Snowflake at the Data Core: PAYSTRAX Journey from day 1

Raimonda Bružienė | BI Team Lead

Mikas Šimoliūnas | BI Developer

PAYSTRAX
A EUROPEAN PAYMENTS COMPANY

About Raimonda Bružienė

BI Team lead at PAYSTRAX

Traveler

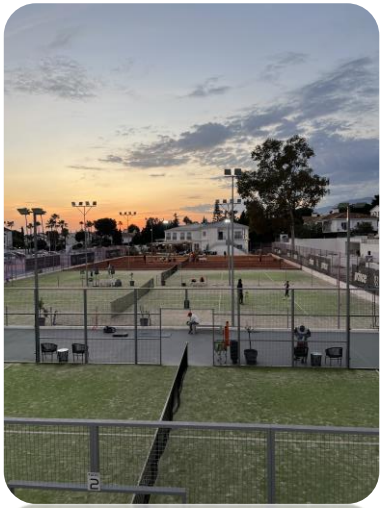
- Always in debt with vacation days.

Padel enthusiast

- Started before it was cool in Lithuania

PAYSTRAX local kombucha producer

- Who doesn’t love a pet scoby and good bubbles!



About Mikas Šimoliūnas

BI Developer at PAYSTRAX

Avid pc gamer

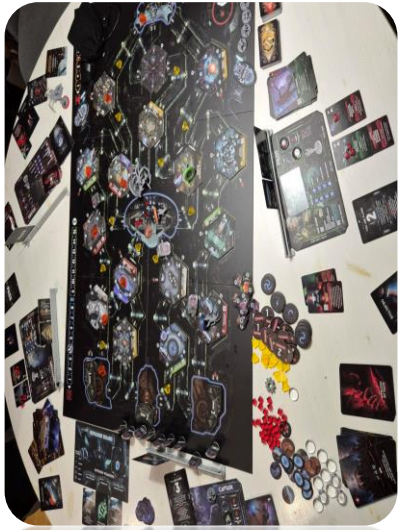
- Path of Exile II
- Stoneshard
- Gunfire reborn
- Escape From Tarkov
- Total War: Warhammer II & III

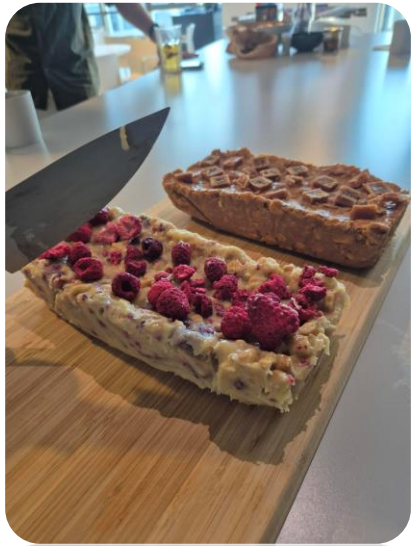
Boardgame enthusiast

- The witcher: Old World (With expansions)
- Dune Imperium: Uprising
- Nemesis (1st release)
- SETI
- And many many more...

Expert lazy cake cook

- Out of this world lazy cake (tinginys). Examples can be seen in the picture. Countless cakes baked for PAYSTRAXERS in Vilnius office.






PAYSTRAX

PAYSTRAX

About PAYSTRAX

PAYSTRAX is a payment services provider with PAYMENT INSTITUTION licences in Lithuania and UK. We help businesses get payments as quickly, securely, and cost-efficiently as possible.

We provide Visa and Mastercard transaction services for POS-terminals, online, mobile or any other electronic payment systems.

6

International offices around Europe and the UK

130+

Employees (over 58% growth since the beginning of 2023)

800+

Merchants served by PAYSTRAX in EEA & UK

~40%

Growth of merchants* since 2023

490 mln.

Acquiring volume in 2024

50+

PAYSTRAX partners in EEA & UK

*Merchants – Businesses or individuals that sell goods or services, typically in exchange for payment.

