



New features to turbocharge pipeline development



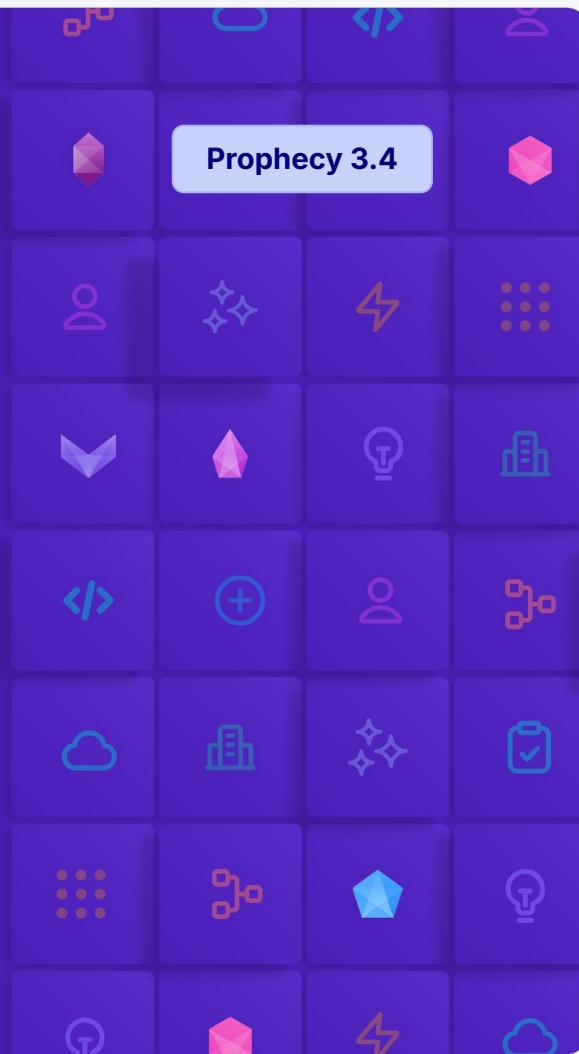
Bob Welshmer
Senior Sales Engineer
Prophecy



Anya Bida
Technical Evangelist
Prophecy



Kuldeep Singh
AI Architect
Prophecy



Productivity wins! Swag wins!





Welcome
Data Engineers,
Data Analysts,
Data Leaders!



Prophecy



New features to turbocharge pipeline development



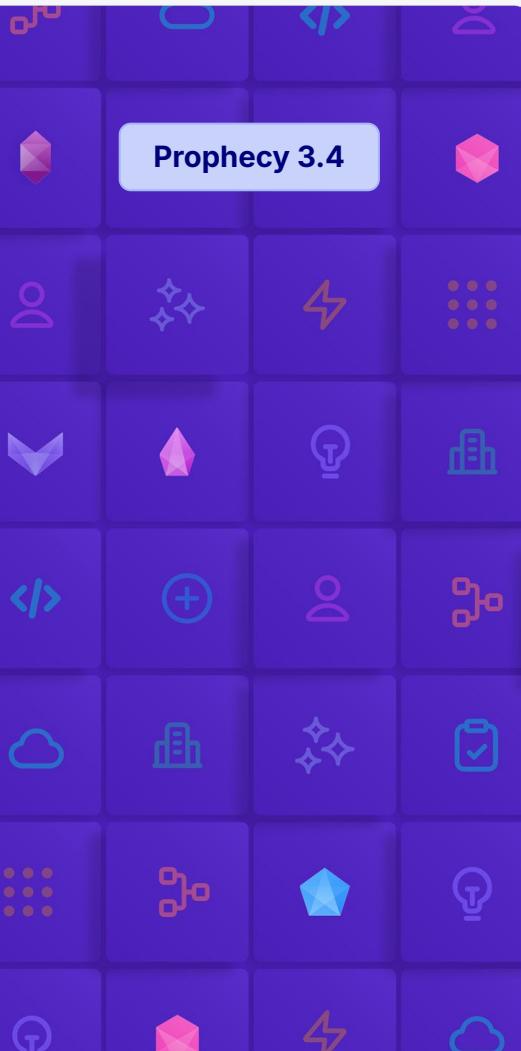
Bob Welshmer
Senior Sales Engineer
Prophecy



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 engineers



data scientists



data analysts



Prophecy

end to end platform
best practices and standards

develop deploy observe



no code
english



low code
visual drag, drop



code
spark, sql

data platforms



SQL Data
Warehouses





Data pipelines
take too long to build.



Development made better

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

Leverage file-based data right on the canvas

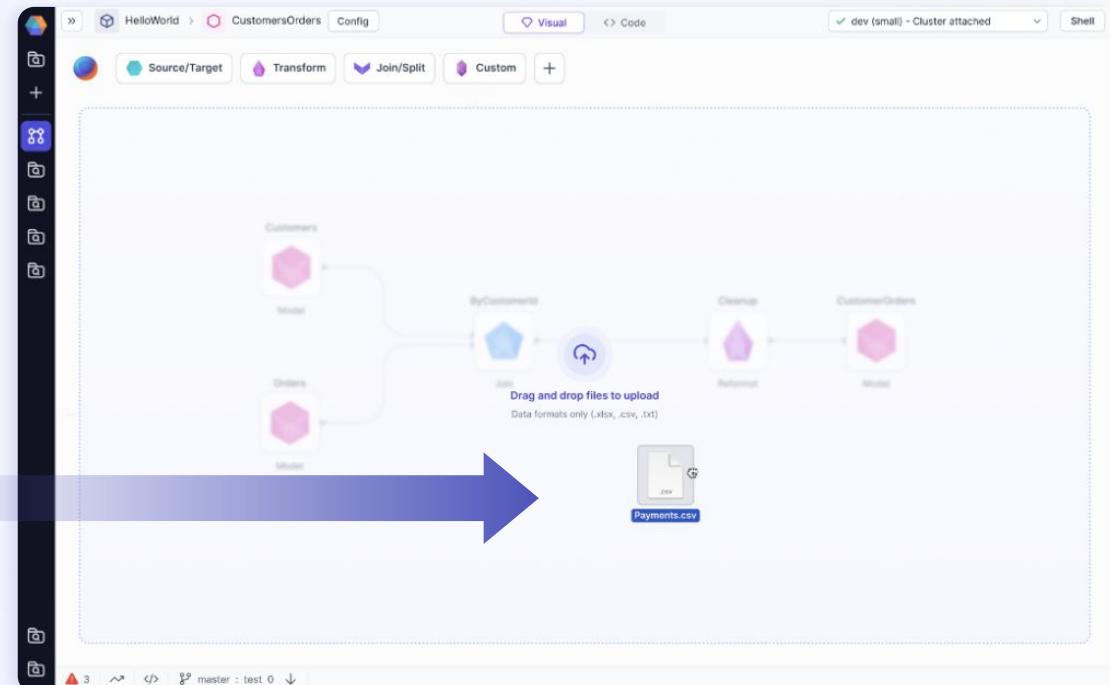
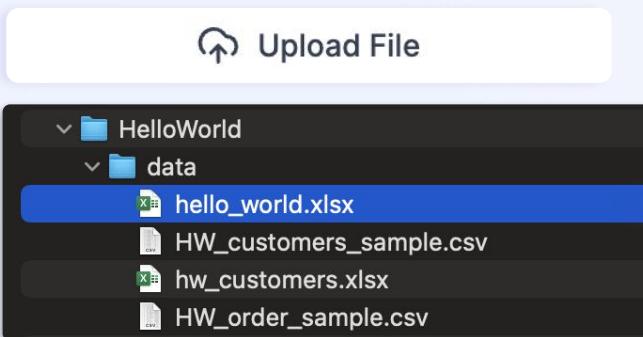
Problem

