side.

---