Our values

Ambition

We reach further by:

- Doing the impossible
- Advancing courageously
- Striving for excellence
- Learning continuously

Professionalism

We create trust by:

- Appearing and communicating professionally
- Committing fully
- Fostering mutual respect

Fun

We make it worthwhile by:

- Getting a laughter going
- Embracing a stress-free environment
- Building teams
- Celebrating achievements and making memories

Choosing the tooling

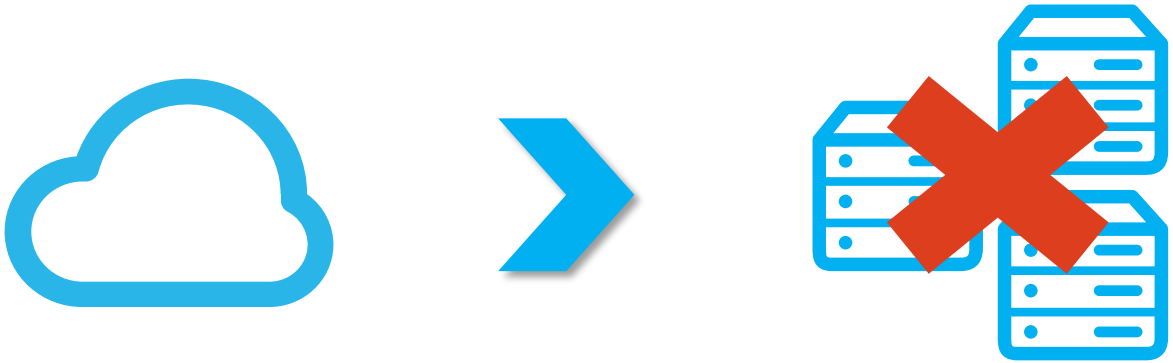
We picked…

What we were looking for

- Started from scratch - no predefined infrastructure or complex migration.
- Our goal was to build a dedicated, centralized data warehouse that could seamlessly scale and support our future analytical and reporting needs.

What we were looking for

- Started from scratch - no predefined infrastructure or complex migration.
- Our goal was to build a dedicated, centralized data warehouse that could seamlessly scale and support our future analytical and reporting needs.
- Avoiding on-premise servers with high up-front costs and maintenance, we turned to cloud-based technology and SaaS providers.



CLOUD

ON-PREMISE

What we were looking for

- Started from scratch - no predefined infrastructure or complex migration.
- Our goal was to build a dedicated, centralized data warehouse that could seamlessly scale and support our future analytical and reporting needs.
- Avoiding on-premise servers with high up-front costs and maintenance, we turned to cloud-based technology and SaaS providers.
- Flexible scalability and cost optimized approach to usage. We pay only for what we use.

FLEXIBLE SCALABILITY

COST OPTIMIZATION

What we were looking for

- Started from scratch - no predefined infrastructure or complex migration.
- Our goal was to build a dedicated, centralized data warehouse that could seamlessly scale and support our future analytical and reporting needs.
- Avoiding on-premise servers with high up-front costs and maintenance, we turned to cloud-based technology and SaaS providers.
- Flexible scalability and cost optimized approach to usage. We pay only for what we use.
- Avoiding dedicated DBA role.

Database Admin

SO WHY SNOWFLAKE ?

What Snowflake delivered

What we needed

- Clear documentation and excellent support, so that setting up the systems would be quick and easy.

Community

What Snowflake delivered

What we needed

- Cloud-based storage and processing to eliminate the need for hardware maintenance and dedicated IT personnel. True **„all-in-one“** package brought in by Snowflake.

Storage

Processing

Security

Administration

What Snowflake delivered

What we needed

- Dynamic resource allocation combined with a transparent, usage-based cost model that ensures precise cost control.

Flexibility

Transparent cost

What Snowflake delivered

What we needed

- Effortlessly scale computing power - up or out - with no upfront investment and no hardware to maintain.*

Scaling up
Scaling out

Pay for what you use

*Keep in mind that compute cost rises!

