Prophecy

Prophecy 3.4

New features to turbocharge pipeline development

Bob Welshmer
Senior Sales Engineer
Prophecy

Anya Bida
Technical Evangelist
Prophecy

Kuldeep Singh
AI Architect
Prophecy

Productivity wins!
Swag wins!

Welcome Data Engineers, Data Analysts, Data Leaders!

Prophecy

Prophecy

Prophecy 3.4

New features to turbocharge pipeline development

Bob Welshmer
Senior Sales Engineer
Prophecy

Anya Bida
Technical Evangelist
Prophecy

Kuldeep Singh
AI Architect
Prophecy

MORE is needed by enterprises

Enable users

Data engineers
Oversubscribed

Data scientists

Data analysts
Blocked

Process data

Structured

Semi-structured

Unstructured

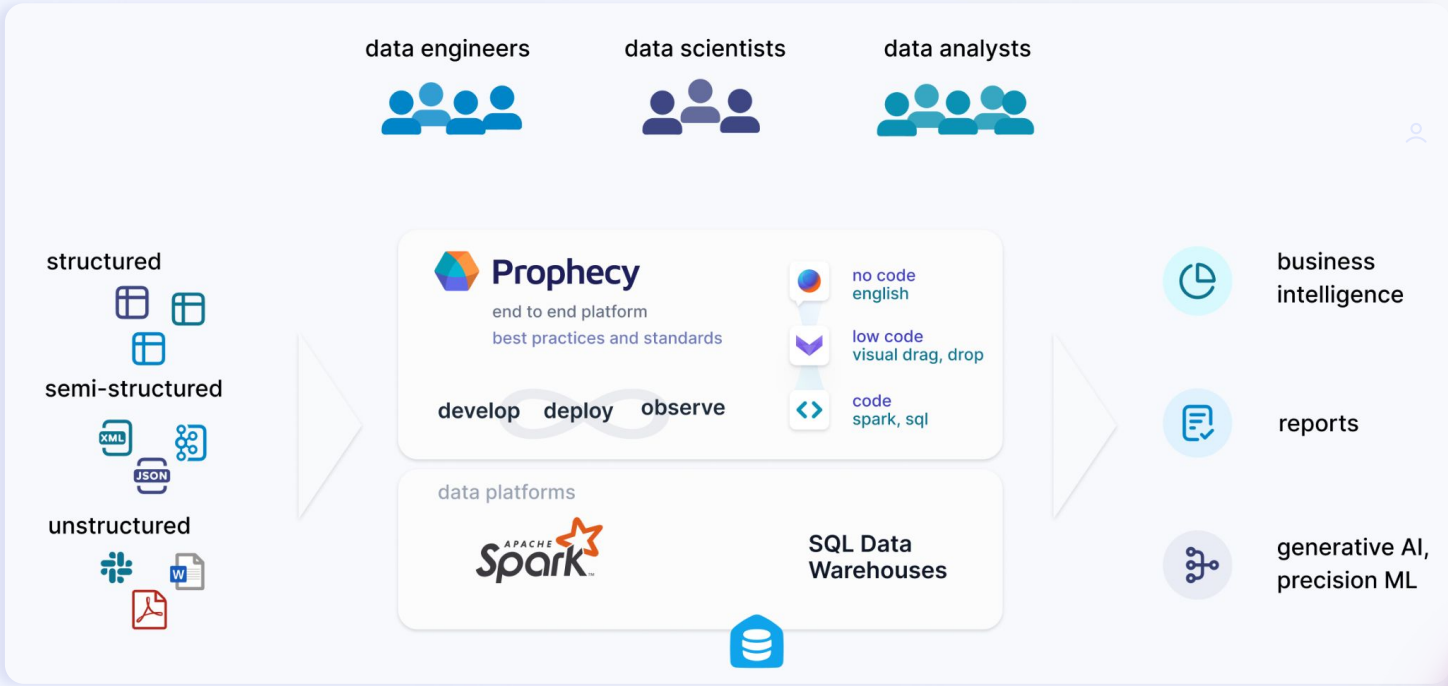
Product analysis

Business intelligence

Generative AI
Precision ML

Reports

Prophecy
Data Transformation Copilot

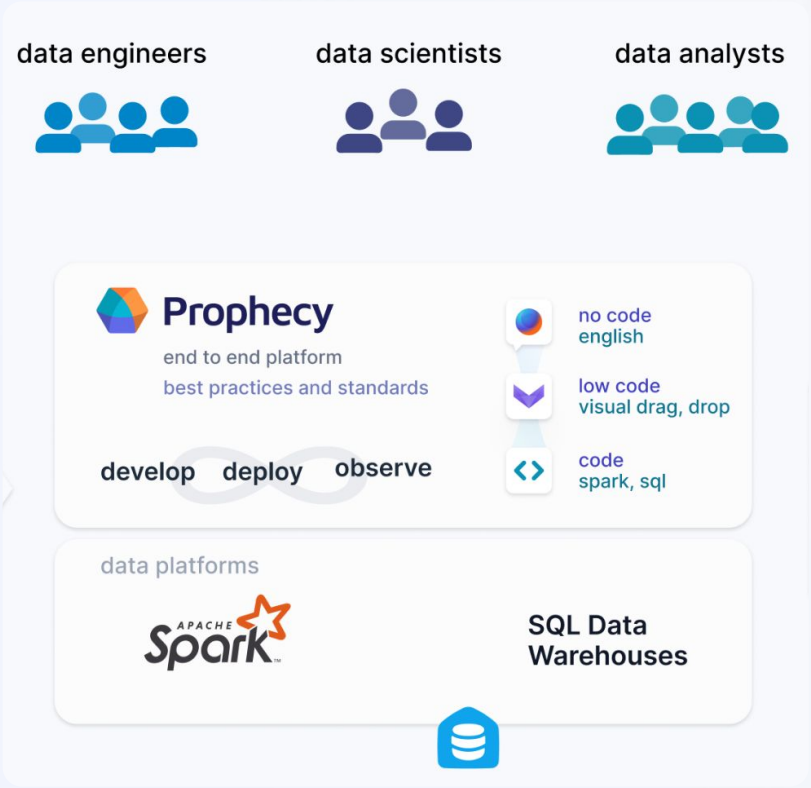


Prophecy

Data Transformation Copilot v3.4

Highlighted innovations

1. Faster development
2. Easy observability
3. AI Capabilities



Data pipelines take too long to build.

