DASERA | snowflake

Data Lifecycle Security for Snowflake

Congratulations, You're Migrating to Snowflake

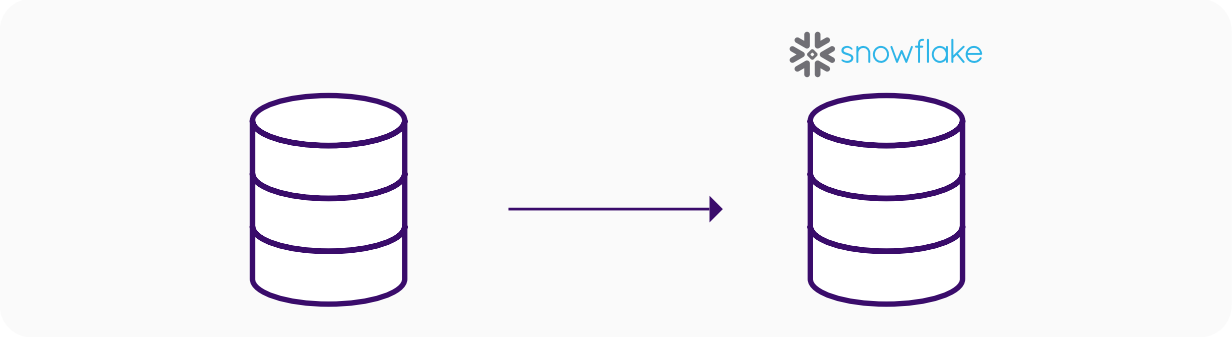
So your organization has made the decision to migrate to Snowflake. Or perhaps you've migrated to Snowflake already.

Chances are, in the past, you've had too many data sources and data stores, and it's tough to scale up and out. You've likely had performance and resource contention issues in the past, and you've had to limit the number of queries and number of users on your data warehouse. To distribute load, maybe you've created data marts for different departments, but you've probably found that data marts become silos of data, in addition to increasing your data management complexity and cost.

You've done your Snowflake POCs, and you're genuinely excited by the promise of Snowflake. You love the fact that Snowflake has completely divorced compute from storage. Divorcing compute from storage means you can create highly independent and infinitely scalable compute clusters, so when those marketing interns write incredibly slow queries, they won't bring the entire data warehouse down to a grinding halt. (And you can add more compute power to the Marketing cluster with the push of a button.) With completely independent storage, you now can truly have **one single source of truth** — i.e., you can eliminate all those data marts and data silos.

> **You love the fact that Snowflake has completely divorced compute from storage... You now can truly have one single source of truth.**

With Snowflake's infinite scale and one single source of truth, you can finally achieve the aspiration of democratizing data across the organization. You can give all your employees easy access to all the data they need to make better decisions faster.



But as Spiderman says, "With great power comes great responsibility." Migrating to Snowflake brings great power to the organization. But you're still responsible for protecting sensitive data throughout its lifecycle, even when it's in Snowflake.