What Snowflake delivered

More than you expect

- Automatic encryption of data at rest and in transit.

Encryption

What Snowflake delivered

More than you expect

- Automatic partitioning via Snowflakes micro-partitioning architecture. You can forget about partitioning and order keys.

Micro-partitioning

What Snowflake delivered

More than you expect

- Support for structured and unstructured data formats: csv, json, xml, avro, parquet

Structured data

Data format support

Unstructured data

What Snowflake delivered

More than you expect

- SQL with added flexibility of Snowpark framework that allows you to write data transformations in Python, Java, Scala.

Snowpark

Python

Java

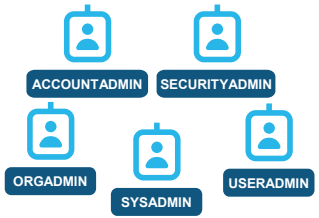
Scala

What Snowflake delivered

More than you expect

- Granular access control implemented through Snowflake's Role-Based Access Control (RBAC) model - managing permissions across users, roles, and objects with precision and scalability.

Users



Roles

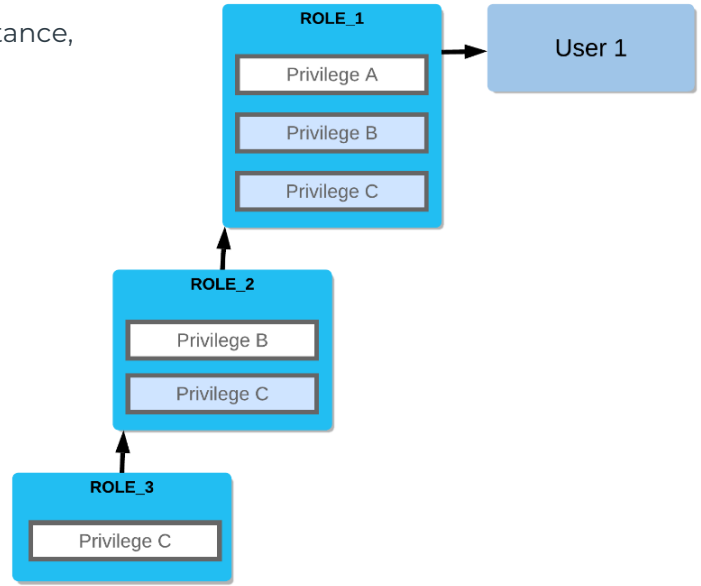
What Snowflake delivered

More than you expect

- Snowflake RBAC* example.

For a more specific example of role hierarchy and privilege inheritance, consider the following scenario:
Role 3 has been granted to Role 2.
Role 2 has been granted to Role 1.
Role 1 has been granted to User 1.

In this scenario:
Role 2 inherits Privilege C.
Role 1 inherits Privileges B and C.
User 1 has all three privileges.



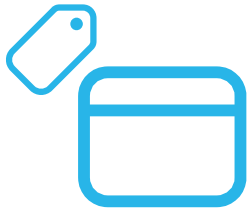
*Role-Based Access Control

What Snowflake delivered

More than you expect

- Object tags and columnar/row level masking policies. Data governance made easy.

