DR - 0004 - Create 'workspace' Schemas in each Snowflake Database

Status	ADT-177 - "Scratch" Schemas CLOSED PROPOSED
Owner	James Goodhand
Deciders	19 Oct 2020 - Decision Forum
Stakeholders	Snowflake users
Date Decided	
Due Date	
Outcome	
Information Level	*50%*

Decision Link	ADT-177 - "Scratch" Schemas CLOSED
Technical Story	
Other Links	

Actions

- [ ] James Goodhand - complete ADR - 0004 - Create 'workspace' Schemas in each Snowflake Database
- [x] James Goodhand - investigate whether a specific ownership / read & write configuration would enable users to only read / write to tables that they had created in the scratch area

Context and Problem Statement

User's do not have a clear space within which to create 'scratch' tables to work with. This makes it difficult for them to perform exploratory analysis of data that is outside of their known data area. They do not know what schemas / tables it is safe or that they have permissions to create within.

Decision Drivers & Background

1. Driver 1 - to have a safe place to play around with data without effecting production data / processes
2. Driver 2 - to know where you can and have permissions within which to create tables / data

Considered Options

1. Option 1 - Scratch schema area (transient tables)
2. Option 2 - Scratch schema area (normal tables)
3. Option 3 - Do nothing

Pros and Cons of the Options

[option 1]

Scratch schema area (transient tables) in each DB

- Good, because ...
  - Tables limited in their time travel period by being forced to be transient
  - Known and clear scratch area for users to work in
  - Temp tables can still be created in Scratch area
- Bad, because...
  - Potential for tables to get very large and runaway
  - Potential for data to be exposed to users who shouldn't be able to see it in these areas

Mitigations:

- Scan scratch areas for old tables which haven't been used in a while and auto-drop
- Scan scratch tables for table size and report on dashboard to highlight these users
- Ban PII data from scratch schema
- (option) Automatically cap tables at a certain size - auto drop rows beyond X GB
  - This could lead to potentially confusing behaviour for users
  - Maybe better to email these users automatically and ask them to delete via reporting

[option 2]

Scratch schema area (normal tables) in each DB

- Good, because ...
  - Known and clear scratch area for users to work in
  - Users can chose between temp, normal and transient tables in the scratch area
- Bad, because...
  - Potential for tables to get very large and runaway
  - Potential for data to be exposed to users who shouldn't be able to see it in these areas
  - Higher risk of large tables due to time travel as would rely on users to set as transient / temp

Mitigations:

- Scan scratch areas for old tables which haven't been used in a while and auto-drop
- Scan scratch tables for table size and report on dashboard to highlight these users

[option 3]

Do nothing

- Good, because ...
  - No implementation effort required
  - Reduced potential for run-away tables as it is harder to create them
- Bad, because...
  - Run-away tables are harder to find

Decision Outcome

Option 1 being recommended to Decision Forum.

Chosen option: "[option 1]", because [justification. e.g., only option, which meets k.o. criterion decision driver | which resolves force force | … | comes out best (see below)].

Positive Consequences

- Improvement of quality attribute satisfaction, follow-up decisions required...

Negative Consequences

- Compromising quality attribute, follow-up decsion required...