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



Visual Expression Builder

Problem

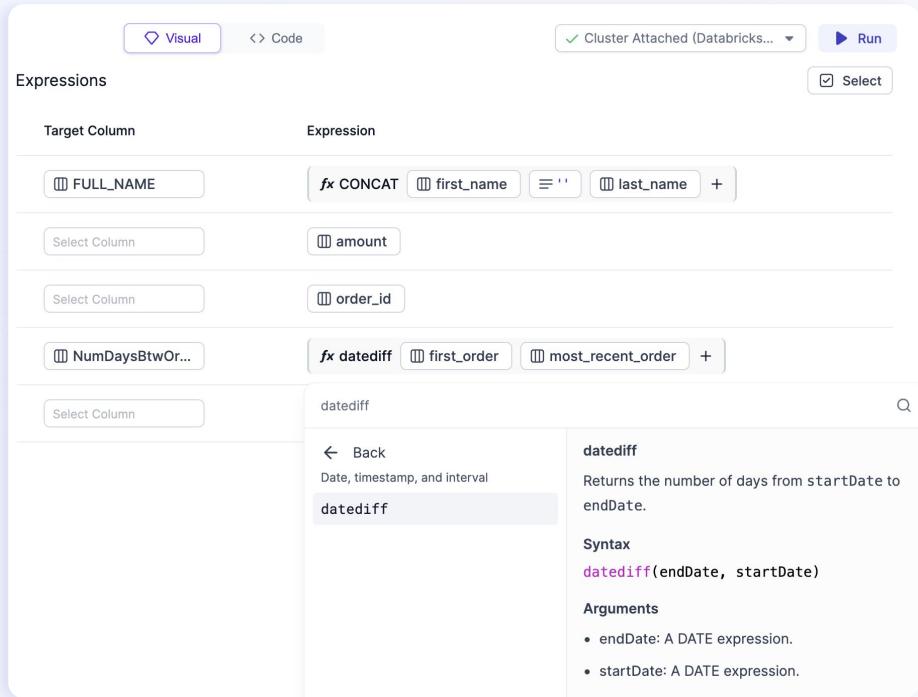
SQL or Python skills are usually required to create complex business expressions

👉 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](#)



The screenshot shows the Visual Expression Builder interface. At the top, there are tabs for "Visual" (selected), "Code", and "Run". To the right are buttons for "Cluster Attached (Databricks...)" and "Run". Below these are sections for "Expressions" and "Functions".

Expressions:

- Target Column: FULL_NAME, Expression: fx CONCAT (first_name || ' ' || last_name) +
- Select Column: amount
- Select Column: order_id
- Select Column: NumDaysBtwOr..., Expression: fx datediff (first_order, most_recent_order) +

Functions:

- datediff**:
 - Description: Returns the number of days from startDate to endDate.
 - Syntax: datediff(endDate, startDate)
 - Arguments:
 - endDate: A DATE expression.
 - startDate: A DATE expression.

Enhanced in 3.4

Synthetic Data Generator

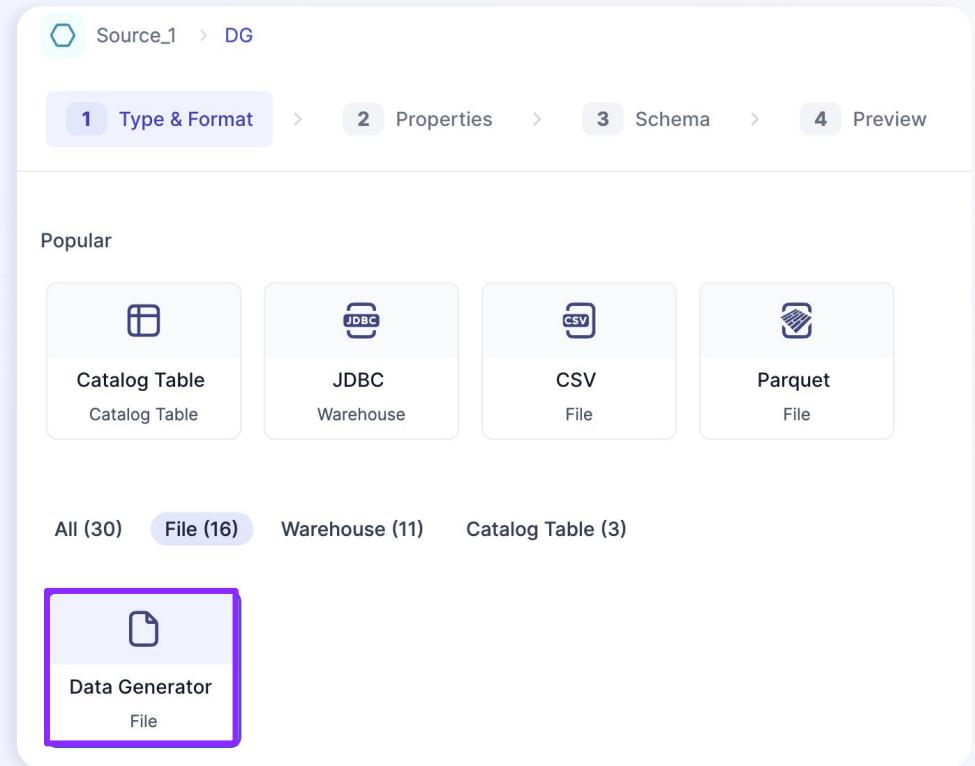
Problem

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



The screenshot shows the Apache Spark UI interface for generating synthetic data. At the top, there is a navigation bar with four tabs: 1 Type & Format, 2 Properties, 3 Schema, and 4 Preview. Below the navigation bar, the title "Source_1 > DG" is displayed. A section titled "Popular" contains four cards: Catalog Table (Catalog Table), JDBC (Warehouse), CSV (File), and Parquet (File). Below these cards, there are four buttons: All (30), File (16) (which is highlighted in blue), Warehouse (11), and Catalog Table (3). At the bottom, there is a card for "Data Generator" (File), which is highlighted with a purple border.