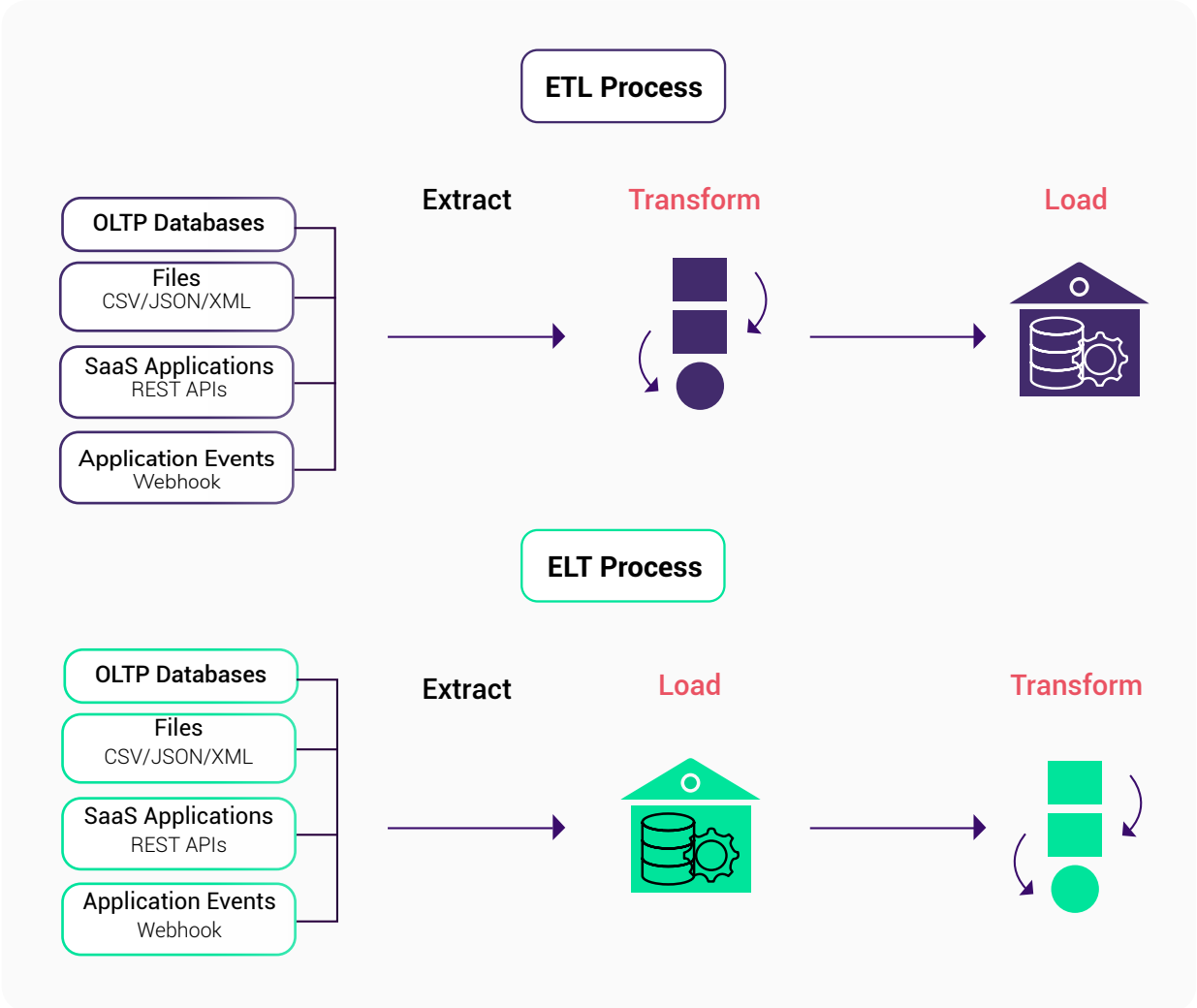
ETL versus ELT

Snowflake is more than just a data warehouse—it can also be a data lake combined with a data warehouse, or as Snowflake likes to say, a **data lake-house**.

Traditional data warehouses use an ETL *(Extract, Transform, and Load)* process, where data is *extracted* from OLTP production databases, *transformed* in a temporary staging area into a columnar format (to optimize the data for reads) and then *loaded* into the data warehouse, where the data resides.

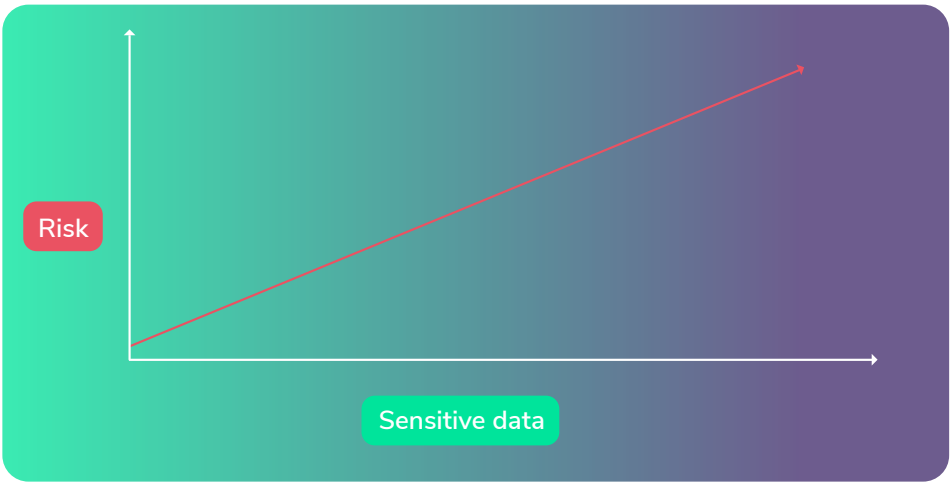
Snowflake borrows from data lake concepts and often leverages an ELT (*Extract, Load, and Transform*) process, where data is first *extracted* from OLTP production databases, stored directly in the data lake, and then when the data needs to be analyzed, it's *loaded* into the data warehouse and *transformed* as necessary.