Columnar/row masking

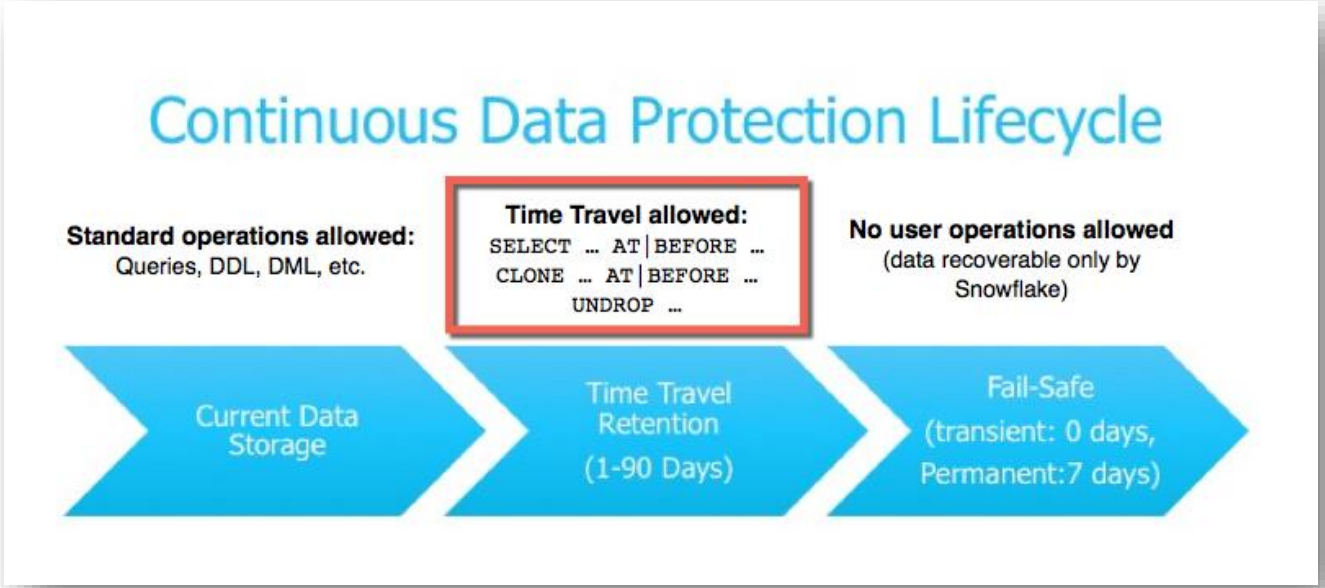


Tags

What Snowflake delivered

More than you expect

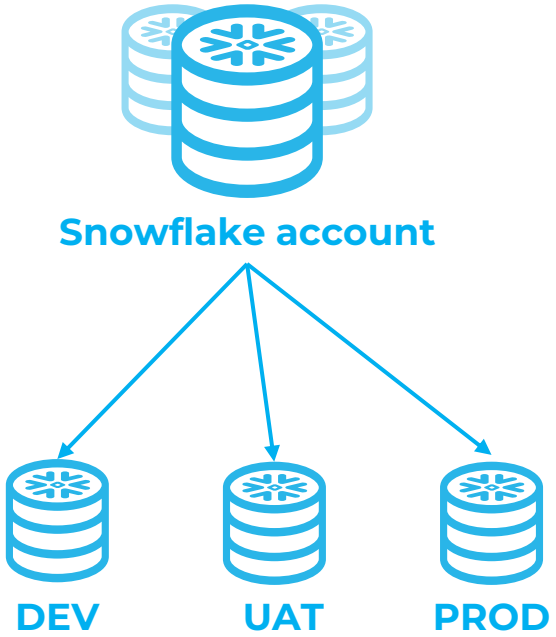
- Disaster recovery of data with time travel and fail-safe.



Places where Snowflake shined for us

Environments

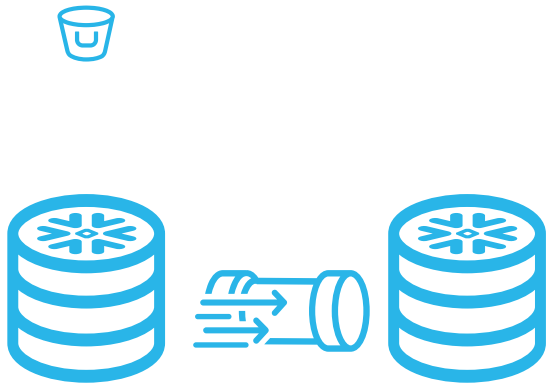
- Easy to setup multiple Snowflake accounts to distribute workloads/business functions.
- Accounts can be set up on:
  GCP, AWS and Azure.
  We use Azure and AWS.
- Segment development environments using separate accounts and turbocharge them via Snowflake private sharing.
  (ex. dev > uat > prod)



Places where Snowflake shined for us

Replication

- Straightforward and easy to use for databases and shares.
- Schedule or run manually on demand. The replication is asynchronous. Scheduled executions do not clash, and a queue mechanism manages every execution.
- Snowflake supports replication across regions (AWS → Azure → GCP) and between cloud providers, making it one of the few platforms that allows true multi-cloud data redundancy without complex setup.