New in 3.4

Cleanup Messy Data

Problem

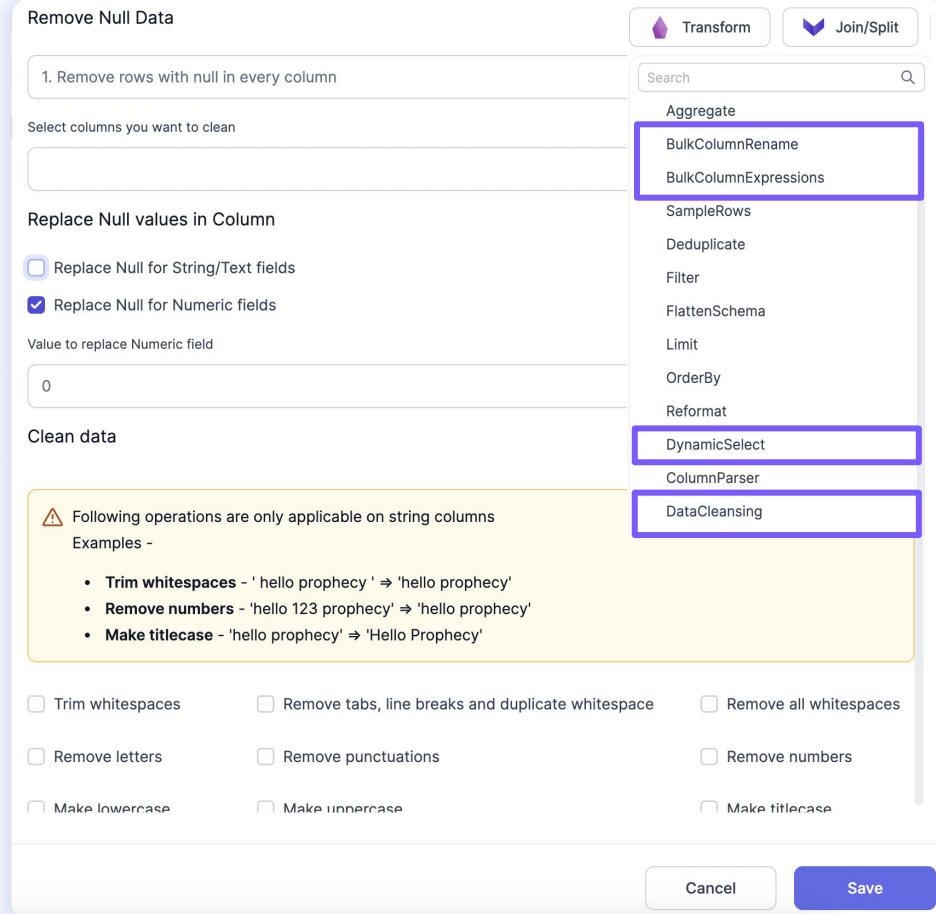
Got dozens(or hundreds!) of columns to tidy?
 Managing these 1 by 1 took lots of time
 consuming steps.

👉 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



The screenshot shows the Apache Spark UI interface. At the top, there are two buttons: 'Transform' (purple icon) and 'Join/Split' (blue icon). Below them is a search bar with the placeholder 'Search'. A sidebar on the right lists various data processing operations: Aggregate, BulkColumnRename, BulkColumnExpressions, SampleRows, Deduplicate, Filter, FlattenSchema, Limit, OrderBy, Reformat, DynamicSelect, ColumnParser, and DataCleansing. The 'DataCleansing' option is highlighted with a purple rectangle. The main area of the window is titled 'Remove Null Data' and contains a step: '1. Remove rows with null in every column'. Below this is a section 'Replace Null values in Column' with two checkboxes: 'Replace Null for String/Text fields' (unchecked) and 'Replace Null for Numeric fields' (checked). A field 'Value to replace Numeric field' contains the value '0'. The next section is 'Clean data', which includes a warning message: '⚠ Following operations are only applicable on string columns Examples -' followed by three bullet points: 'Trim whitespaces - 'hello prophecy' ⇒ 'hello prophecy'', 'Remove numbers - 'hello 123 prophecy' ⇒ 'hello prophecy'', and 'Make titlecase - 'hello prophecy' ⇒ 'Hello Prophecy''. Below this are several checkboxes for cleaning string columns: 'Trim whitespaces', 'Remove tabs, line breaks and duplicate whitespace', 'Remove all whitespaces', 'Remove letters', 'Remove punctuations', 'Remove numbers', 'Make lowercase', 'Make uppercase', and 'Make titlecase'. At the bottom right are 'Cancel' and 'Save' buttons.

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

The screenshot displays the Prophecy Data Export interface. On the left, the 'Update_tableau' configuration screen shows settings for 'Gem Settings' (Email on Failure, Retry count 3), 'Source Configuration' (Snowflake source type, connection 'snowflake_default'), and 'Target Tableau Configuration' (Tableau Connection, Project Name 'Samples', extract name 'WORD_COUNT'). On the right, an email titled 'Updated Orders' is shown in the inbox, with the subject 'Prophecy Data Export'. The email body contains a table titled 'Prophecy User' with 20 rows of data, and an icon of an envelope with four arrows pointing outwards.

Dr. CustomerName	Customer_ID	ShippingType
Andrew Young	d48421db-cd0c-4c71-b76e-e7af7a783642	None
Amara Jensen	f825cd2a-0531-4094-a123-1cd3195604be	3
Amber Johnson	43d0320a-4301-405e-9b68-a093b5bfaf70	
Rebecca Reyes	3e74a443-f6a7-451b-92a8-0b49496e09c9	
Laura Watson	None	
Kevin Simon	d8976a2a-46fb-4a22-9bb2-3752b6f5981	
Laura Wright	e5cfaa65-bb84-4d77-9bc5-7330e3869bc9	2
Frederick Williams	5376ce67-7bde-425f-830e-e55b02a5a819	3
Kathy Butler	2f840881-6225-4d31-8174-8784891fb3f7	2
Kevin Burns	388be949-892c-471c-b611-1692d199de07	4

New in 3.4

NOTE: Admins to control who can send data

Sophisticated write options

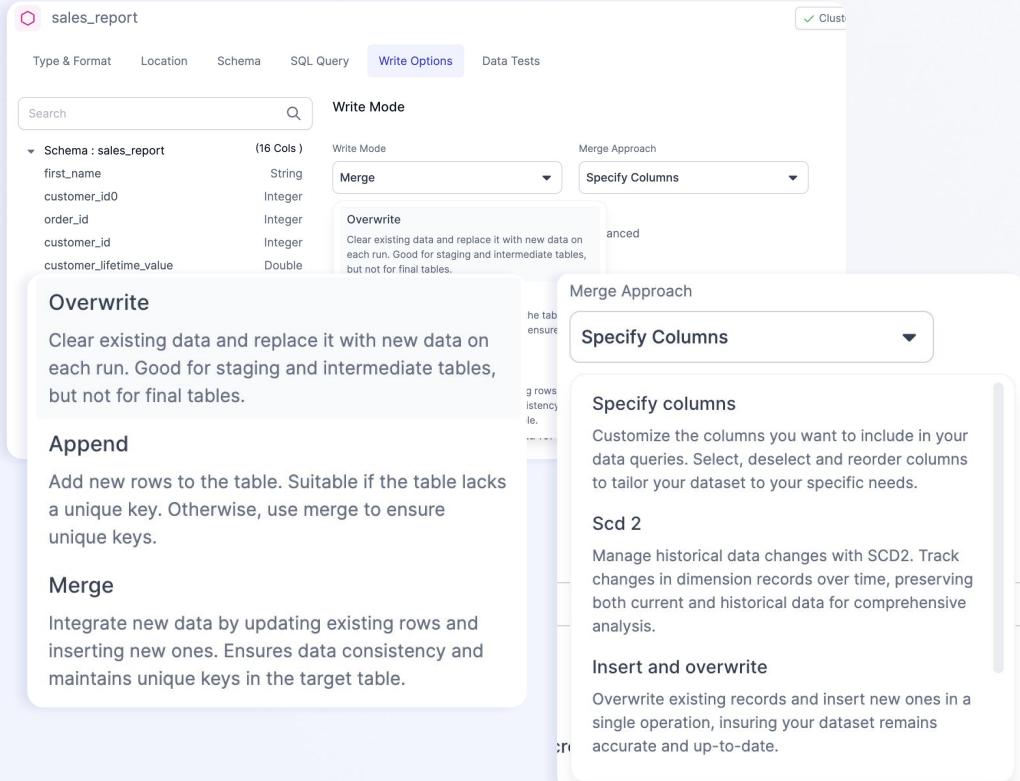
Problem

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](#)



The screenshot shows a data management interface for a table named "sales_report". The top navigation bar includes tabs for Type & Format, Location, Schema, SQL Query, Write Options (which is selected), and Data Tests. A cluster status icon indicates "Cluster healthy".