The primary benefit of ELT is the infinite scalability of the data lake. Much more data can be stored in the data lake (i.e., Snowflake Data Cloud, AWS S3, Google Cloud Storage, or Azure Cloud Storage) and data can be loaded into the data warehouse on an as-needed basis. Furthermore, storing data in a data lake can be much cheaper (due to less frequent access) than loading data in the data warehouse, where frequent access to data is assumed and storage is priced accordingly.



But there's a downside to ELT. Before, with ETL, part of the transform step often includes removing sensitive data before the data is stored in the data warehouse. With ELT, because the transform step happens last, no sensitive data is removed.

So, if you're migrating to Snowflake and switching from ETL to ELT, you're probably going to have a lot more sensitive data stored in your data lake. More sensitive data being stored creates more risk that needs to be managed.



> **More sensitive data being stored creates more risk that needs to be managed.**

Privileges Galore in Snowflake

As part of your Snowflake migration, you'll need to carefully map all of your users to the right privileges and roles in Snowflake.

Snowflake has a veritable cornucopia of privileges to choose from, including user, role, resource monitor, virtual warehouse, integration, data exchange, stored procedure, and database/schema/table privileges.

With all those privileges, you'll have the power to create some very fine-grained privileges for all types of users.

But with that power comes several key questions:

- How will you know if all those privileges were created correctly?
- How will you know when privileges change?

- When privileges change, how will you know if they were changed correctly?
- How can you make sure the right conditions and time limits are enforced for conditional or time-bound privileges
- How will you detect over-privilege?
- How can you ensure that data usage complies with applicable regulations, including privacy regulations
- How can you ensure data is being used in accordance with opt-ins/opt-outs?
- How can you ensure data is being used in accordance with third-party DPAs?

Can't We Use Snowflake's Dynamic Data Masking to Protect Sensitive Data?

Snowflake has a great feature called Dynamic Data Masking, which is a column-level security feature that selectively masks plain-text data based on role or custom entitlements. Just mask sensitive data to everyone who shouldn't see sensitive data, and don't mask sensitive data to people who need to see sensitive data.

> **Among those with privilege, how can you ensure that data usage is compliant with all other regulations, including privacy regulations?**

This binary approach fails to accommodate the wide variety of data usage of your users. In one situation, an employee should be able to access sensitive data while in another situation, the same employee should not be able to access the same sensitive data. A few examples will help clarify this point.

A customer success representative will need access to sensitive data in order to update a customer's phone number, verify their email address, or look up an individual customer's support history.

But will that customer success rep need to see the credit card numbers of 10,000 customers at the same time? Probably not. Conversely, a marketing analyst may need the ability to generate segmented email mailing lists of 10,000 or even 100,000 customers on a regular basis.

However, should the same marketing analyst be allowed to look at the purchase history of his former girlfriend? Probably not. Legitimate use versus misuse of data depends on the context of data usage. Dynamic Data Masking is a blunt, context-free approach which falls short in the above scenarios and many others.

Data Sprawl and the Limits of Read-Only Access

In large Snowflake deployments with lots of employees (and other parties) accessing data, sensitive data is copied and moved to a variety of different locations. To prevent data sprawl, some companies think they can:

- Identify all sensitive data,
- Move it all into one "central location" within Snowflake, and
- Give employees read-only access to the sensitive data.

There are 2 major problems with the read-only access solution.

First, employees with read-only access **can still violate privacy.** Even if you give employees read-only access, you still have a major compliance challenge on your hands.

> **Even if you give employees read-only access, you still have a major compliance challenge on your hands.**

Second, read-only access is probably an unrealistic solution. Many users need the ability to create temporary and transient tables from queries, and then run additional queries on those temporary and transient tables. Data analysis can often be a multi-step process; limiting users to read-only access can significantly reduce their productivity and effectiveness.

The good news is, temporary tables exist only within a session and are then automatically dropped, so they don't contribute to data sprawl.

