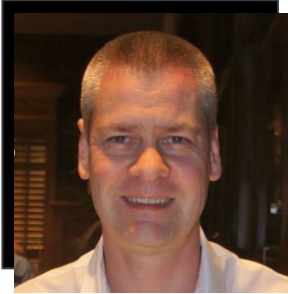# ThoughtSpot Insight Hours

August, 2025

Speakers

Paul Froggatt
Director, Technical
Architects
ThoughtSpot

Eliza Dee
Customer Success Manager
ThoughtSpot
Host

Cam Nicholls
Solutions Engineer
ThoughtSpot

Three pillars of **Insights Hours**

Product spotlight

How-to ThoughtSpot

Community thread

- Hosted **last Thursday of every month** by ThoughtSpot's GTM team

- Register for **one or all sessions**

- Ask your questions using Q&A for the **Community thread** discussions

Agenda

01

Introductions

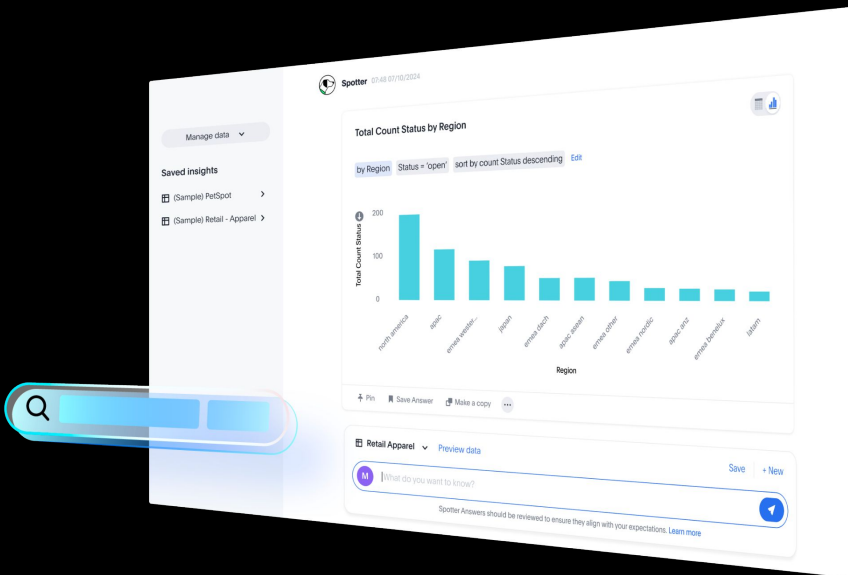Hello from speakers and what we'll cover today

02

10.11 cl release updates

Cam Nicholls

03

Optimisations and best practices

Paul Froggatt

04

Wrap up & next steps

Eliza Dee

10.11 cl release highlights

10.11 cl Agenda

01

Asking 'Why' questions in Spotter

02

Liveboard Styling and Grouping

03

Enhanced Dynamic Parameters

04

Filtering Liveboards for users with limited access to its Worksheets or Models

05

KPI Monitor Alerts in Slack

06

AI and BI System Liveboard

Asking 'Why' Questions in Spotter

- **What is it:** A new diagnostic mini-agent built into Spotter. It's designed to explain the reasons behind changes in your key business metrics.

- **Value:** Accelerated time to uncovering more insights

Note:
1. **Needs chart data to be shared with LLM.**
2. **EA: 10.11; with Spotter Agent**