In the "Write Options" tab, the "Write Mode" dropdown is set to "Merge". The "Merge Approach" dropdown is set to "Specify Columns".

The "Overwrite" section describes replacing existing data with new data on each run, suitable for staging and intermediate tables. It includes a note about clearing data before each run.

The "Append" section describes adding new rows to the table, suitable if it lacks a unique key. It notes that merge is used to ensure unique keys.

The "Merge" section describes integrating new data by updating existing rows and inserting new ones, ensuring data consistency and maintaining unique keys.

The "Merge Approach" section is expanded, showing the "Specify Columns" dropdown.

The "Specify columns" section allows users to customize columns for data queries, selecting, deselecting, and reordering them.

The "Scd 2" section describes managing historical data changes with SCD2, tracking changes in dimension records over time.

The "Insert and overwrite" section describes overwriting existing records and inserting new ones in a single operation to keep the dataset accurate and up-to-date.

Enhanced in 3.4

Custom Gems

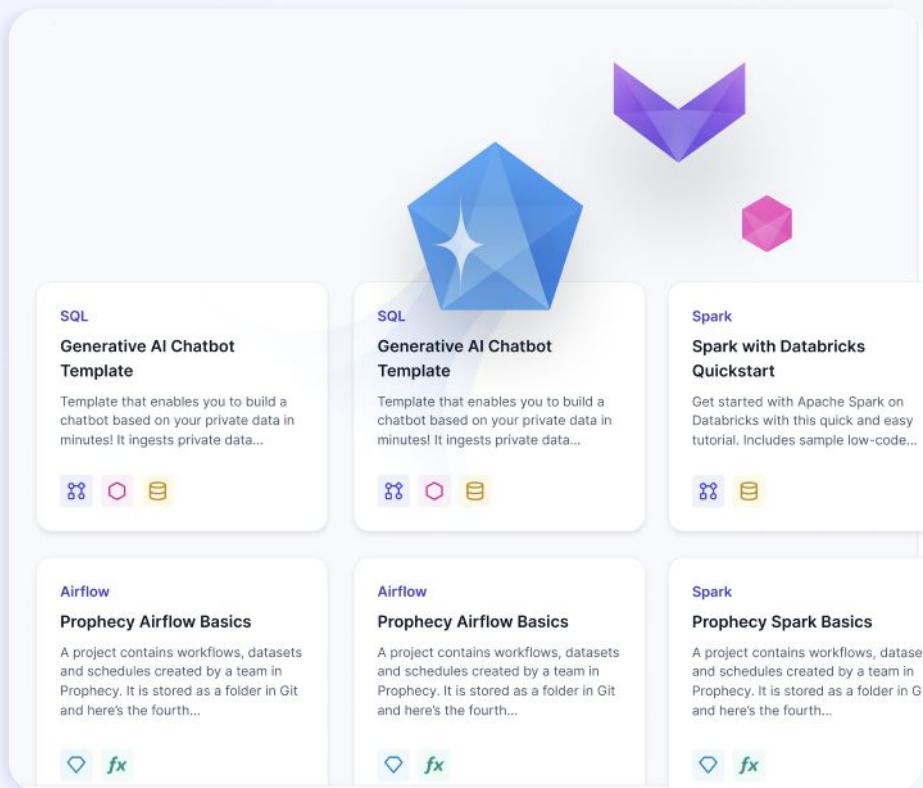
Problem

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](#)



The screenshot displays the Prophecy interface with a grid of six cards, each representing a different custom Gem template:

- SQL Generative AI Chatbot Template**: A template for building a chatbot based on private data in minutes, ingestible via a CSV file.
- SQL Generative AI Chatbot Template**: Another template for building a chatbot based on private data in minutes, ingestible via a CSV file.
- Spark with Databricks Quickstart**: A quickstart guide for getting started with Apache Spark on Databricks, including sample low-code... (truncated)
- Airflow Prophecy Airflow Basics**: A project containing workflows, datasets, and schedules created by a team in Prophecy, stored as a folder in Git.
- Airflow Prophecy Airflow Basics**: Another project containing workflows, datasets, and schedules created by a team in Prophecy, stored as a folder in Git.
- Spark Prophecy Spark Basics**: A project containing workflows, datasets, and schedules created by a team in Prophecy, stored as a folder in Git.

Each card includes a preview icon at the bottom.