Development made better

Develop pipelines **faster** and use **best practices**

Faster Development

Leverage file-based data right on the canvas

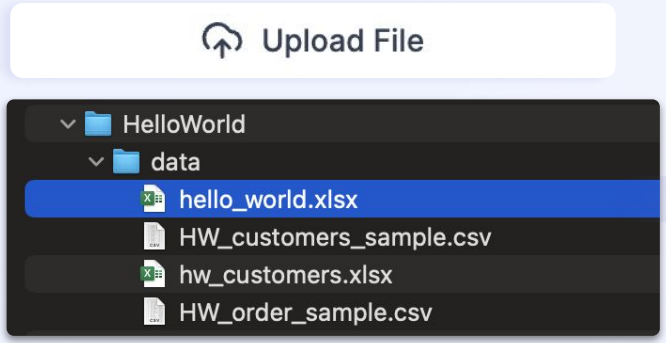
Problem

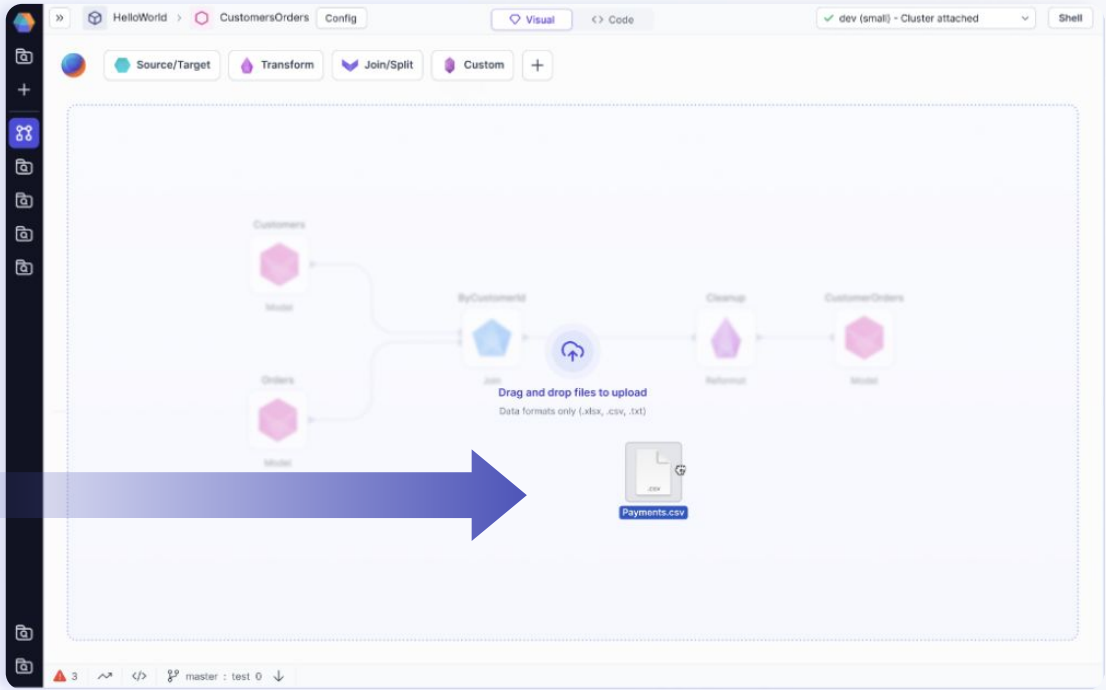
Enterprise data isn't always in enterprise platforms and it takes admins to add, using valuable time.

👉 Solution

Drag & Drop Upload docs

- CSV, JSON, XML, XLSX





Faster Development

Visual Expression Builder

Problem

SQL or Python skills are usually required to create complex business expressions

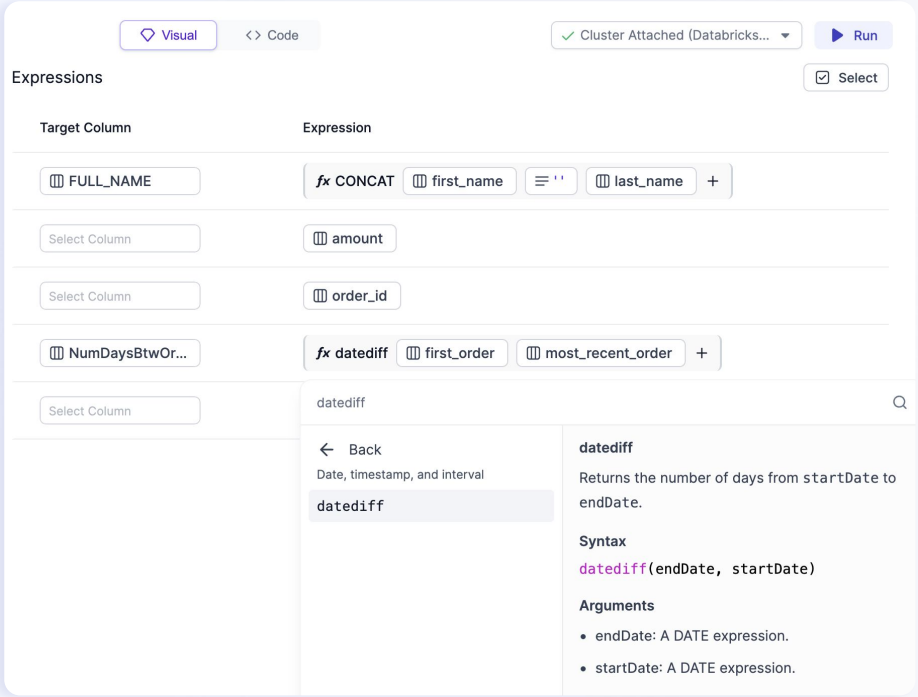
👉 Solution

Extensions to visual expression builder let everyone easily create the most complex business logic with **NO SQL or Python skills required**

Supports the most important constructs: **columns, functions, variables, and business rules**

docs

Enhanced in 3.4



Faster Development

Synthetic Data Generator

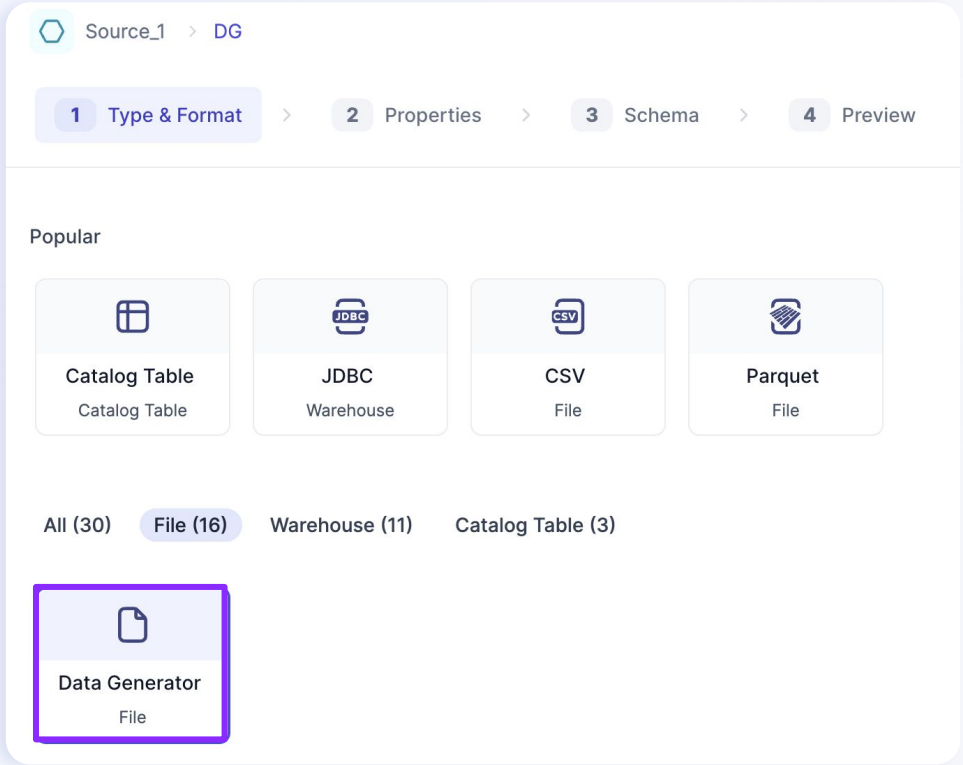
Problem

It's hard to make production-grade pipelines without access to production data and making synthetic data is a highly skilled, resource intensive task.

👉 Solution

Generate realistic sample data for development and testing of pipelines

Easily address security or privacy concerns



New in 3.4

Faster Development

Cleanup Messy Data

Problem

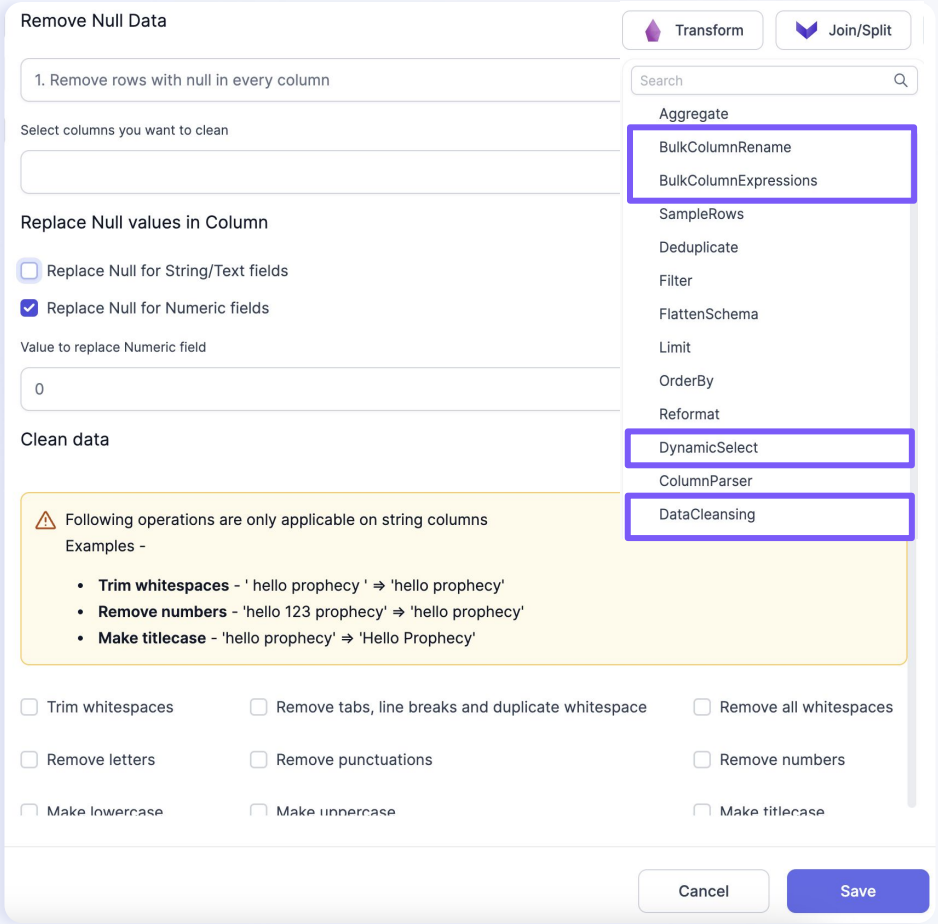
Got dozens(or hundreds!) of columns to tidy? Managing these 1 by 1 took lots of time consuming steps.