https://docs.thoughtspot.com/cloud/10.11.0.cl/spotter-why
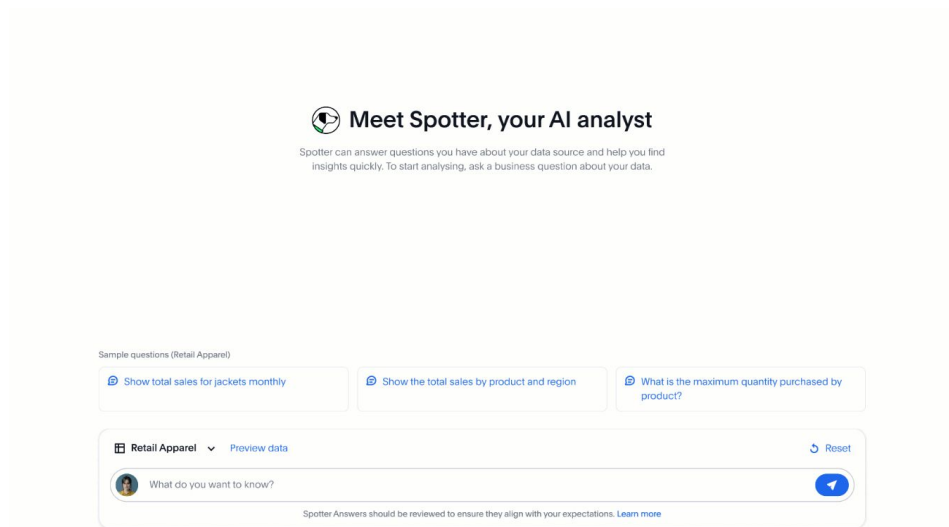
Liveboard Styling & Grouping

- **What is it:** The ability to group tiles together on a Liveboard and choose from a wider set styling options

    - Grouping: You can group Answers and Note tiles together in a Liveboard.

    - Styling: With a new styling panel in the Liveboard, you can do the following: Style parts of the Liveboard. Style parts of a group. Style parts of an Answer or note

- **Value:** Broader styling options to add to your Liveboards!

https://docs.thoughtspot.com/cloud/10.11.0.cl/liveboard-grouping-styling
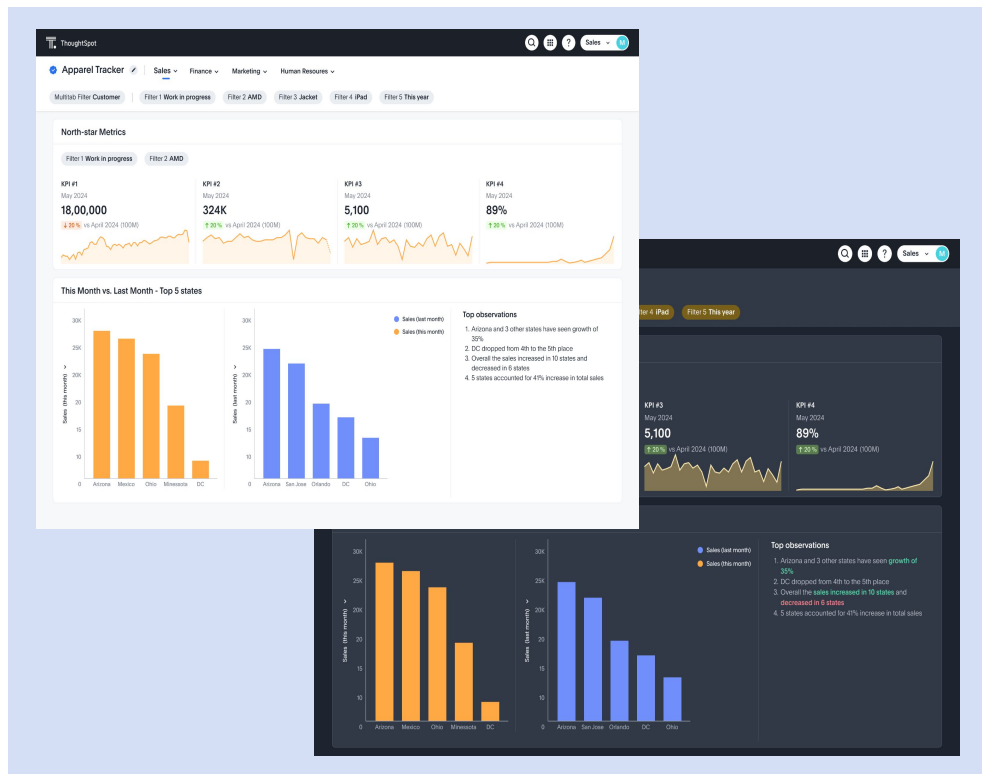
Enhanced Dynamic Parameters