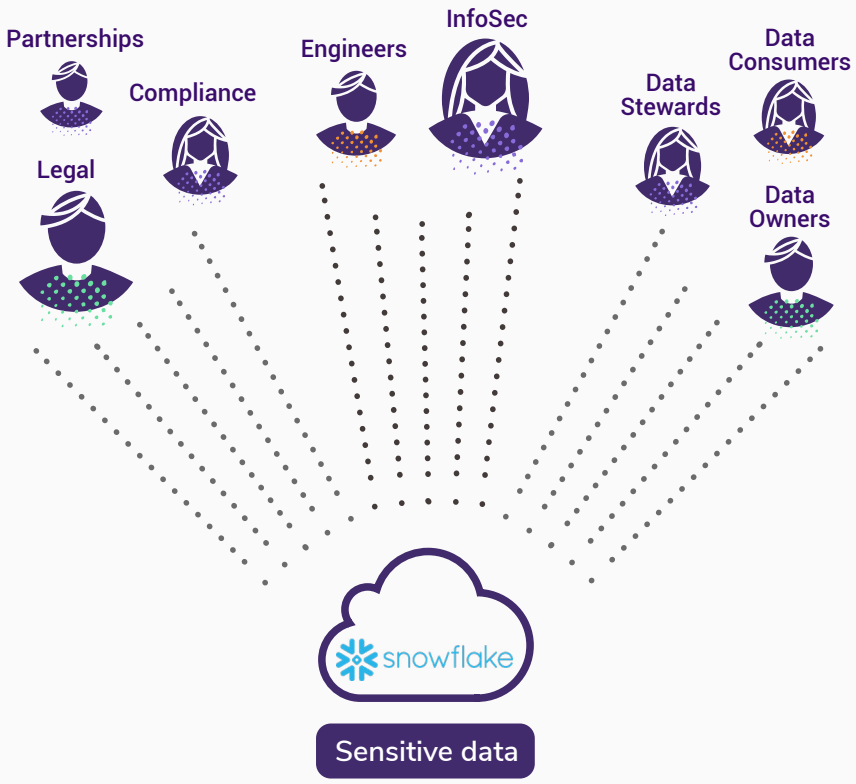
Tables in Snowflake

Temporary Tables	Transient Tables
✓ Exist only within a session	✗ Exist across sessions
✓ Are automatically dropped	✗ Need to be explicitly dropped
✓ Don't contribute to data sprawl	✗ Contribute to data sprawl

But transient tables are a different story. Transient tables persist beyond the end of a user session. In fact, they need to be explicitly dropped. If the person who created a transient table forgets to drop the table, you've basically got a de facto permanent table on your hands.

Transient tables are also available to all users with sufficient privilege. If a user has GRANT SELECT access to an entire database or schema, they have GRANT SELECT access to all transient tables in that database or schema as well.

So, the reality is, sensitive data can quickly leak out of the one "central location" in Snowflake. Unless monitored and managed properly, the number of users who have access to that sensitive data can start to inadvertently grow over time as well.



Protect Your Entire Snowflake Data Lifecycle with Dasera

Migrating to Snowflake is a great step forward. But you still need to protect your data lifecycle. As discussed above, you still need to:

- Identify and protect structured and semi-structured sensitive data in your Snowflake data lakehouse.
- Monitor privileges within Snowflake and detect when privileges change or when over-privilege occurs.
- Go beyond access control and monitor actual data usage among employees and partners with Snowflake access.

- Track data lineage in Snowflake — so derivative copies of data can retain all properties of parent/ancestor data sets and be similarly protected.
- Consider the full context around sensitive data use, including employee teams and departments.