Enhanced in 3.4



Pipeline observability



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



Lineage Run

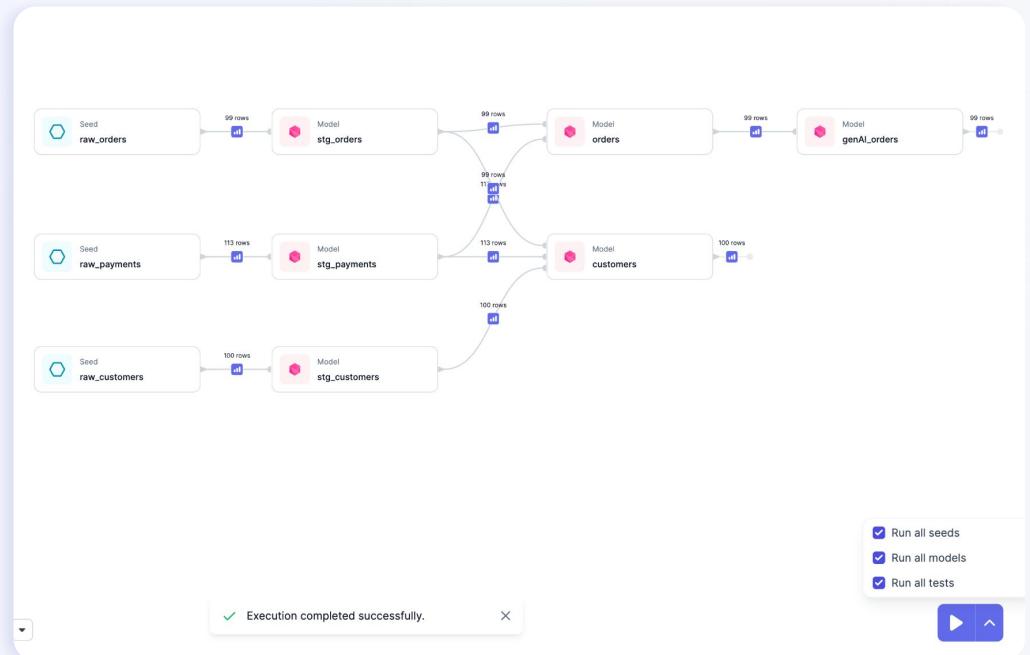
Problem

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

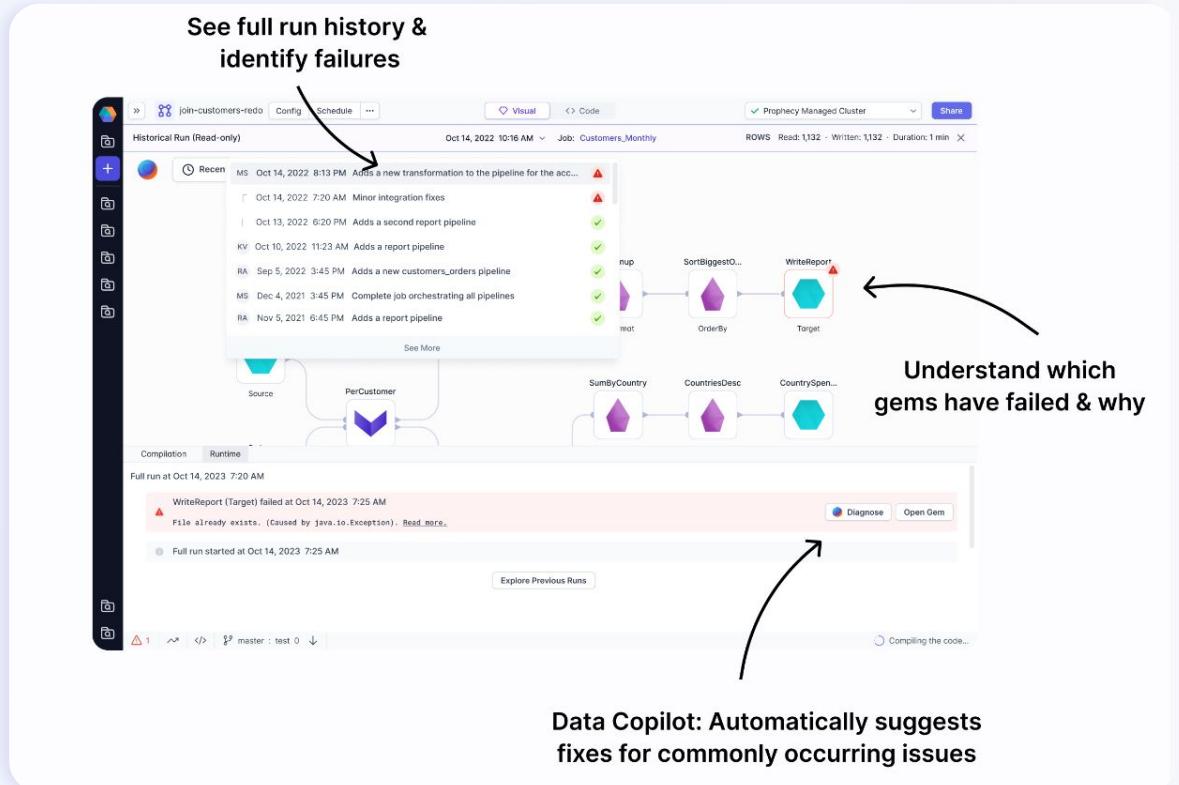
Pipeline monitoring and debugging

Problem

Debugging pipelines is hard when you lack context.

👉 Solution

We've enhanced the pipeline UI for better visibility.



The screenshot displays a pipeline monitoring interface with several key features highlighted:

- Historical Run (Read-only):** Shows a timeline of recent pipeline changes and additions, such as "Oct 14, 2022 8:13 PM Adds a new transformation to the pipeline for the acc..." and "Oct 14, 2022 7:20 AM Minor integration fixes".
- Pipeline Graph:** A visual representation of the data flow, starting from a "Source" node, followed by "PerCustomer", "SumByCountry", "CountriesDesc", and finally "WriteReport" (Target). Nodes are color-coded (green for successful, red for failed).
- Failure Details:** A callout points to a specific failure: "WriteReport (Target) failed at Oct 14, 2023 7:25 AM File already exists. (Caused by java.io.Exception). Read more...".
- Data Copilot:** A callout points to a section titled "Data Copilot: Automatically suggests fixes for commonly occurring issues".
- Code Editor:** At the bottom, a code editor shows a snippet of Scala code related to the pipeline.