- **What is it:** Dynamically name axis and chart names and descriptions based on the selected parameter value

- **Value:** Any place a parameter is used in Liveboards or Answers automatically displays the current information associated with the selected parameter value for even more scalability and customization.
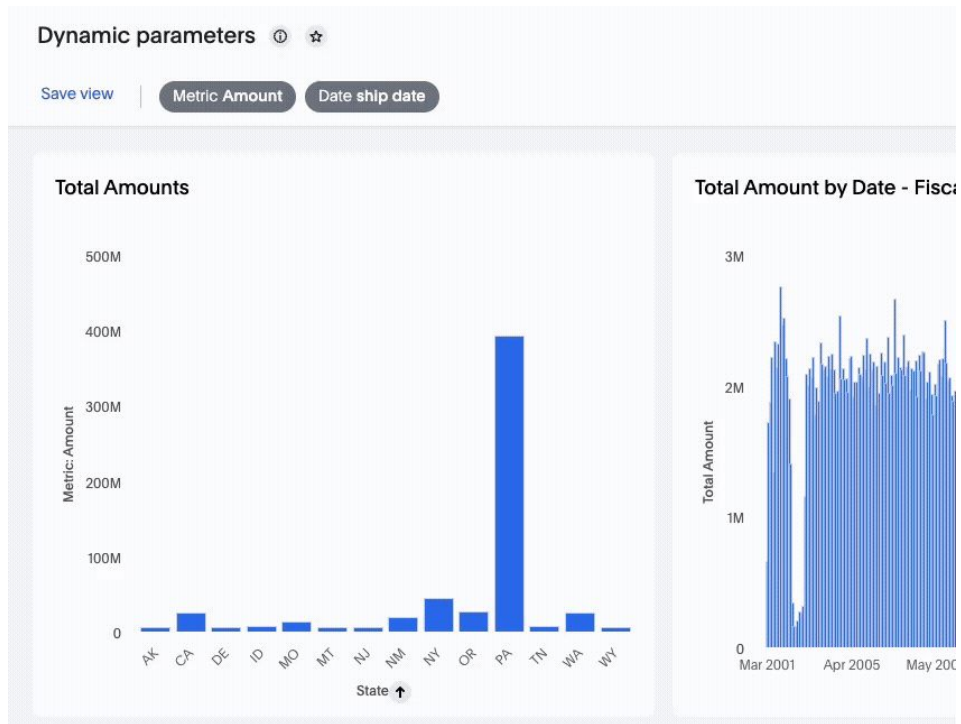
Filtering a Liveboard for users with limited access to its Worksheets or Models

- **What is it:** Greater interactivity in multi-Modal Liveboard scenarios where users don't have access to all Models in the Liveboard.

- **Value:** Greater accessibility for users without the need for data permission trade offs