👉 Solution

Bulk actions for selected columns

- Column name - prefix/suffix
- Column data type
- Column expressions
- Column dynamic select
- Column values - find/replace

New in 3.4



Last Mile Ops

Data Delivery & Access Simplified

Problem

Data Warehouse may not be origin or destination for data and creating new destinations took valuable admin resources

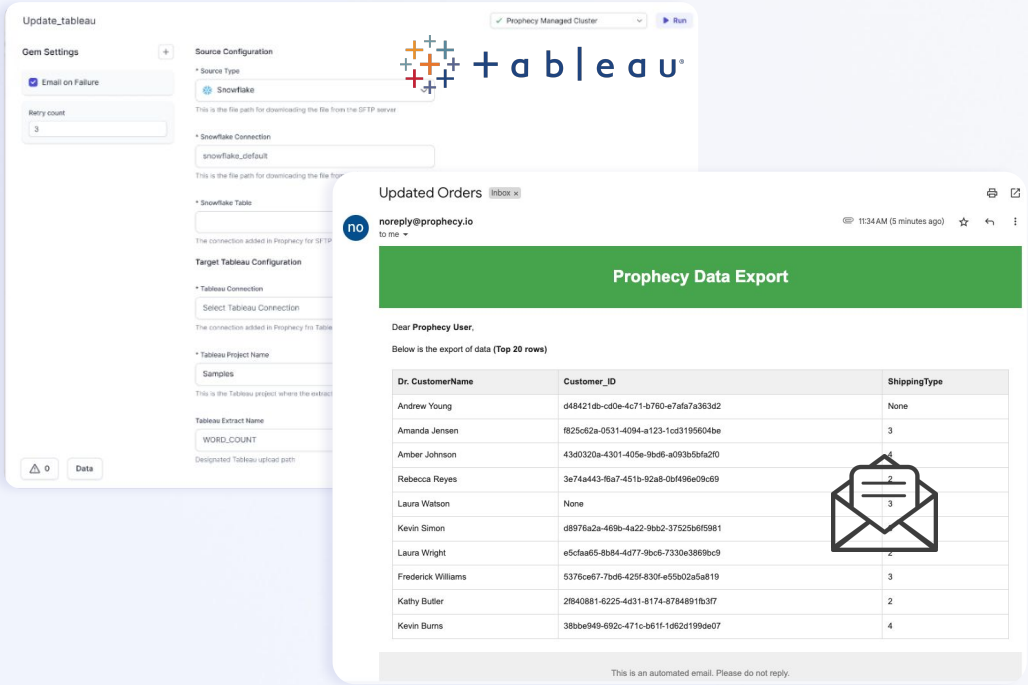
👉 Solution

New Source Gems

- Sharepoint
- SFTP

New Target Gems

- Tableau
- Email



New in 3.4

NOTE: Admins to control who can send data

Performance & Best Practices

Sophisticated write options

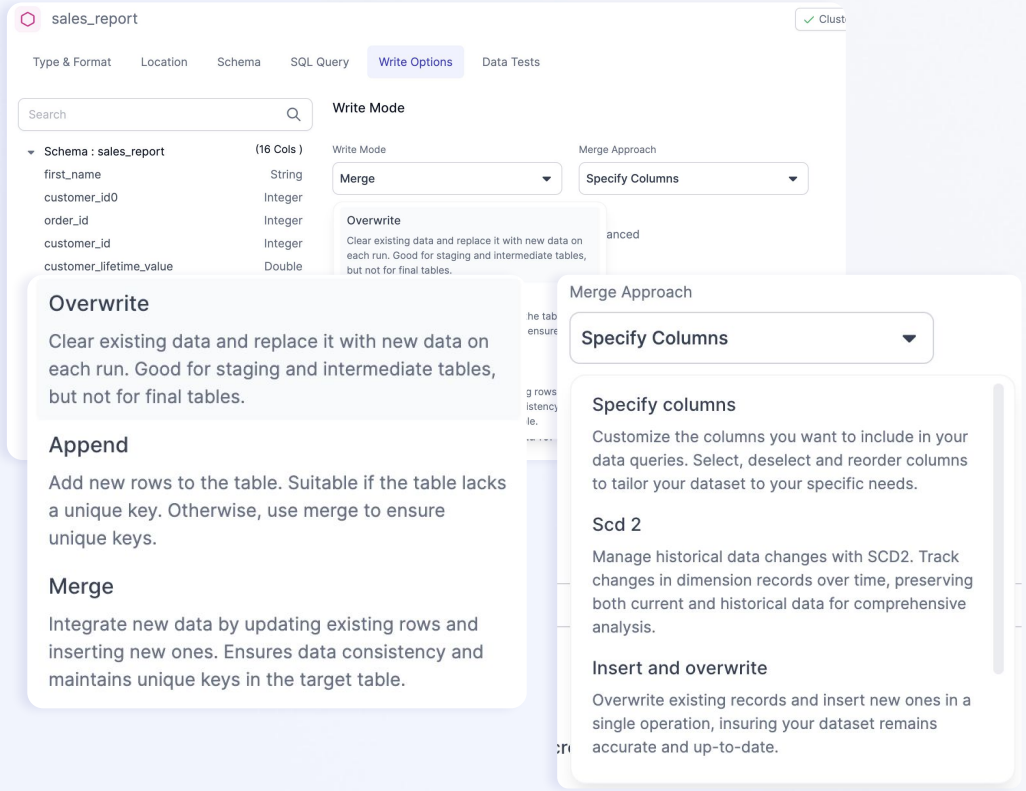
Problem

Slowly changing data **must** be tracked, but few people know how to understand & the various options had to be coded by hand.

👉 Solution

Enable configuration of sophisticated data write modes without coding

Includes simple appends and overwrites as well as complex delsert merges and slowly changing dimensions [doc]



Enhanced in 3.4

Extensibility

Custom Gems

Problem

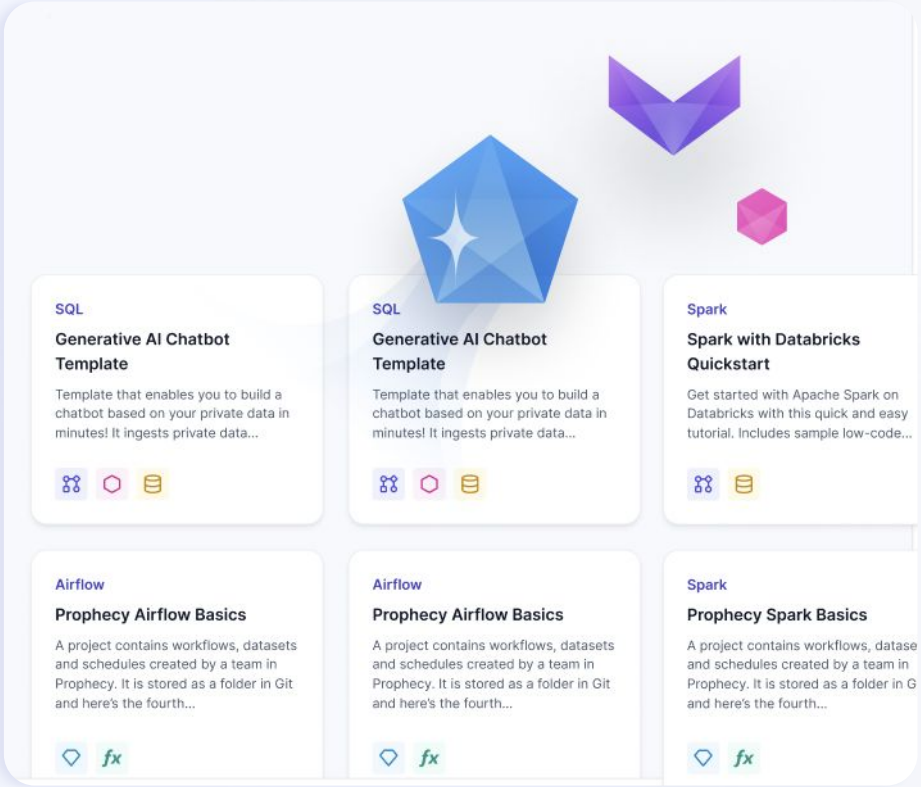
Every company has logic they'd like to standardize, version, and share. SQL users have been left out, until now.