Enhanced in 3.4

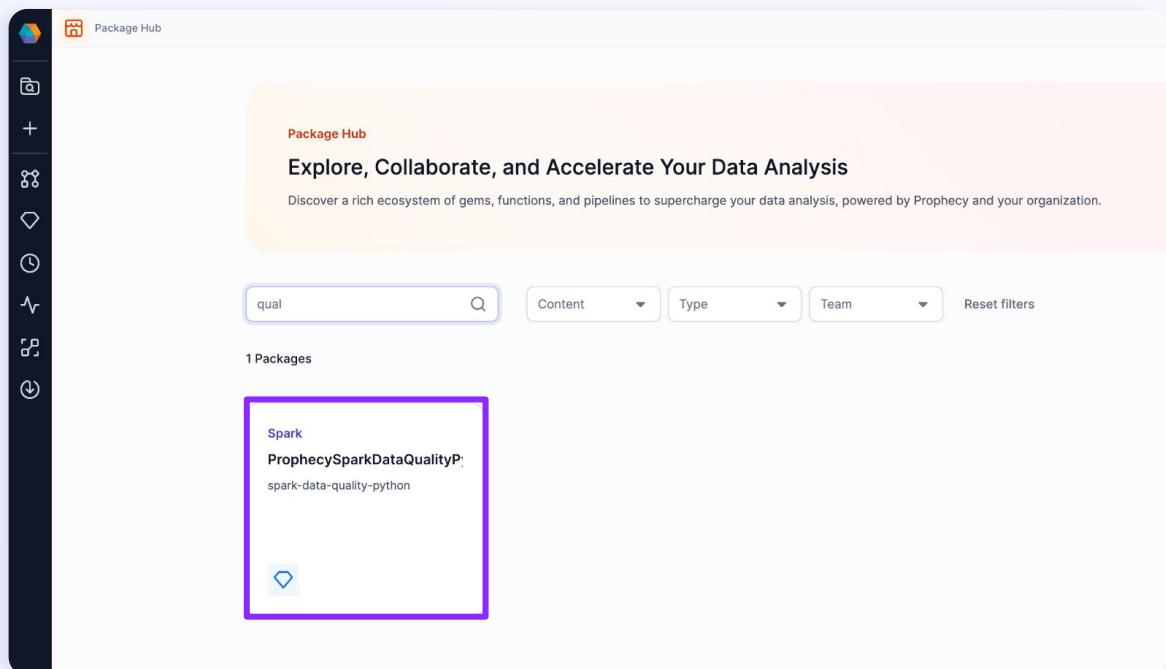
Data Quality Tests

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

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

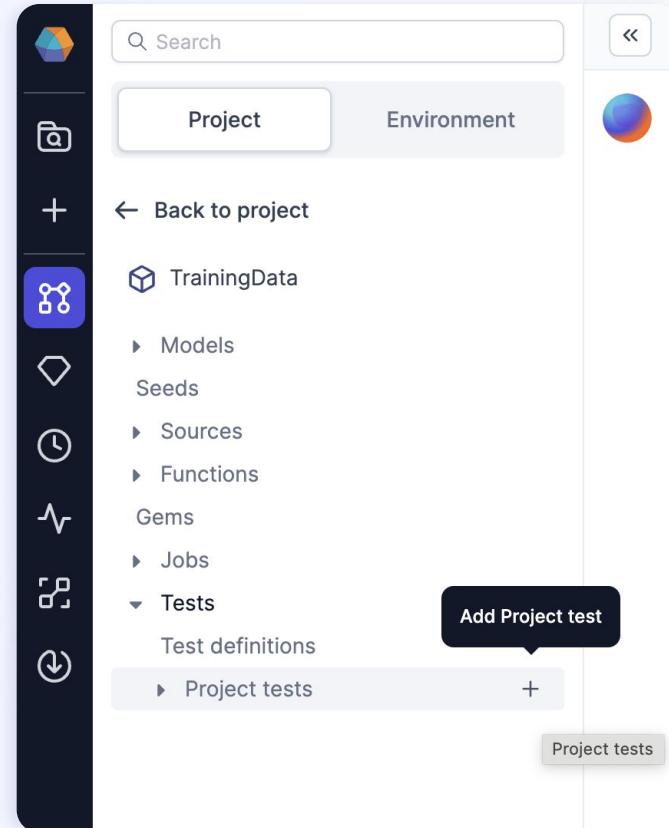
Project tests: single-use tests for models

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

data engineers



data scientists



data analysts



Prophecy

end to end platform
best practices and standards

develop deploy observe



no code
english



low code
visual drag, drop



code
spark, sql

data platforms



SQL Data
Warehouses





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](#)

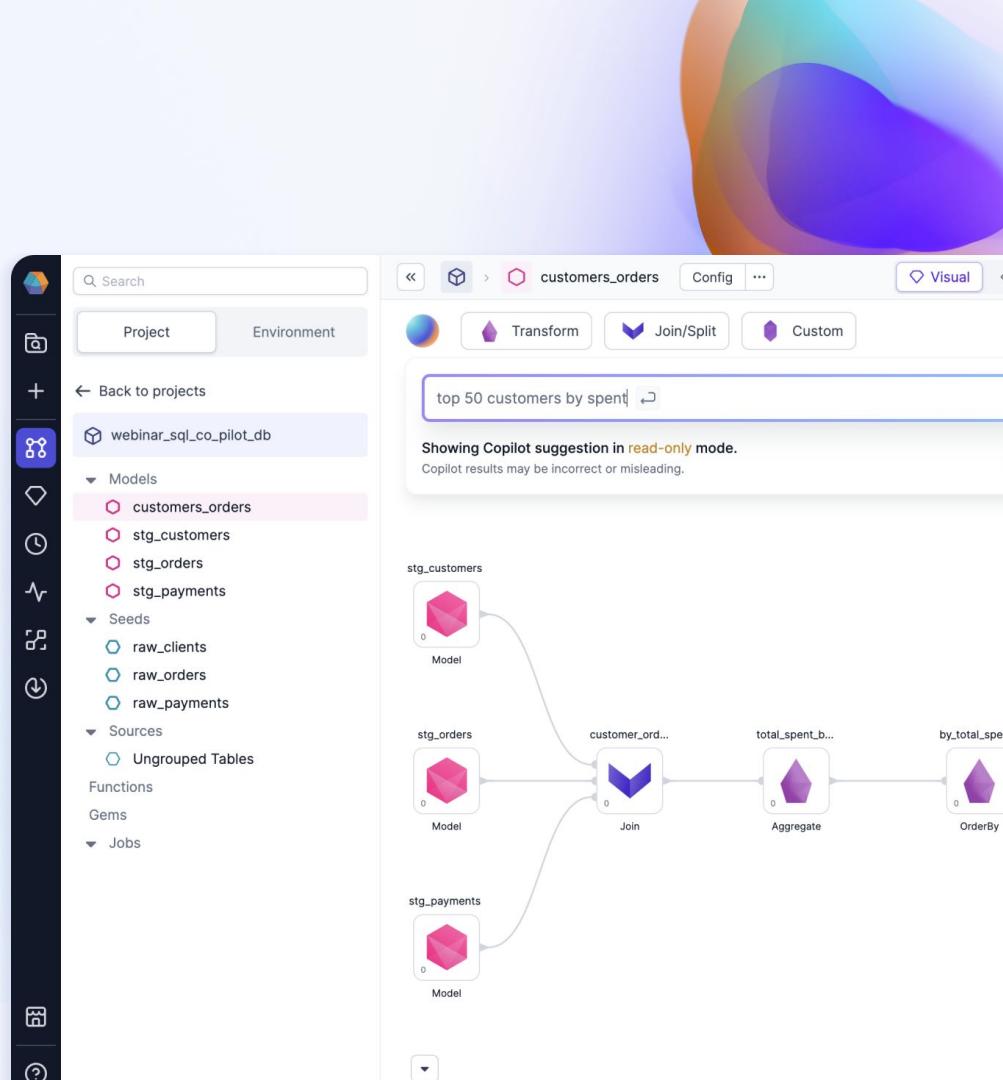
👉 Auto documentation

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

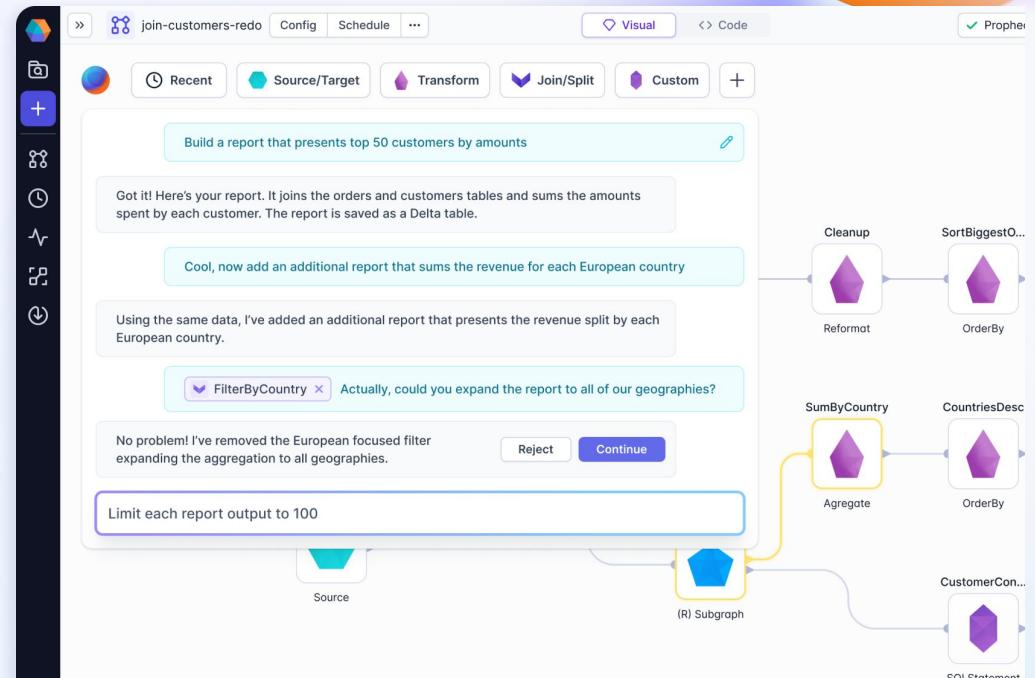
🌟 Chat interface

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

Data engineers



Data scientists



Data analysts



Data transformation copilot

Business logic - code on git

Cloud Data Platform



Cloud Data Platform





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!