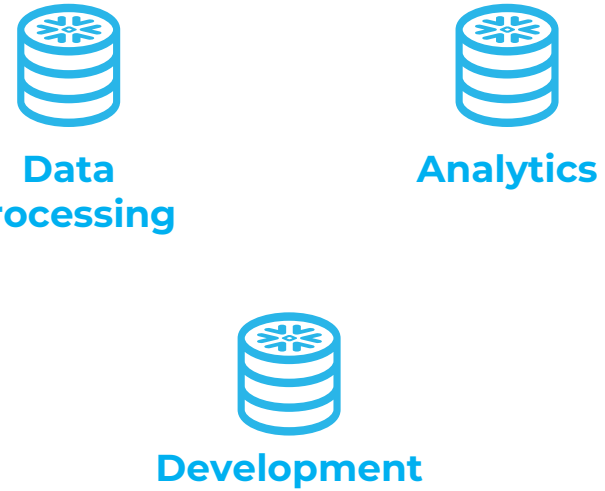


Replication

Places where Snowflake shined for us

Flexibility

- Distribute workloads over multiple accounts and/or warehouse units.
- Possibility to automatically allocate more resources for demanding tasks (multi-cluster warehouses, QAS*)
- Mix and match Snowflake account types with specific workloads (storing and processing in standard, masking in Enterprise etc.)



*Query acceleration service

Places where Snowflake shined for us

Data sharing

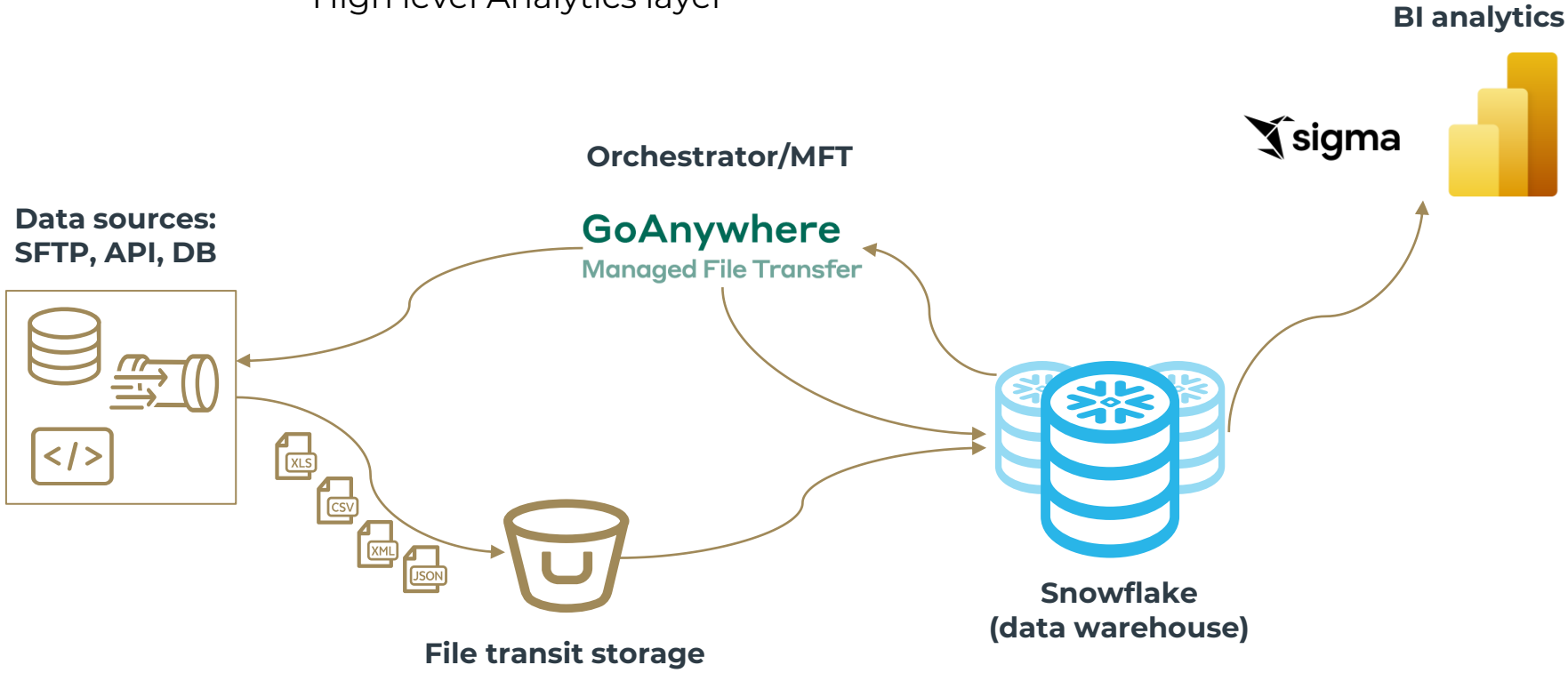
- Supports csv, json, xml, avro, parquet and other formats.
- Smooth integration via data shares with any service provider who also uses Snowflake for seamless integration and sharing.
- Excellent functionality for mirroring prod data in dev.
- Supports data ingestion and distribution through cloud storage platforms such as AWS S3 and Azure Blob Storage.