👉 Solution

Extend Prophecy's interface with custom logic built into a new, custom Gem - now available in both Spark and SQL!

doc

Enhanced in 3.4



Pipeline observability

You can't scale if only one person can tackle issues.

Observability

Lineage Run

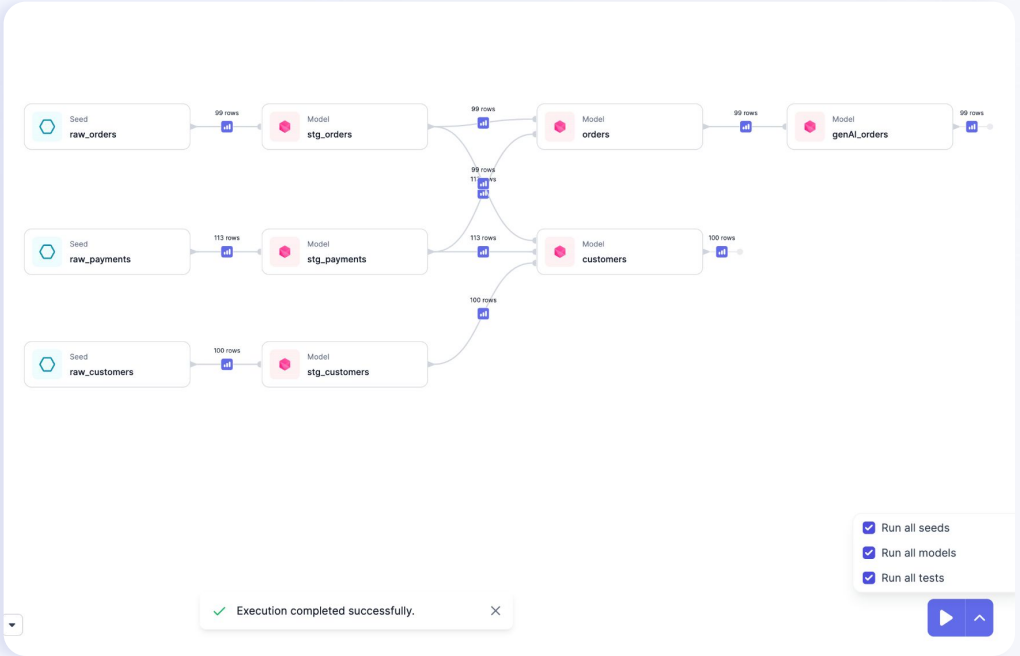
Problem

I love the lineage view for tracking data changes at the column level. Can I see the data at the model level too?

👉 Solution

Now run the project lineage to see interim data across the entire SQL project.

doc



Enhanced in 3.4

Observability

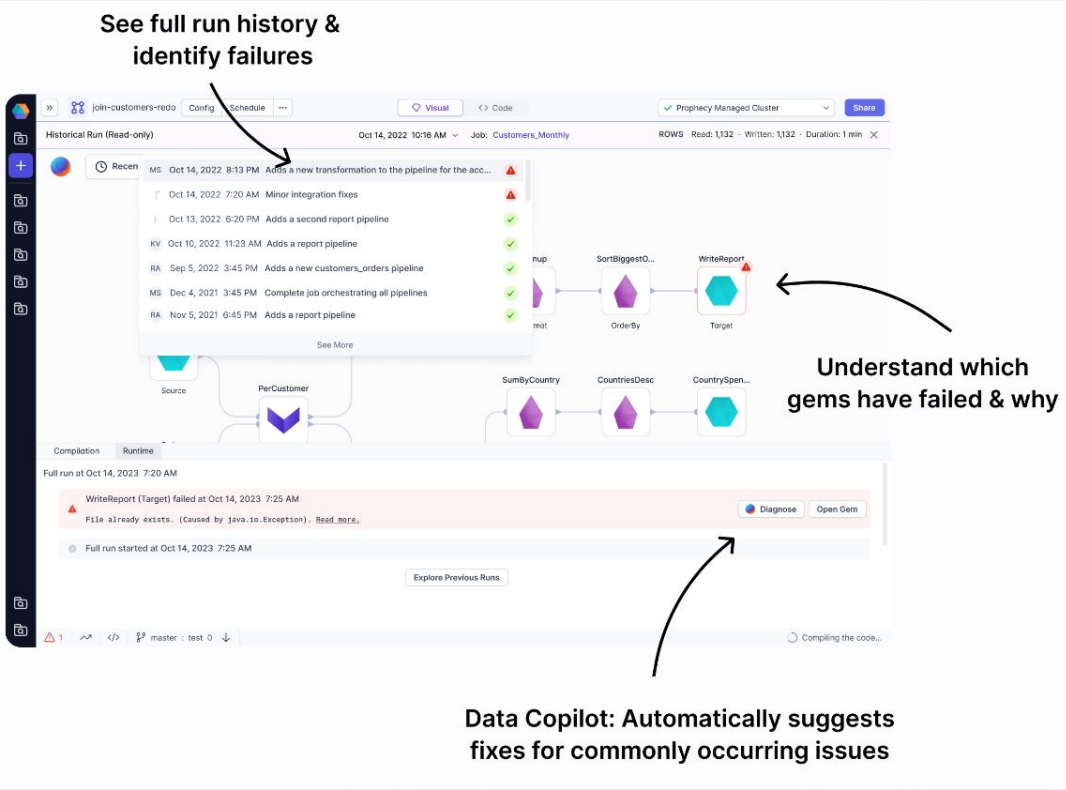
Pipeline monitoring and debugging

Problem

Debugging pipelines is hard when you lack context.

👉 Solution

We've enhanced the pipeline UI for better visibility.



Enhanced in 3.4

Observability

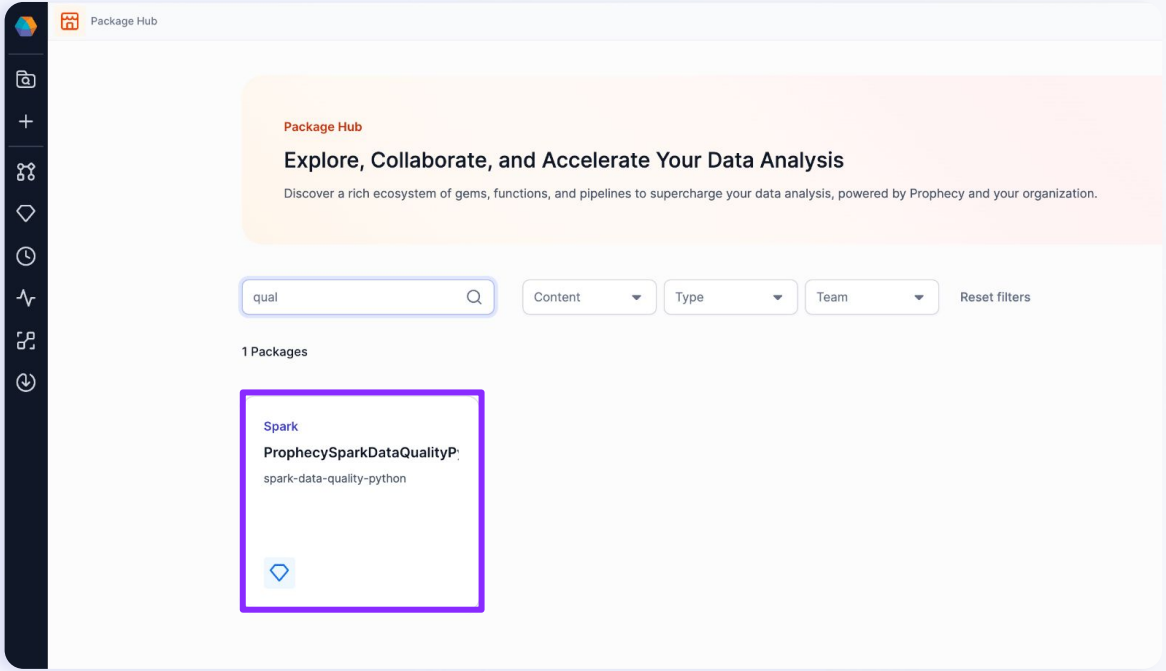
Data Quality Tests

Problem

Testing requires the ability to code and understand the data requirements.