The screenshot shows the Prophecy Data Pipeline interface. At the top, there are tabs for DataCleanup, Config, Schedule, Visual, and Code. Below the tabs is a toolbar with icons for Source/Target, Transform, Join/Split, Custom, Machine Learning, Subgraph, and a plus sign. A button to "Attach a cluster (SparkDev)" is also present. On the left, there's a sidebar with icons for camera, plus, cluster, clock, and power. The main area displays a data flow graph with nodes connected sequentially: Employee... (Source) → deduplicate... (Duplicate) → dynamic_... (DynamicSelect) → type-demo (TypeConversion) → rename-d... (Rename) → bulkcolumnex... (BulkColumnExtraction) → clean_de... (DataCleansing) → reorder (Reorder) → CleanedData (Target). Each node has a small preview icon above it.

How likely are you to recommend Prophecy to a friend or colleague in the data space?

Not likely 0 1 2 3 4 5 6 7 8 9 10 Very likely

Tell us a bit more about why you chose 10

Submit

Powered by Delighted



Which features will you try first?

Check them out here!

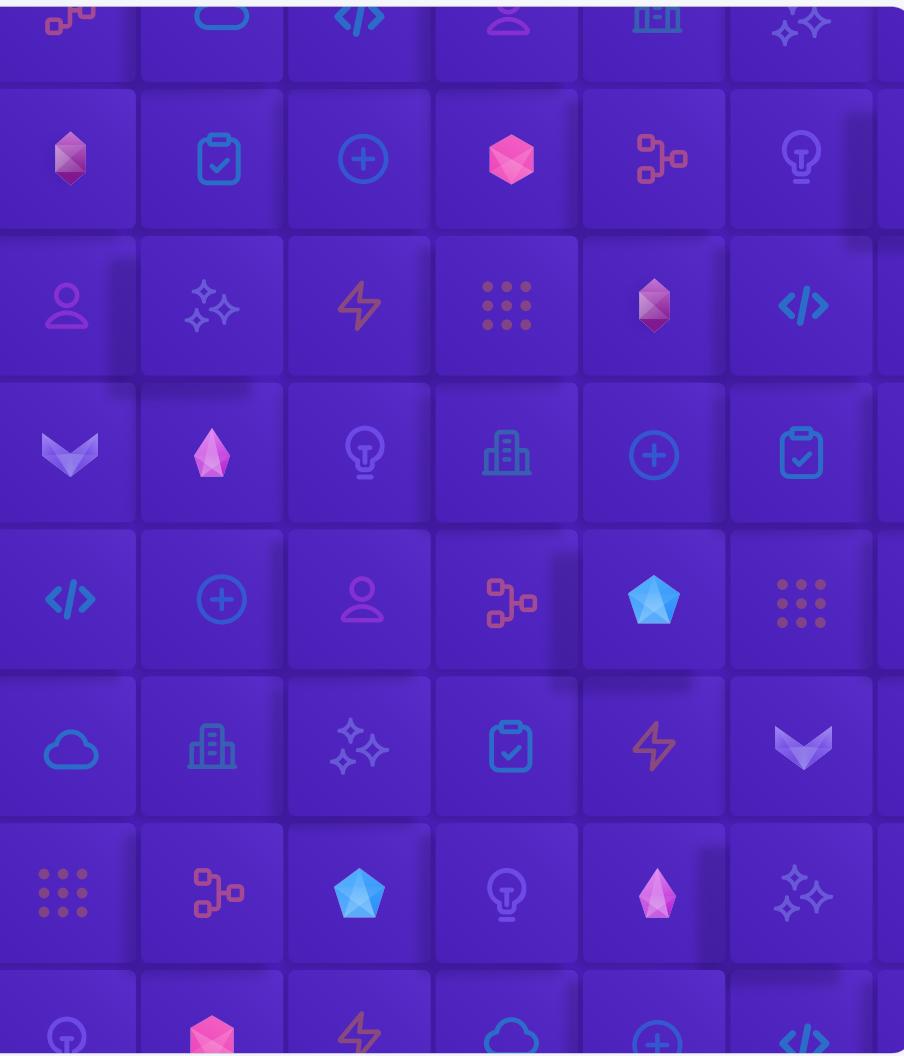
Ask your CSM to walkthrough during Office Hours
or reach out to support@prophecy.io



Prophecy

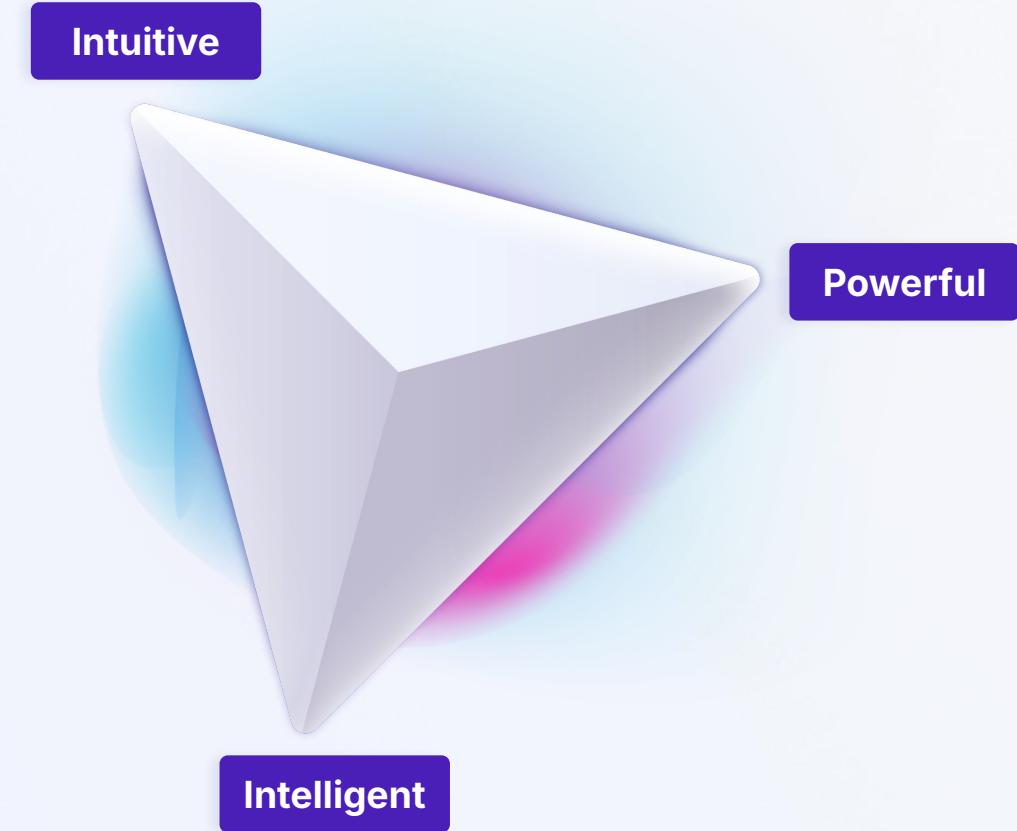


Appendix





The data transformation iron triangle



Project Environment

[← Back to projects](#)