https://docs.thoughtspot.com/cloud/10.11.0.cl/liveboard-filters
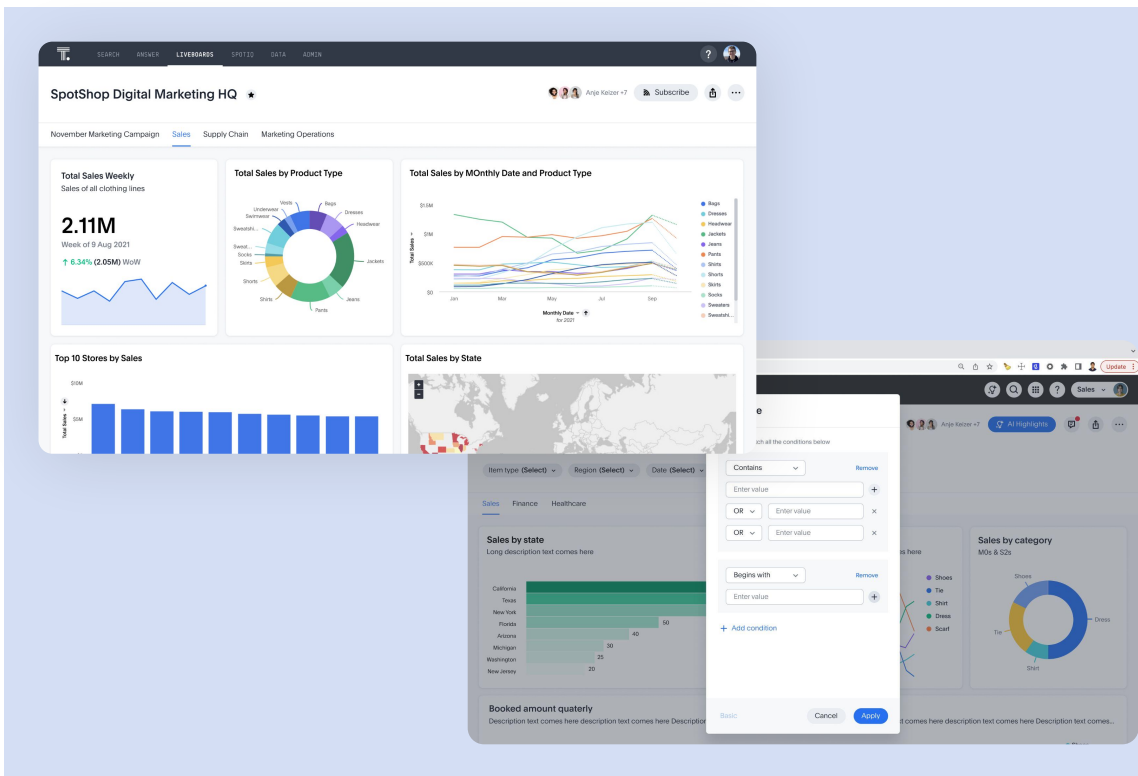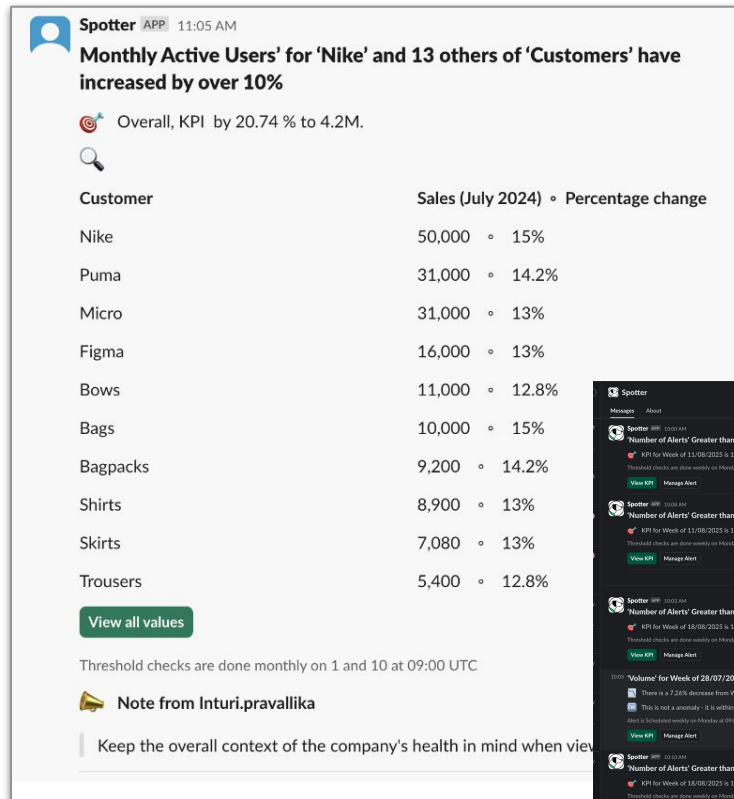
KPI Monitor Alerts in Slack