👉 Solution

Any data practitioner can use our Data Quality Package **without knowing how to code.**



Enhanced in 3.4

Observability

Data Quality Tests

Problem

Data tests are typically accessible only to the coding user. Prophecy has long supported testing on the Spark side, and now supports Data Quality Tests on the SQL side too.

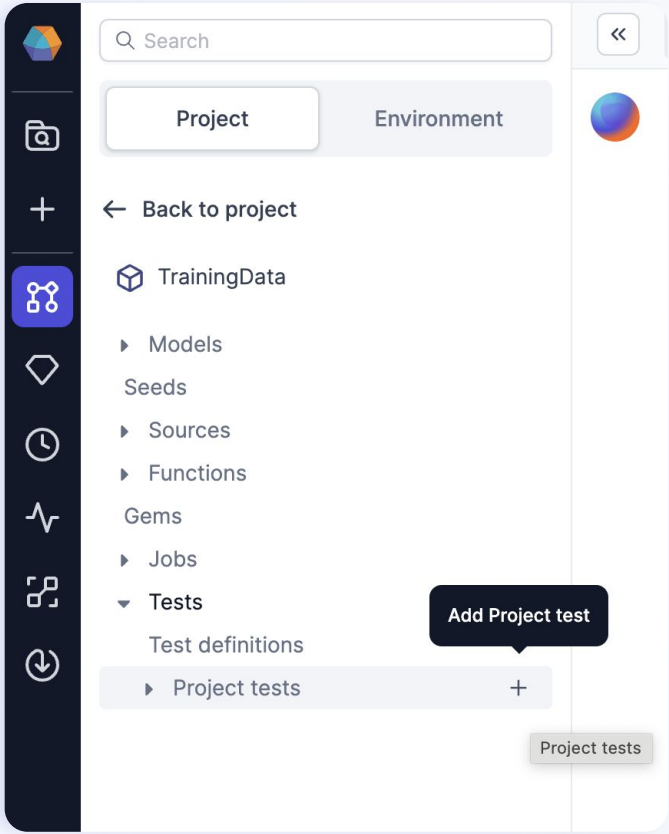
👉 Solution

Project tests: single-use tests for models

Model tests: reusable tests for models

Column tests: reusable tests for columns

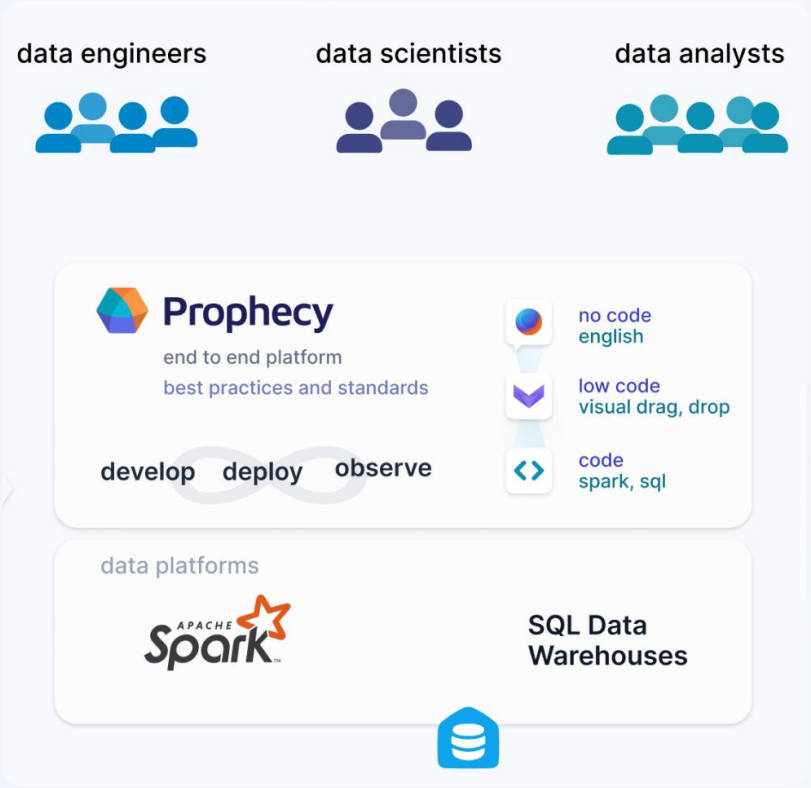
docs



Enhanced in 3.4

Highlighted innovations

1. Faster development
2. Easy observability
3. AI Capabilities



AI capabilities

Prophecy Copilot AI
Democratization & Productivity

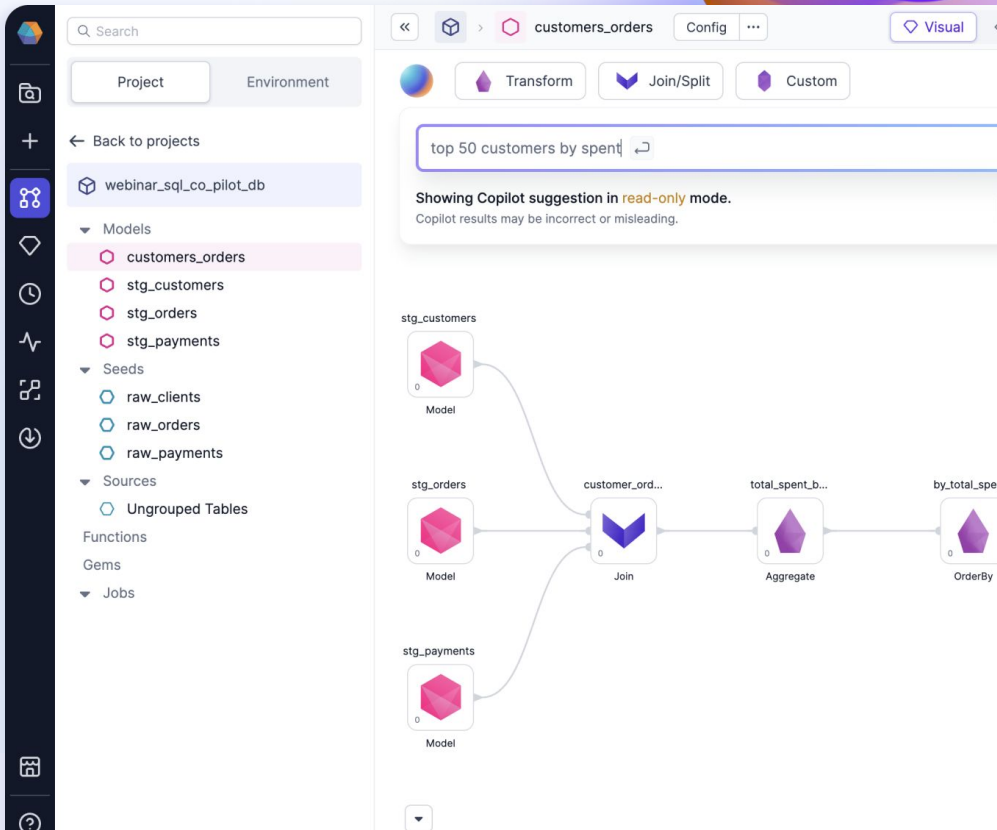
👉 English → Visual
Generate visual data pipelines, and pipeline edits from descriptions. Unlike Github co-pilot the user does not need to know coding to complete the pipeline. doc & video

👉 Predictive auto-complete
Suggest next transforms and expressions interactively. video