Plethora of formats

Data sharing

Architecture

High level Analytics layer



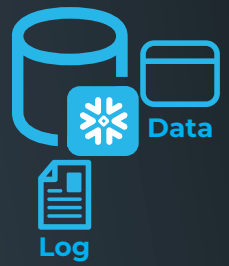
What we achieved using Snowflake

Centralised data storage and avoided data silos.

Prod environment contains the golden copy of data and using Snowflake private sharing we can mirror the data and “test in production” through our development environment.

What we achieved using Snowflake

Logging framework integrated with our orchestration tool.
Monitoring service & data health and availability.



What we achieved using Snowflake

Thousands of data objects and transformation processes for self-service analytics made possible via Power BI & Sigma (& Excel..).

Actively supplying hundreds of analytic solutions.

What we achieved using Snowflake

Data quality framework.

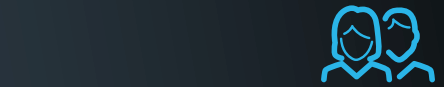
Defined business quality rules for validation, system comparison and pruning.

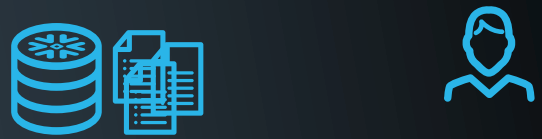
Added layer of data integrity by involving our end users.

What we achieved using Snowflake

Client statement data delivery service.

Clients receive detailed information via email about their business performance.





Snowflake cortex forecasting

Snowflake Cortex Forecasting is an AI-powered feature within Snowflake's Data Cloud that enables organizations to generate forecasts directly from their Snowflake environment using simple SQL without requiring extensive data science expertise.