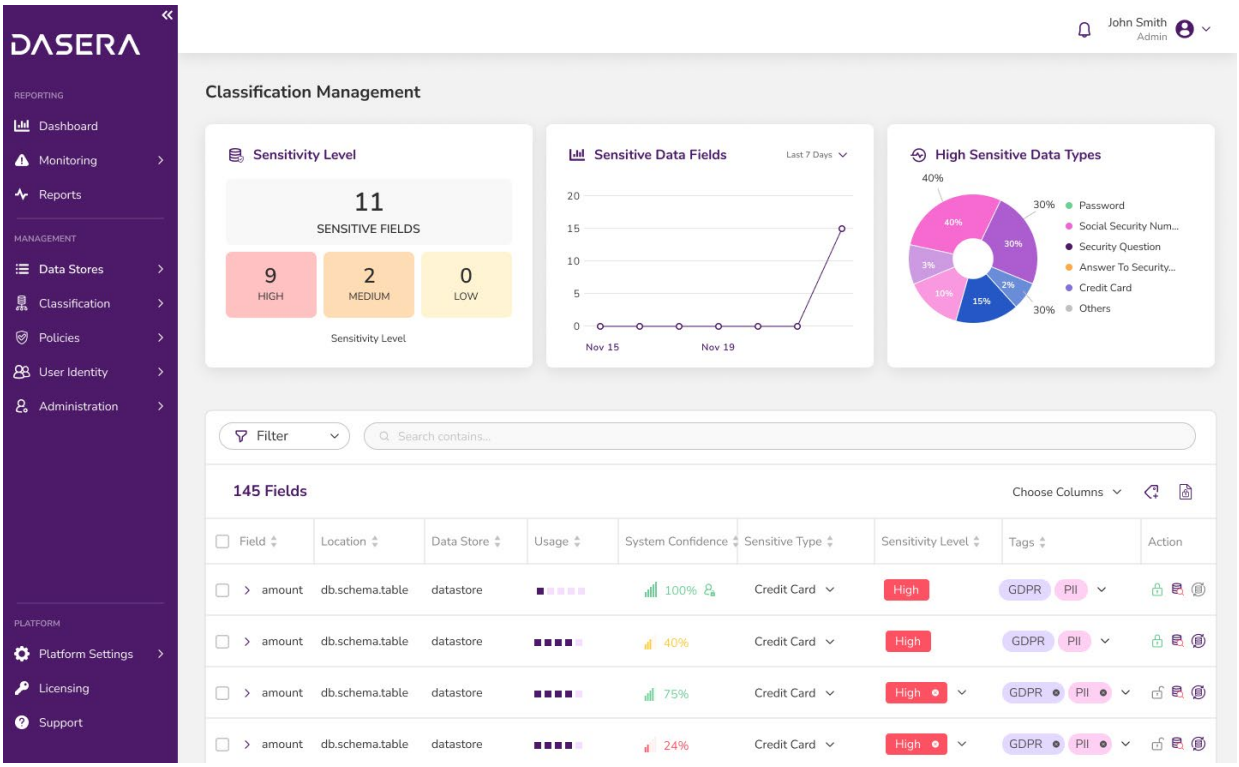
Dasera Radar protects the entire lifecycle of data stored in Snowflake. Dasera does this with a host of capabilities, including:

- Automatic Discovery & Classification
- Privilege Monitoring
- Data Use Monitoring via Query Analysis
- Data Lineage Tracking
- No-Code Policy Editing
- Alerts, Workflows, and Remediations

Automatic Discovery & Classification

Dasera continuously scans your Snowflake data lakehouse to identify your Snowflake structured and semi-structured data stores, and any changes to them. Dasera also samples and classifies all data fields to identify any fields that contain sensitive data.

Dasera includes a number of built-in sensitive data classifiers, like Mailing Address or Social Security Number. In addition, you can define your own regular expressions so Dasera can detect sensitive data that is specific to your environment. Data fields are also tagged by sensitivity level and regulations that apply to that sensitive data type, like GDPR and CCPA. Data owners can then be notified to review classifications.