👉 Auto documentation
Pipeline and dataset descriptions, and auto generated commit messages reduce repetition. doc & video

👉 Automated error fixes
Suggest fixes to syntactic, semantic, and runtime errors and potential changes to modify pipelines for performance and clarity. doc & video

Enhanced in 3.4

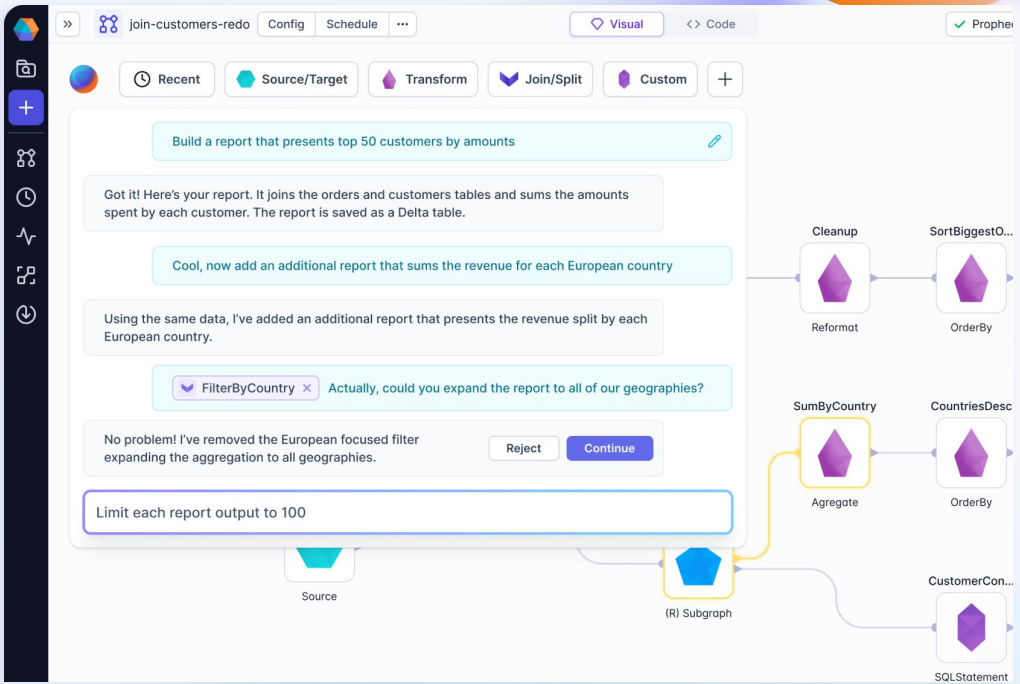


Prophecy Copilot AI
Latest innovations

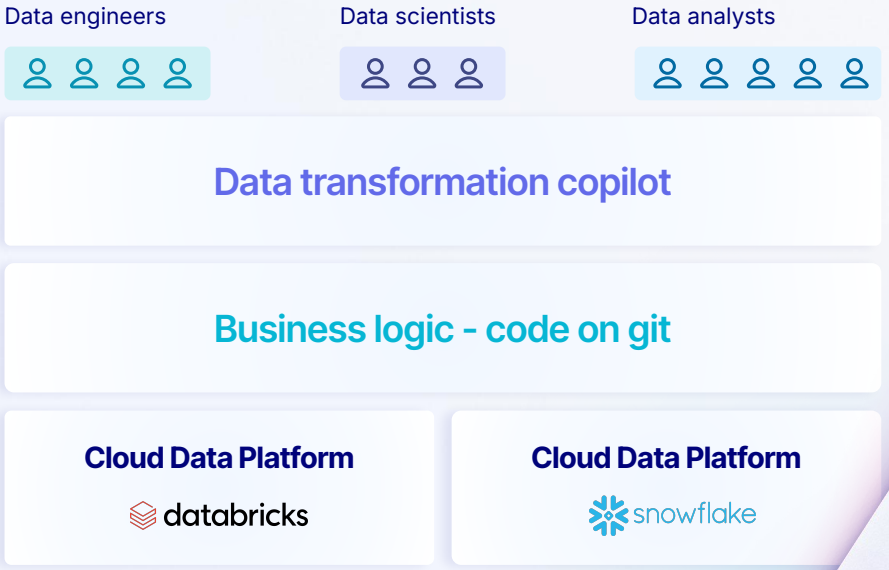
✨ Chat interface
Automatically Benefit from a more conversational experience - build on previous prompts to keep improving suggestions. video

✨ Schema mapping suggestions
Automatically suggests SQL expressions that transform the provided sources to the pre-defined required target schema. video

New in 3.4



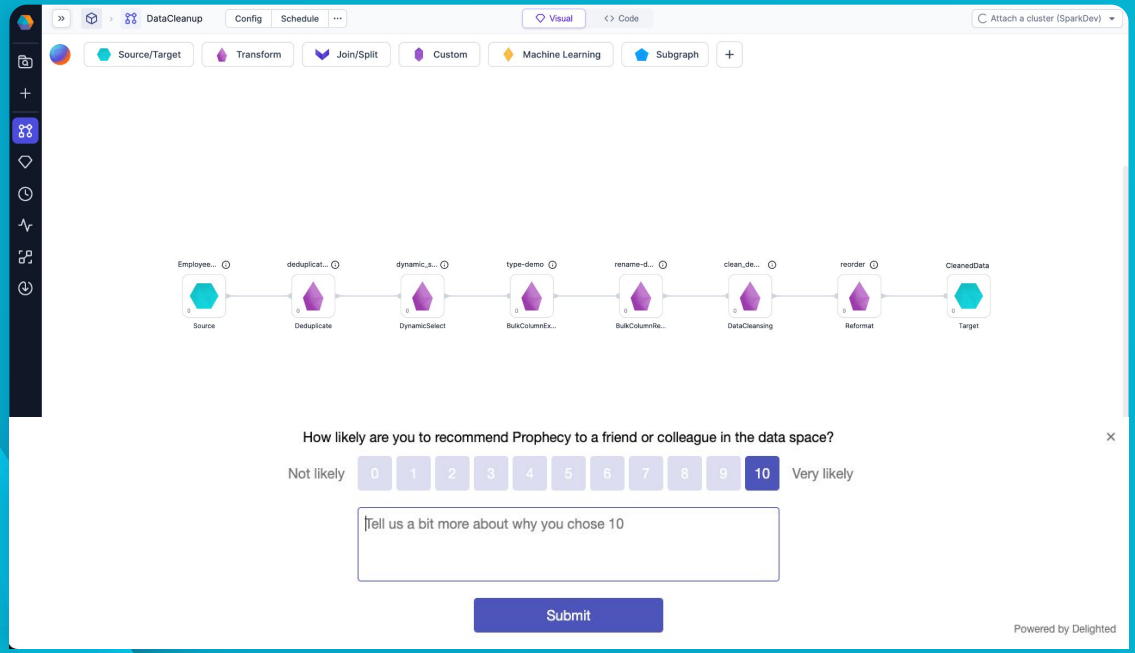
We are the productivity layer for users



Write us a comment!

As you're using the features, let us know how it's working for you

PLUS: *Gartner Peer Insights*, important for industry visibility and customers like you!



Which features will you try first?

Check them out here!