HelloWorld

Pipelines

customers_orders

farmers-markets-irs

join_agg_sort

report_top_customers

Datasets

Jobs

Functions

Gems

DEPENDENCIES 5

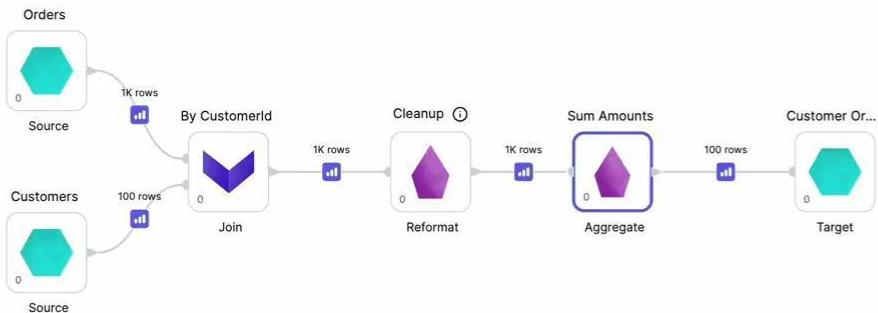
ProphecyLibsPython 1.9.22

ProphecySparkDataQuali... 0...

ProphecyWarehousePyt... 0.0...

ProphecySpark... Update 0...

ProphecySpark... Update 0...





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?

gº dev/anya

Commit Changes (13 uncommitted files)

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 successful ones. (dbt term: *singular*)

Generic Data Tests

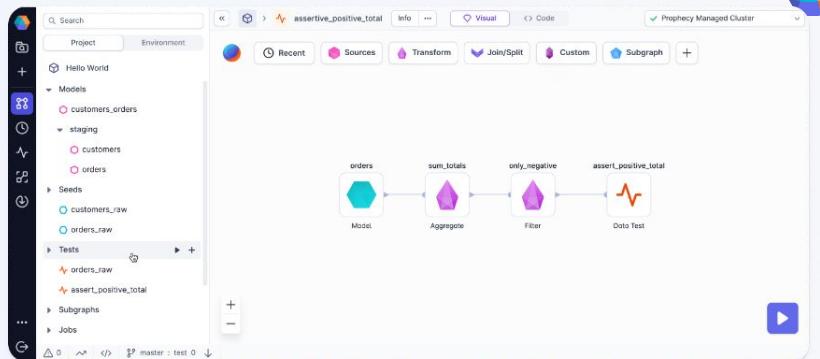
Dbt macro generated queries that can be parametrized and applied to a given model or a (set of) column(s).

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

New in 3.4

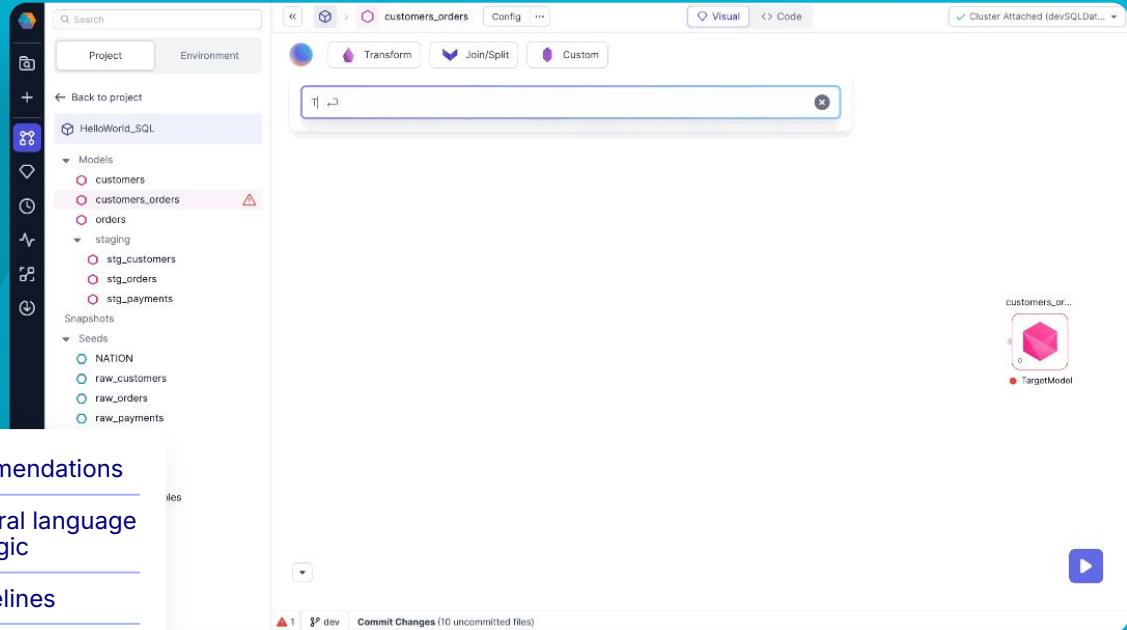
SQL

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





New features to turbocharge pipeline development

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



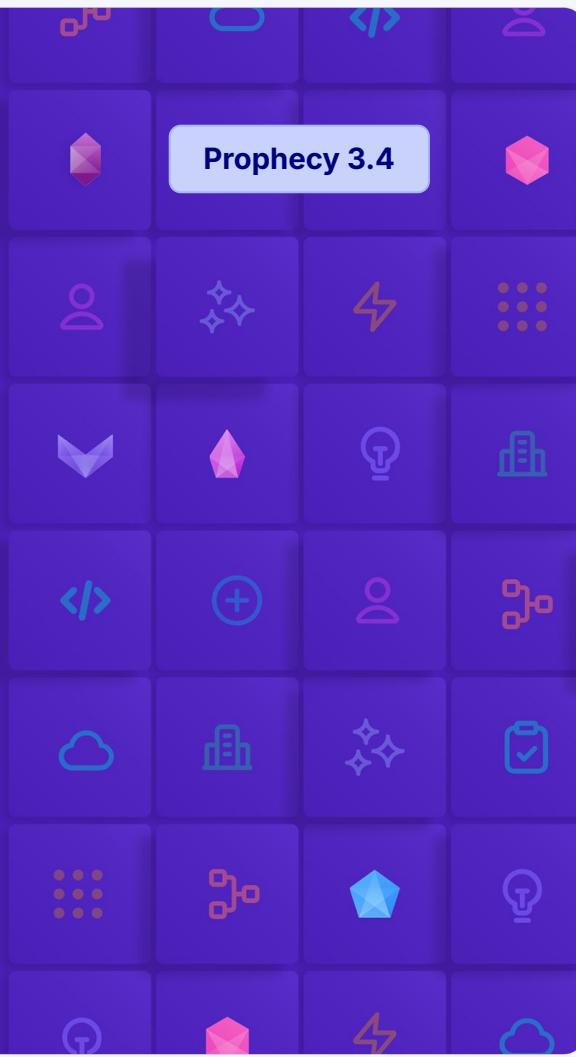
Bob Welshmer
Senior Sales Engineer
Prophecy



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