Privilege Monitoring

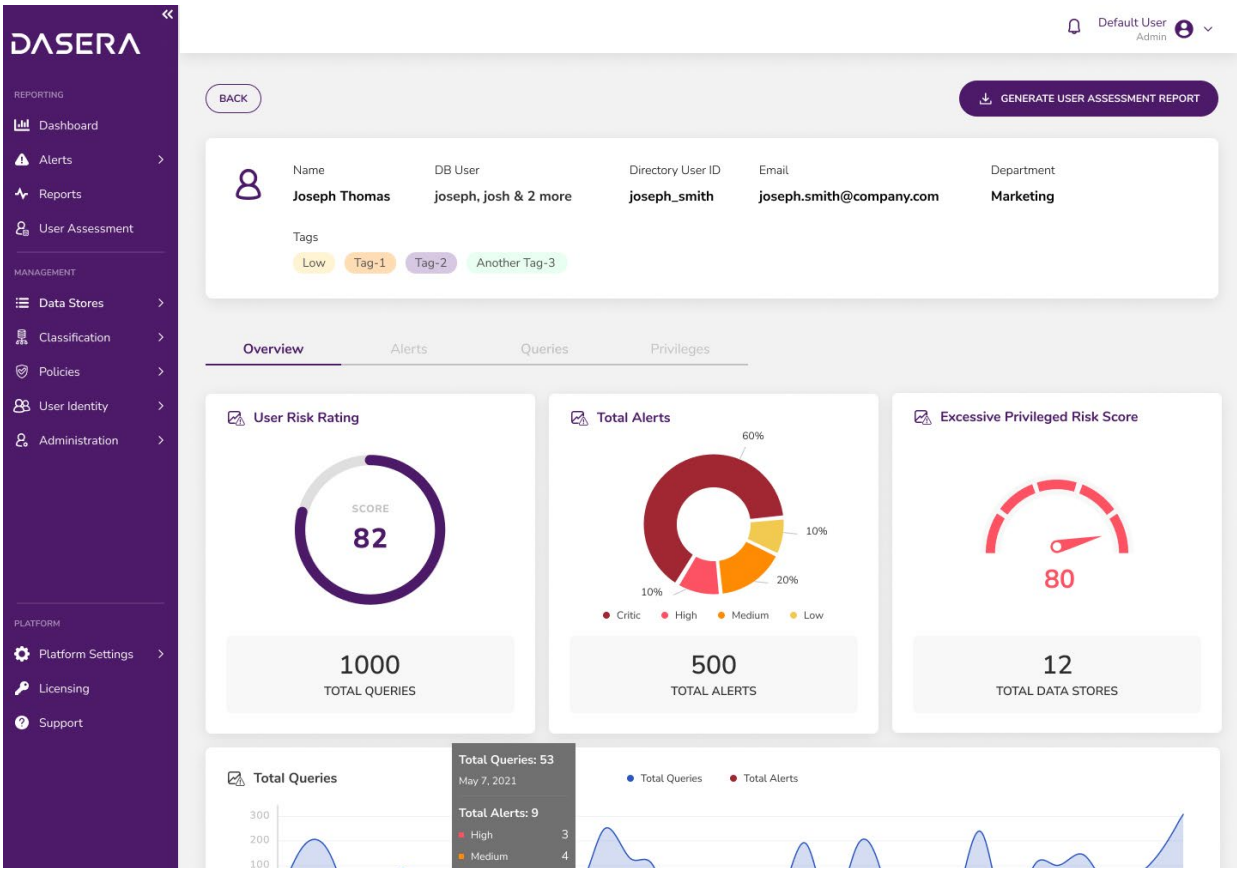
Dasera crawls privileges within Snowflake, keeping track of who has what kind of privileged access to sensitive data throughout in your Snowflake data lakehouse.

Combined with data discovery & classification, Dasera can be set up to automatically find mistakes in granting privilege, e.g., data lake buckets that are open to the public (but shouldn't be), or one customer inadvertently having access to another customer's bucket. Dasera can also be used to monitor privilege sprawl across Snowflake.

Data Use Monitoring via Query Analysis

Every data interaction with Snowflake ultimately occurs through an SQL query. Dasera capitalizes on this fact by passively analyzing every query to understand how data is actually used, with extremely high fidelity and precision without slowing down or disrupting the work of data users. Dasera's query analysis reveals meaningful insights into which data interactions were risky, which employee was responsible and how many rows of data were affected.

Data use monitoring, when combined with privilege monitoring, can also be used to automatically detect over-privilege — i.e., when someone has privilege to access sets of sensitive data, but doesn't use that privilege for extended periods of time. Dasera can then be set to automatically reduce privilege.