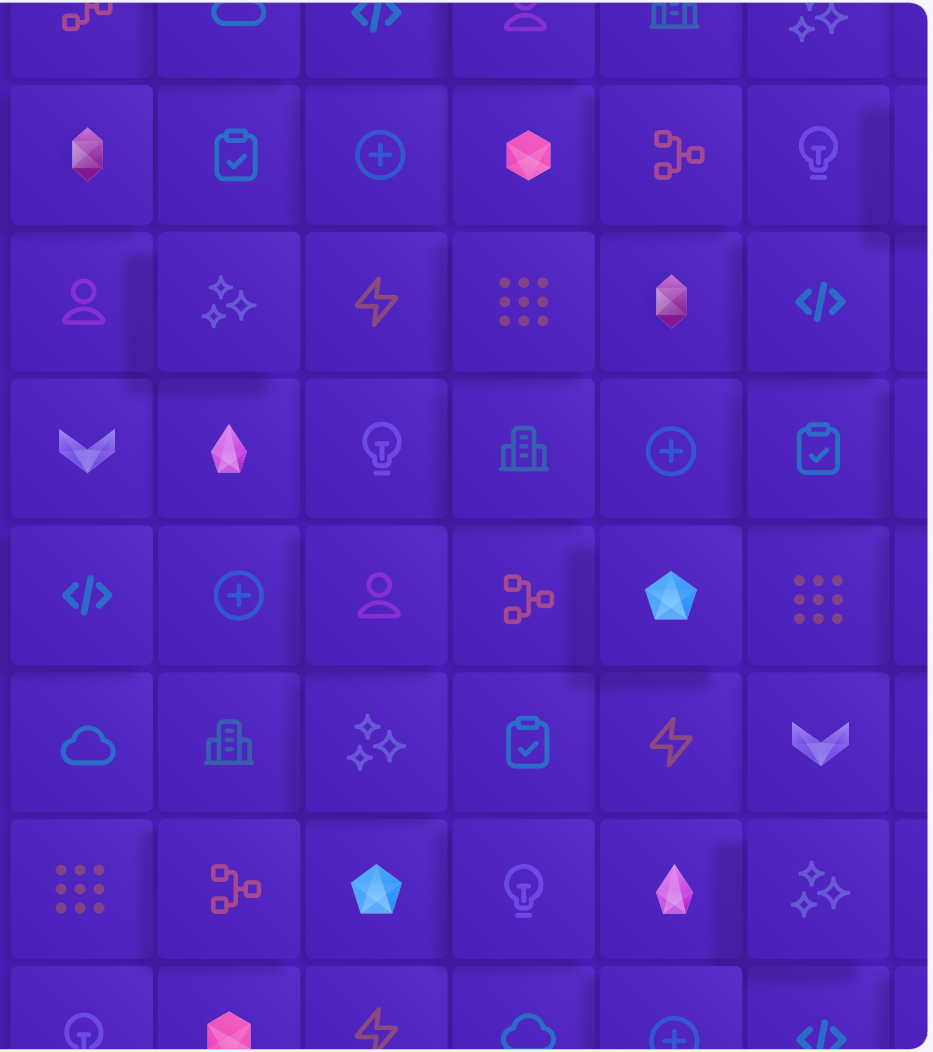
Ask your CSM to walkthrough during Office Hours

or reach out to support@prophecy.io

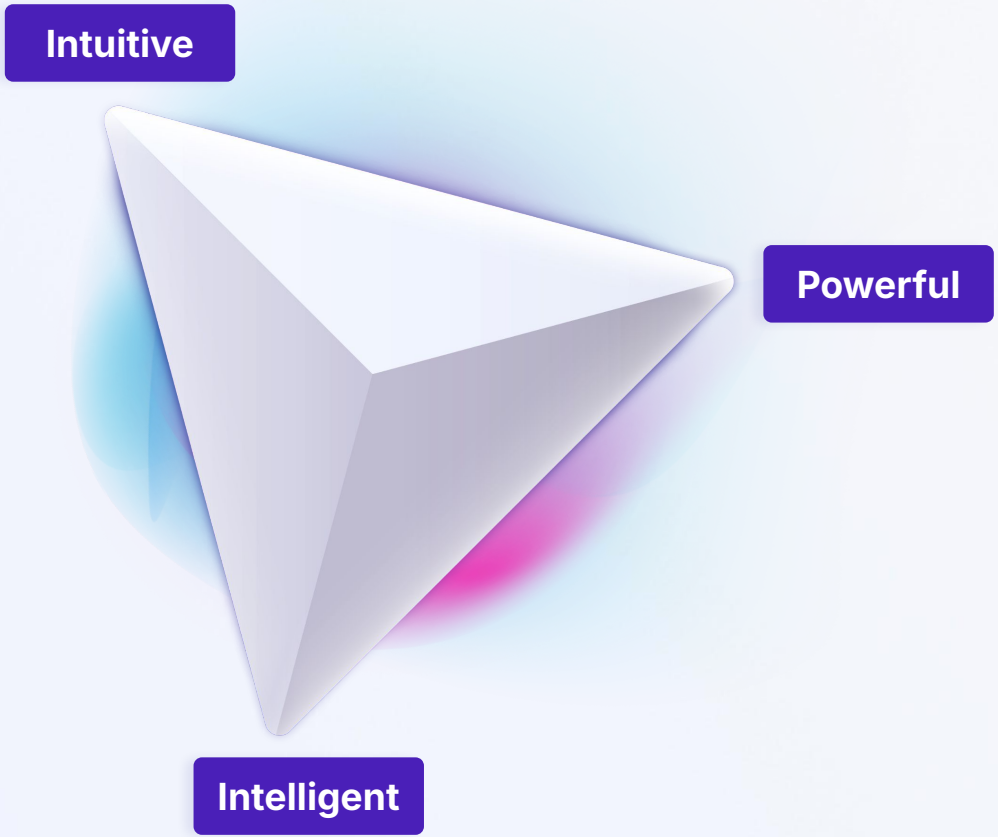
Prophecy

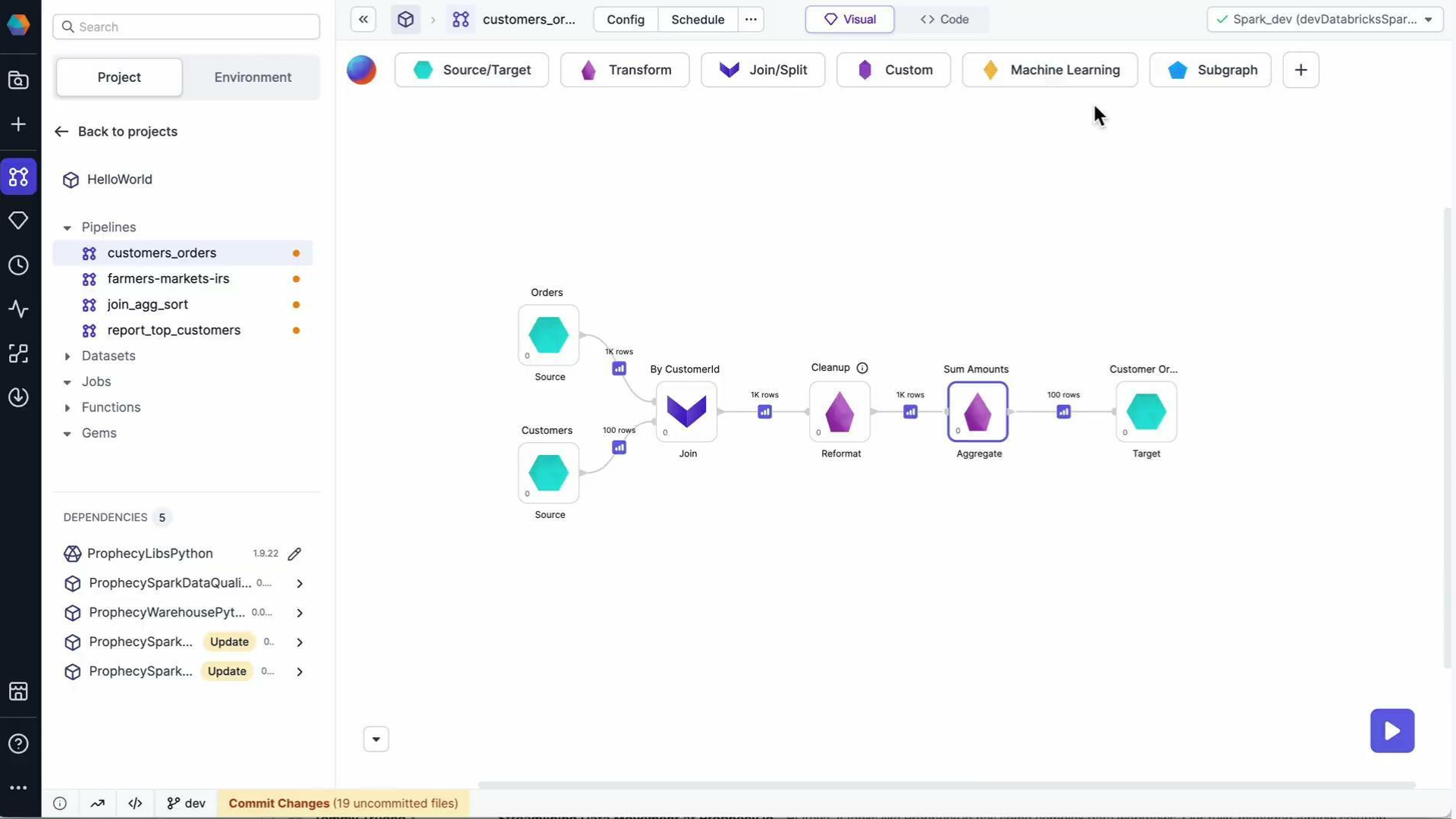
Prophecy

Appendix



The data transformation iron triangle




Search
Project
Environment
Back to projects
HelloWorld
Pipelines
customers_orders
farmers-markets-irs
join_agg_sort
report_top_customers
Datasets
Jobs
Functions
Gems
DEPENDENCIES 5
ProphecyLibsPython 1.9.22
ProphecySparkDataQuali... 0....
ProphecyWarehousePyt... 0.0...
ProphecySpark... Update 0..
ProphecySpark... Update 0...
customers_or...
Config
Schedule
Visual
Code
Spark_dev (devDatabricksSpar...
Source/Target
Transform
Join/Split
Custom
Machine Learning
Subgraph
Orders
Source
1K rows
Customers
Source
100 rows
By CustomerId
Join
1K rows
Cleanup
Reformat
1K rows
Sum Amounts
Aggregate
100 rows
Customer Or...
Target
dev
Commit Changes (19 uncommitted files)