- **What is it:** Monitor alerts can send alerts directly into Slack

- **Value:** Meet your users where they are and get notified in Slack for the KPIs you care about

Note:

1. **This feature is available in clusters that have the Spotter App on Slack**

AI and BI System Liveboard

- **What is it:** A new system Liveboard to monitor key metrics of adoption across ThoughtSpot

- **Value:** Metrics are directly tied to query logs, reducing ambiguity and ensuring precise reporting.

Note:

1. **Users need to be Admin to see**
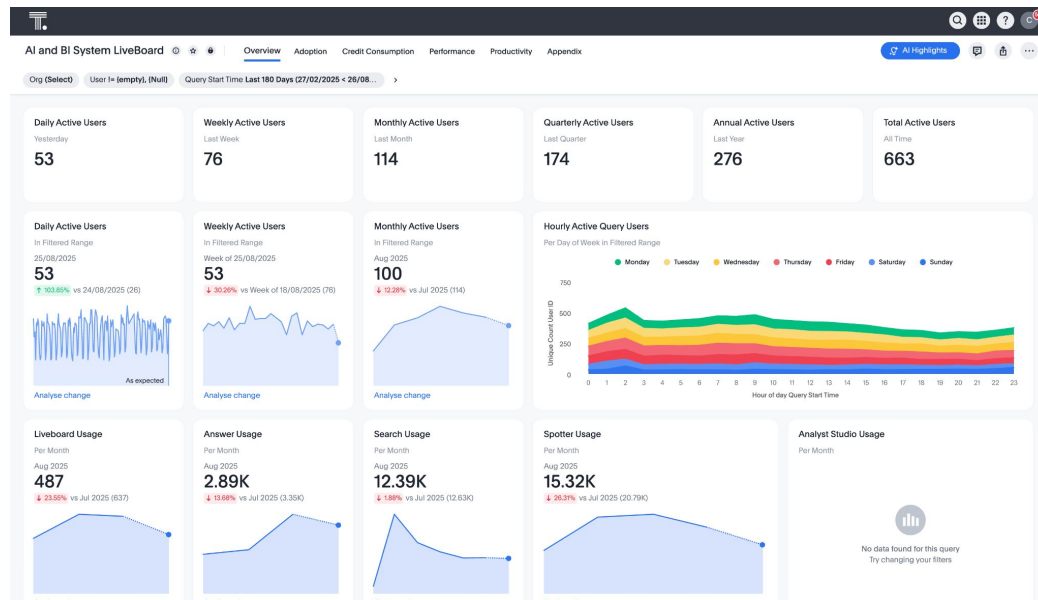
How to

Optimisations and best practices

Optimisations & Best Practices - Topics

Liveboards

Formulas

Data Model Design

Frequency & Concurrency
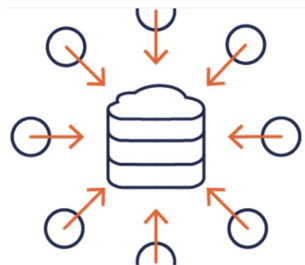
Query Performance