Data Lineage Tracking

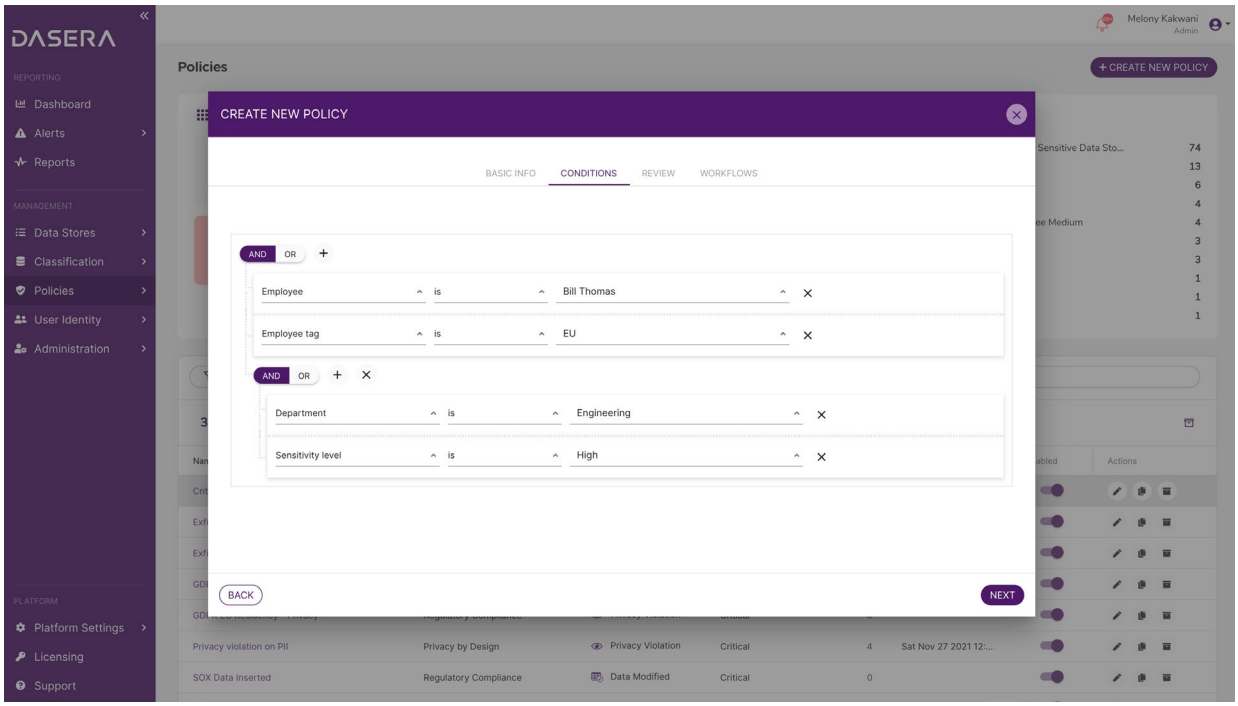
In any cloud data store — including Snowflake — data is constantly in motion, being copied and transformed. Taking a periodic snapshot of data inventory in order to decide which policies should be applied to how that data is used is therefore ineffective.

That's why Dasera also tracks data lineage. Any field that's copied from a field previously known to Dasera will inherit the same sensitive data classification and tags and be protected by the same data use policies that protect its parent(s). Additionally, you'll be able to track all the ancestors and descendants of a given field, find permission mismatches across different generations of fields, and write lineage-centric policies.

No-Code Policy Editor

Sensitive data has many stakeholders, and many of those stakeholders don't know how to code. That's why Dasera comes with a no-code policy editor. Members of your legal and compliance team can write/edit sensitive data use policies themselves using our no-code policy editor, saving the SecOps team valuable time and effort.

The no-code policy editor takes into consideration the full context of the data use — e.g., is sensitive data incorporated into the query itself? Does the query produce sensitive data? — as well as the context of the data user. Dasera integrates with your employee directory so policies can target specific employees, their teams, and/or their departments.



Alerts, Workflows, and Remediations

Your security and compliance teams want to know when bad things happen, not just on a quarterly basis, but ideally in real time or near-real time. That's why every Dasera policy violation results in an alert with a configurable criticality. But it doesn't end with alerts. With Dasera, you can define workflows and target exactly who in your organization should be notified — Data Owners, SecOps, Legal, and/or Compliance — and exactly how they should be notified — by email, Slack, PagerDuty, SNS, etc.

With some violations, you may want some immediate remediations to happen as a precaution. With Dasera, you'll be able to specify automatic remediations within the workflows, like suspending a user's database privileges and quarantining a highly sensitive field.

DASERA

Learn More About Dasera

Dasera's data security platform combines collaboration and automation to ensure data is protected from creation to deletion. To learn how you can secure your Snowflake data with Dasera, please contact sales@dasera.com.

snowflake

Learn More About Snowflake

Snowflake enables every organization to mobilize their data with Snowflake's Data Cloud. Customers use the Data Cloud to unite siloed data, discover and securely share data, and execute diverse analytic workloads. For more information, please call **1-844-766-9355**.