This is especially beneficial because it allows organisations to predict key metrics such as:

- transaction volumes
- fraud risk
- Customer churn
- ARPC*

*Average revenue per customer

Snowflake cortex forecasting

CREATE SNOWFLAKE.ML.FORECAST

```
CREATE [ OR REPLACE ] SNOWFLAKE.ML.FORECAST [ IF NOT EXISTS ] <model_name>(
  INPUT_DATA => <input_data>,
  [ SERIES_COLNAME => '<series_colname>', ]
  TIMESTAMP_COLNAME => '<timestamp_colname>',
  TARGET_COLNAME => '<target_colname>',
  [ CONFIG_OBJECT => <config_object> ]
)
[ [ WITH ] TAG ( <tag_name> = '<tag_value>' [ , <tag_name> = '<tag_value>' , ... ] ) ]
[ COMMENT = '<string_literal>' ]
```

Snowflake cortex forecasting

<model_name>!FORECAST

We build a model with a simple SQL DDL statement.
The data in the **forecast_training_dataset** contains records up until year 2025.

```
create or replace snowflake.ml.forecast ml_volume_forecast (
    input_data => TABLE(select ts,volume from test_db.ML.forecast_training_dataset),
    timestamp_colname => 'ts', // timestamp
    target_colname => 'volume' // numeric
);
```

We then forecast the year 2025 by calling the **!forecast** module.
By supplying the parameter **forecasting_periods**, we tell our model how far to forecast.

```
call ml_volume_forecast!forecast (forecasting_periods => 365);
```


```
call ml_volume_forecast!forecast (forecasting_periods => 3
```