Liveboards

ThoughtSpot's Liveboards use a "lazy loading" technique:
- Executes queries (in parallel) for all the immediately visible visualisations when a Liveboard is opened
- Defers executing other queries, e.g.
  - For additional visualisations, until you scroll down to display them
  - For visualisations on another tab, until you open the tab
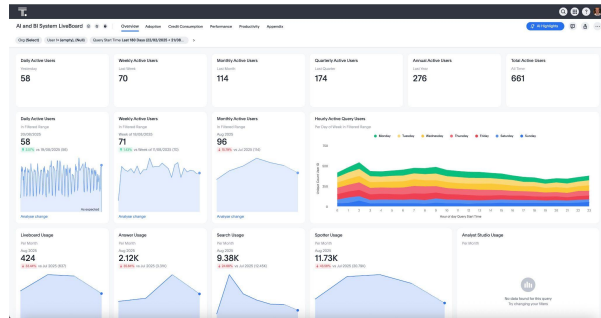
Use Multiple Liveboard Tabs
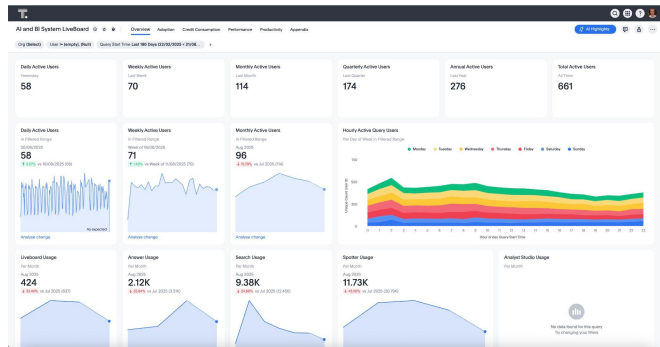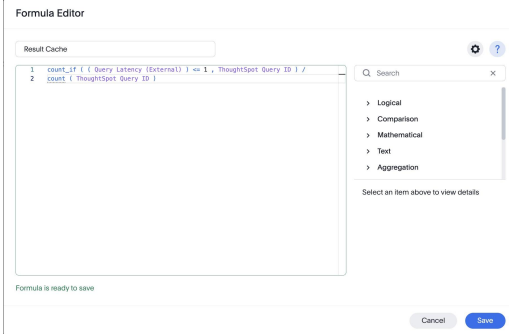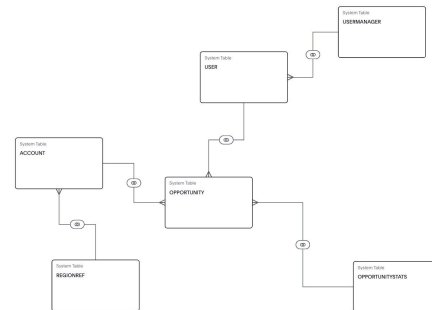- Split visualisations into logical distinct topics/subjects
- Split high level KPIs from more granular visualisations or tables
  - Or put high level KPIs at the top, and more granular visualisations further down

Limit number of visualisations per tab
- Rule of thumb - no more than 20
  - Tabs with larger number of visualisations carry some performance overheads

Leverage Liveboard filters
- Choose sensible default values, e.g. filter a date to show recent data
- Use mandatory filters, to avoid executing queries until filter values have been chosen
- Encourage use of Personalised Views, to automatically apply user preferred personal filters

Formulas

Avoid row-based calculations
- Calculate these in the data pipeline and store in a physical column, e.g.
  - You have an orders table with order date and ship date columns
    - You often want to calculate the days between those dates
   - Calculate this in the data pipeline and store this in a column in the orders table
  - You often need to analyse transactions compared to a customers first ever transaction
    - Once a customer transacts, that date will never change
    - Detect this in the data pipeline and store this as an attribute in the customer table
    - Rather than calculating the minimum transaction date every time you query the data

Avoid use of now()
- If you just need today's date, use today() instead
- Queries using now() will never be cached by a database
  - Every time you execute it the result will be different

Parameters - when used in conditional (if/then/else) formulas
- Ensure this flag is enabled on your service to optimise & simplify the SQL generated
  - orion.dataGenRules.enableConstantFoldingInQueryGen true
- Generates the SQL for just the logic required, rather than for all conditions
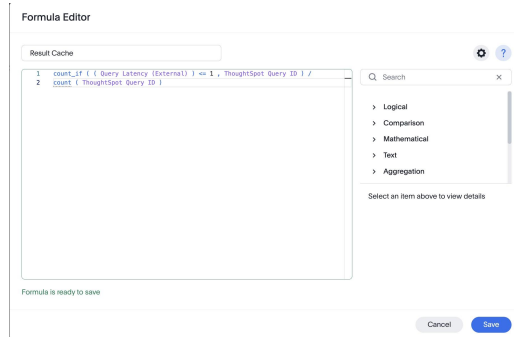
Data Model Design