HelloWorld_SQL

by anya+analyst@prophecy.io_team

Open in Editor

About | Content | Commits | Releases & Deployments | Code | Access | Settings

Testing dbt project: `jaffle_shop`

`jaffle_shop` is a fictional ecommerce store. This dbt project transforms raw data from an app database into a customers and orders model ready for analytics.

What is this repo?

What this repo *is* :

- A self-contained playground dbt project, useful for testing out scripts, and communicating some of the core dbt concepts.

What this repo *is not* :

- A tutorial — check out the Getting Started Tutorial for that. Notably, this repo contains some anti-patterns to make it self-contained, namely the use of seeds instead of sources.
- A demonstration of best practices — check out the dbt Learn Demo repo instead. We want to keep this project as simple as possible. As such, we chose not to implement:
  - our standard file naming patterns (which make more sense on larger projects, rather than this five-model project)
  - a pull request flow
  - CI/CD integrations
- A demonstration of using dbt for a high-complex project, or a demo of advanced features (e.g. macros, packages, hooks, operations) — we're just trying to keep things simple here!

What's in this repo?

About

No Description

Language

Sql | Snowflake

Content

Models (5) | Seeds (3) | Sources (1)

Releases

Not yet released

Created: 3 months ago by Anya Bida

Data Tests for consistency

Custom Data Tests

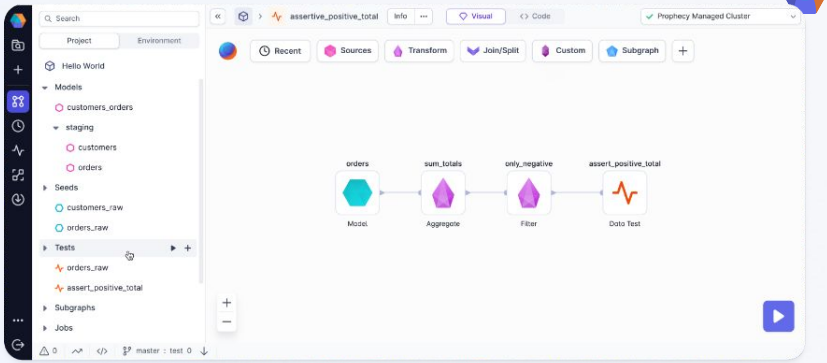
Full SQL queries that can contain arbitrary logic and after execution must satisfy a passing condition to mark data test as succesfull ones. (dbt term: *singular*)

Generic Data Tests

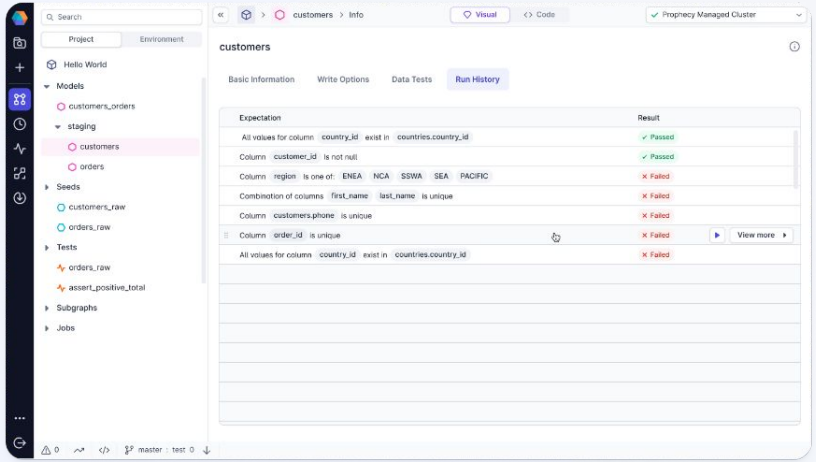
Dbt macro generated queries that can be parametrized and applied to a given model or a (set of) columns(s).

Dbt Core distinguishes between two generic test types:

1. **column-level tests**, which are defined on a column level of each model
    a. e.g. *not_null*, *unique*, *dbt_utils.not_empty_string*, etc
2. **model-level tests**, which can span across many columns for a given model, or even multiple models, and are defined at a model level



Singular Data Test - Manual tests at a project-level



Generic Data Test - Generated tests at model & column-levels

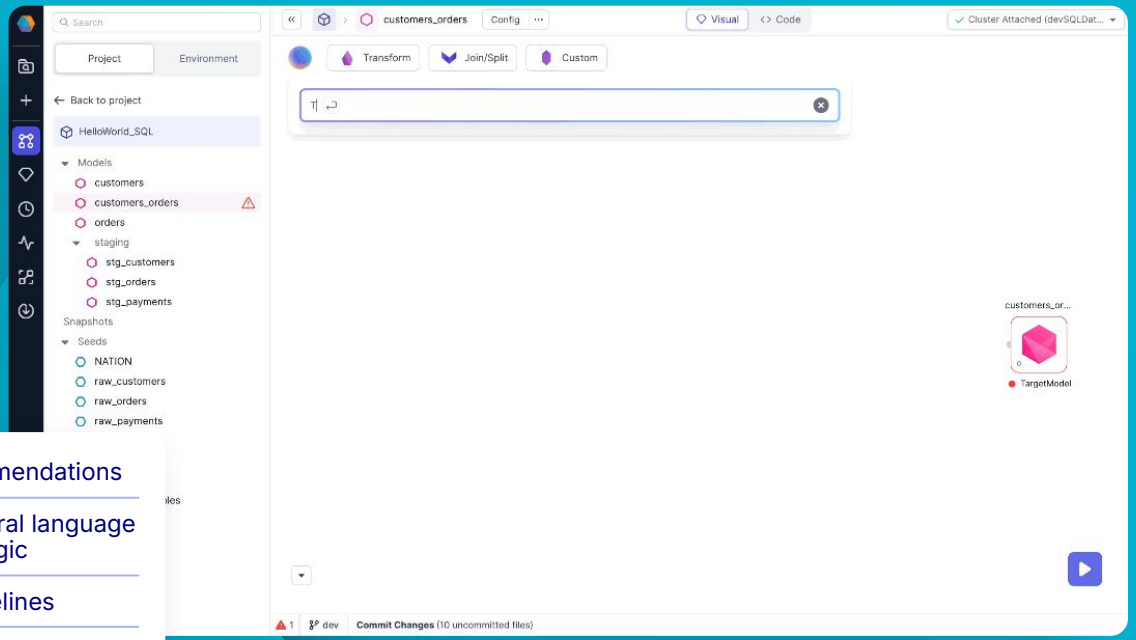
SQL

New in 3.4

More productivity for every team member

Higher productivity per user

Focus on analytics



- Makes recommendations
- Converts natural language to business logic
- Complete pipelines
- Generate tests
- Writes documentation
- Suggests fixes for errors

Prophecy

Prophecy 3.4

New features to turbocharge pipeline development

Exclusive Webinar:
Nov 13 at 9am PT
Nov 13 at 12pm GMT

Bob Welshmer
Senior Sales Engineer
Prophecy

Anya Bida
Technical Evangelist
Prophecy

Kuldeep Singh
AI Architect
Prophecy

Enable every user

Ease of use

Remove barriers

High productivity

Low code

Drag & drop

Spark & SQL