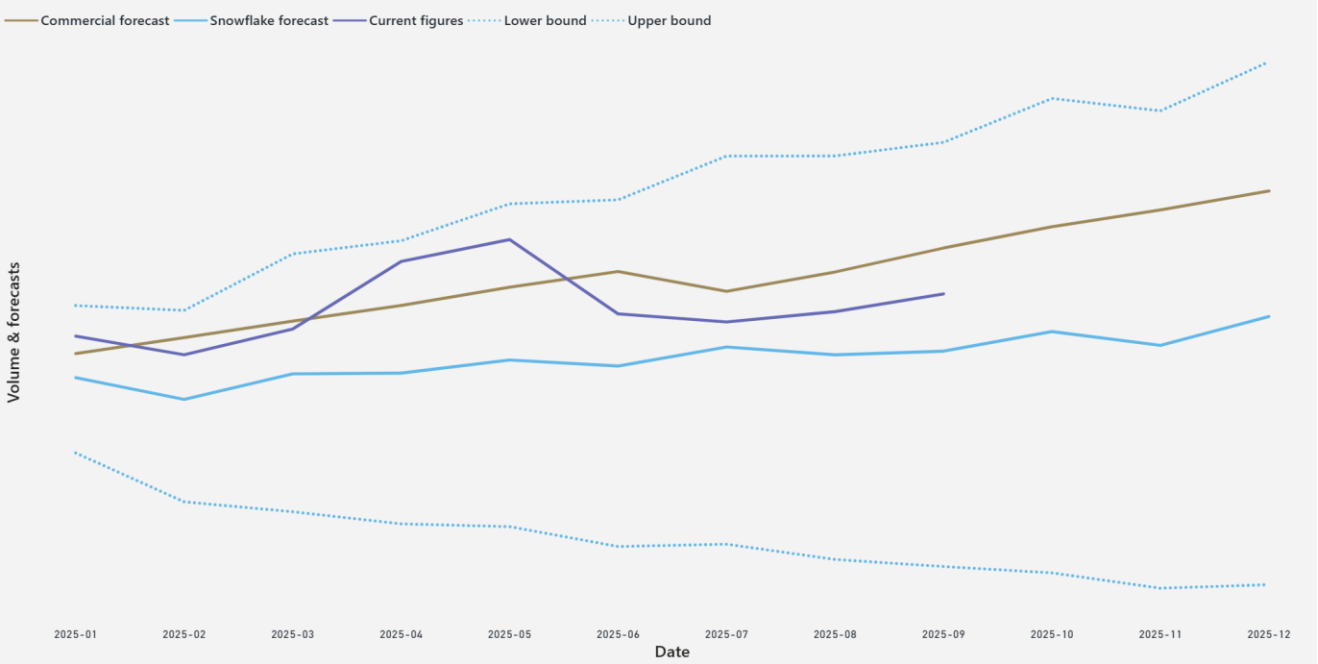
Snowflake cortex forecasting

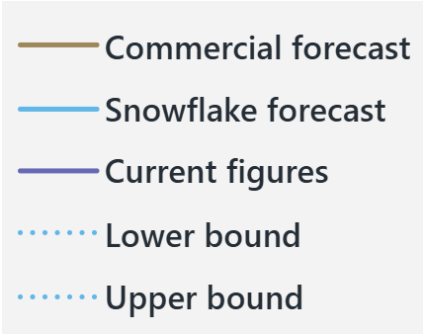
We then reference the call to the forecasting model and store the result in a table.

```
create or replace table test_db.ml.forecasted_volume as
select
    series,
    ts,
    forecast,
    lower_bound,
    upper_bound
from table(result_scan('01bfeb58-0207-6bd7-[redacted]'));
```

Snowflake cortex forecasting

A few magic tricks later we get here.





Data in the visual is partly synthesized!

Snowflake cortex forecasting

Something to keep in mind:

- Native to Snowflake, easy to use and no complex setup.
- Getting a forecast like this up and running took <1 hour.
- General understanding of time series forecasting.
- Model finetuning and performance benchmarking.
- **Data quality matters** & understanding feature engineering.
- Model degradation, drifting and maintenance overhead.

Lessons learned

Logging

Logging framework or tool is essential to maintain smooth and reliable data operations.

Having eyes on your data pipelines and jobs will reduce maintenance overhead if something goes unplanned (and it will eventually happen).

Without proper logging, diagnosing data inconsistencies, job failures, or latency problems becomes time-consuming and error-prone.

Comprehensive logs support compliance and audit requirements which are critical in the fintech sector.

Verifiable record of data processing activities and systemic events.

Logging

Lessons learned

Store historical data or capture data changes

Time-series data without historical snapshots is a recipe for disaster.

Historical records enable organizations to track metric evolution, identify patterns, and forecast future outcomes based on real-world trends.

Maintaining history also supports data recovery, reliability, and continuity for analysts, engineers, and developers alike.

Historical data makes audit trailing and point-in-time analysis stress free.

If you're not capturing historical data yet, start today - your stakeholders and analysts will thank you when historical insights are just seconds away.

History tracking

Lessons learned

User and role management

Role-based access control (RBAC) is a foundational framework that enables organizations to manage user permissions effectively, restricting access based on specific roles within the organization.

Implementing RBAC offers numerous benefits that can enhance security, privacy, and operational efficiency within an organization. By assigning permissions based on user roles rather than individual identities, organizations can reduce the risk of unauthorized access to sensitive data.

This structure not only helps protect critical information but also simplifies compliance with regulations like HIPAA, GDPR and SOX by providing a clear and auditable framework for managing access to sensitive data.

Lessons learned

Read the documentation

Read the documentation. Read the documentation. Read the documentation. Snowflake has one of the best written documentations about their platform and processes.

From simple SQL syntax to Machine learning functions that come out of the box.

Documentation

Lessons learned

Fail fast. Reiterate. Upgrade

There is no benefit for an organisation or a contributor to spend time designing and perfecting the solution when requirements can change like the temperature.

My recommendation to grow individually and deliver at the same time:

1. Start with a POC
2. If the POC is a success, move on to an MVP
3. If the stakeholders approve of the MVP, you can move onto designing the final product.

Please keep in mind that project scope varies and this might not be the best fit for you, it is just something that worked for us quite well.

Reiterate

Lessons learned

Time travel and fail-safe are lifesavers

You can never be too prepared for disasters and there is no better feeling than knowing you are protected and taken care of.

Time Travel allows teams to instantly access historical data versions - enabling quick recovery from errors. Data retention period of up to 90 days.

Fail-Safe - provides an additional 7 day period after Time Travel expires, during which Snowflake can recover data that has been lost due to accidental or malicious actions.
Fail-Safe recovery can only be initiated by Snowflake Support.

Time travel

PAYSTRAX
A EUROPEAN PAYMENTS COMPANY

Thank you!

www.paystrax.com