Avoid use of logical views
- Database views, custom SQL views, answers saved as views
- The entire view logic gets executed at runtime, even if you only need one attribute from the view
    - The more complex the view, the more overhead and performance drag
- Use physically materialised views or tables in the database instead

Avoid using model filters
- Unless using the new progressive model filters feature (Beta in 10.10.0.cl release)
    - Without this feature, the filter is applied to ALL queries
    - With this feature, the filter is only applied when the table with the filtered
    - column is referenced in a query

Use progressive model joins (the default)
- Ensures only tables required are included in the query

Ensure join cardinality is accurate
- Make sure the 1 side of a join is unique
- Get this wrong and you may have incorrect results, or unexpected poor performance
- Break M:M joins into M:1:M joins, by introducing a "bridge" table with unique join keys
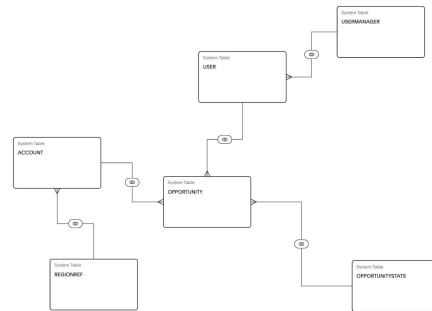
Data Model Design

Choose appropriate join direction for a 1:1 join
- Join direction, and the convergence of joins influences the query generation
- Ideally, ensure joins across a model flow consistently away from facts (many)
- Get this wrong and you may have incorrect results, more complex queries and degraded performance

Convert boolean datatype to a meaningful varchar
- Enables a better search experience
- More user friendly to search for a phrase such as "active customers"
  - Than to search for "true" or "active customers = true"
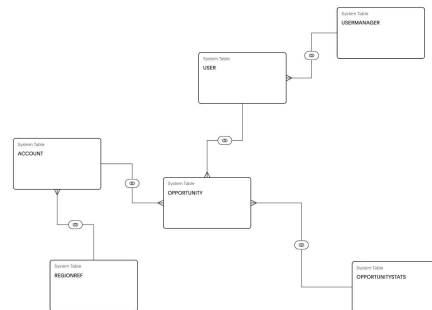
Frequency & Concurrency
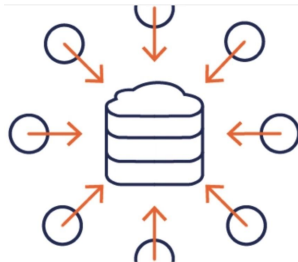
Review frequency and timing of recurring tasks
- Avoid over frequent scheduled recurring tasks
- Concurrent tasks in ThoughtSpot generate concurrent queries in the database
- Concurrent database queries can lead to resource contention and performance degradation
- Example tasks:
  - Liveboard schedules
  - Sync schedules
  - SpotIQ schedules
  - KPI alerts
  - Search Indexing

Leverage connection configurations
- Snowflake only at present, will be available for Databricks and Google BigQuery in 10.12.0.cl
- Enables one connection to be used for different workloads
- Split user workload from background workload (search indexing, row counting)
- Split groups of users into different performance profiles, e.g. using different sized warehouses

Leverage database scale-out options
- Some databases provide dynamic scale-out features
- As concurrency increases, they can automatically scale out to provide more resources to avoid queries queueing

Query Performance

Leverage Analyst Studio to extract and cache data
- Great use case for highly aggregated data
- Query database once, use multiple times

Use the most appropriate granularity of data
- Not efficient to use daily grain data if most users analyze at a monthly grain
- Consider using:
  - Multiple models - granular model, aggregated model
  - Materialized views for aggregated data (if database support query rewrite)

Define primary/foreign keys in the database (if supported)
- Can help the database optimiser build optimal query execution plans

Build regular database statistics (if required)
- Can help the database optimiser build optimal query execution plans

Use appropriate data types
- More performant to join or count INTEGER data types than VARCHAR's

Query Performance

Use appropriate physical data storage options
- Most databases provide features such as:
    - Partitioning, clustering, compression, indexing, sorting etc.
    - Some databases try to automate these based on query access patterns they don't always get it right!

Avoid suspending the database too frequently, due to inactivity
- Causes some or all database cache to be lost (data, query plan, query results)
    - Subsequent queries will be slower
- Clearly, there is a balance between cost and performance

Avoid unnecessary database functions
- If you have columns with data guaranteed to be in lower or upper case
    - Enable ThoughtSpot column casing feature
        - By setting flag orion.embraceConfig.useColumnValueCasing true
        - Update table columns to define them as ALL lowercase or ALL uppercase
    - Reduces query execution overhead
        - By avoiding a LOWER() function to convert data to lowercase
    - Enables better query optimisation
   - Use of a column function can prevent use of partition elimination or use of an index

What's coming & next steps

SpotGuide

Welcome to SpotGuide!

SpotGuide is your go-to hub for best practices, expert insights, and practical recommendations curated by ThoughtSpot's professional services and customer success teams. Here, you'll find everything you need to implement ThoughtSpot effectively, drive adoption, and maximize the value of your investment. Dive into the wealth of resources and strategies to help you confidently achieve your business goals.
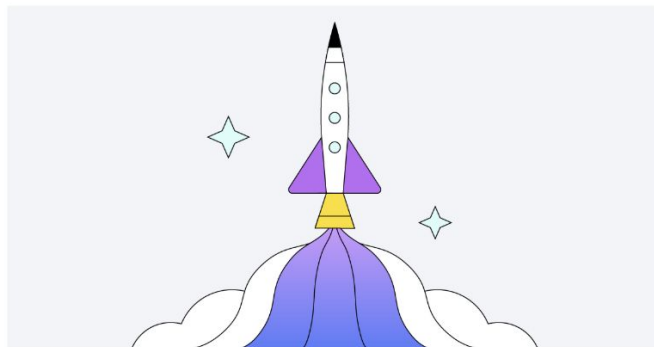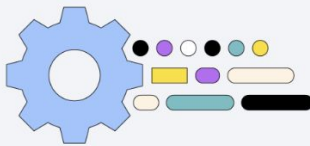
Onboarding & Implementing ThoughtSpot
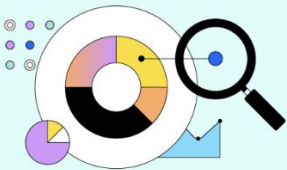
Defining a Use Case

Learn how to document the business context, challenges, success criteria, and the business user questions to be answered.

Access Controls and Security

Establish a strong security model to ensure effortless user access to necessary information, while also preventing unauthorized access and protecting security.
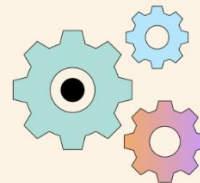
Data Security Field Guide

Learn about security controls, user frameworks, and best practices for data security, including scenarios, use cases, examples, and illustrations of data security governance.

Welcome to ThoughtSpot University

Learn anytime, anywhere. ThoughtSpot University is your central hub for all things training. Take on-demand, self-paced eLearning or register for instructor-led training classes and certifications.

Get started

Self-paced eLearning

Our self-paced learning paths help guide and prioritize your ThoughtSpot learning experience.

Explore learning paths →

Instructor-led Training

Attend virtual and in-person instructor-led classes run by the best instructors in the business.

Find a class →

Certification

Demonstrate your knowledge of ThoughtSpot through our self-serve certification platform.

Get certified →

Events for September

THE FUTURE IS INSIGHT

September 24-25, 2025   Olympia, London

Register Now

Events for September

See you next month!

September 25th
INSIGHTS HOUR 2025